DEVISE AND DEVELOPMENT OF DEEP LEARNING MODELS FOR STRATIFICATION OF HYPERSPECTRAL IMAGES

A Thesis
Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
by

PALLAVI RANJAN
(2K20/PHDCO/511)

Under the supervision of

Dr. Rajeev Kumar (Supervisor)

Delhi Technological University

and

Dr. Ashish Girdhar (Joint-Supervisor)

Kurukshetra University, Kurukshetra

Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2024

CERTIFICATE

Certified that **Pallavi Ranjan (2K20/PHDCO/511)** has carried out her search work presented in this thesis entitled **DEVISE AND DEVELOPMENT OF DEEP LEARNING MODELS FOR STRATIFICATION OF HYPERSPECTRAL IMAGES** for the award of **Doctor of Philosophy** from Delhi Technological University, New Delhi, India under our supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature
Dr. Rajeev Kumar
Assistant Professor
Computer Science and Engineering
Delhi Technological University
Date

Signature
Dr. Ashish Girdhar
Assistant Professor
Computer Science and Applications
Kurukshetra University
Date

CANDIDATE DECLARATION

I, Pallavi Ranjan, 2K20/PHDCO/511, hereby declare that the thesis entitled " Devise and Development of Deep Learning Models for Stratification of Hyperspectral Images," submitted to Delhi Technological University, Delhi, in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy in the department of Computer Science and Engineering, is my own original work and has been completed under the supervision of Dr. Rajeev Kumar (Supervisor), Department of Computer Science and Engineering, Delhi Technological University, Delhi, India and Dr. Ashish Girdhar (Joint-Supervisor), Department of Computer Applications, Kurukshetra University, India.

The explanations presented are based on how I read and comprehended the original texts. I have never submitted this work to any other institution for the award of any other degree, diploma, associateship, fellowship, or other title or honor.

Pallavi Ranjan
2K20/PHDCO/511
Department of Computer Science and Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Shahbad Daulatpur Main Bawana Road
Delhi-110042, India
Place:
Date :

Abstract

The classification of hyperspectral images (HSI) into categories that correlate to various land cover sorts such as water bodies, agriculture and urban areas, has gained significant attention in research due to its wide range of applications in fields such as remote sensing, computer vision, and more. This has led to the development of various deep learning models that include supervised and semi-supervised, for HSI classification. Among the aforementioned classes of models, the supervised networks have evolved to achieve almost perfect classification accuracy. Nevertheless, the process of obtaining labelled samples continues to pose a challenge in HSI classification, as the labelling remains a manual, time-consuming, and labour-intensive task, which necessitates the expertise of individuals to identify and label each pixel in the image. Furthermore, most of the existing supervised models are computationally slower due to the massive computations involved. To address these problems of the existing works, in this thesis, we propose five new deep learning-based models for HSI classification.

In our first work, we propose a novel lightweight network, Xcep-Dense, a hybrid classification model that combines the core benefits of the extreme version of inception and dense networks. The Xception network employs depth-wise separable convolutions, and the 3D slicing phenomenon, which requires fewer parameters, is computationally efficient and provides excellent classification accuracy. The proposed network is configured with dense network and optimized to alleviate overfitting. Xcep-Dense's performance is validated using two benchmark hyperspectral datasets, Indian Pines and Salinas.

In our second work, we propose a Siamese network based deep learning model which implements one shot classification model and can work with limited samples and/or imbalanced samples. The proposed Siamese network has a handcrafted feature generation network that extracts discriminative features from the image. Experimental findings demon-

strate that the proposed network is capable of improving the classification performance with an overall accuracy, with a small scale training data.

To tackle the labelled limited samples problem, the third work introduces a novel semi-supervised network constructed with an autoencoder, siamese action, and attention layers that achieves excellent classification accuracy. The proposed convolutional autoencoder is trained using the mass amount of unlabelled data to learn the refinement representation referred to as 3D-CAE. The added siamese network improves the feature separability between different categories and attention layers improve classification by focusing on discriminative information and neglecting the unimportant bands. The efficacy of the proposed model's performance was assessed by training and testing on both same-domain as well as cross-domain data and found to achieve 91.3 and 93.6 for Indian Pines and Salinas, respectively.

In our fourth work, we integrate autoencoders and Generative Adversarial Networks for enhancing feature representations and mitigating the constraints imposed by limited labeled data. Leveraging the power of semi-supervised learning paradigms, this innovative approach offers substantial progress in feature extraction, data augmentation, and classification accuracy. It extends beyond traditional hyperspectral image classification boundaries by addressing zero-shot learning and integration of text embeddings to enrich feature representations. The outcome is a precise classification framework that accommodates the intricacies of both same-domain and cross-domain datasets, ultimately pushing the boundaries of hyperspectral image classification.

In our final work, we present an innovative semi-supervised framework that harmoniously combines unsupervised feature learning with the employment of graph-based convolutional neural networks (GCNs). Our approach harnesses the latent knowledge hidden within vast pools of unlabelled HSI data using autoencoders, which extract meaningful features. These features are then incorporated into a GCN-based architecture, leveraging spatial relationships among neighboring pixels. The fusion of autoencoder-based learning and graph-based techniques enables our model to achieve excellent classification accuracy, even in scenarios with minimal labelled samples.

ACKNOWLEDGEMENT

Karpuragauram Karunavataram, Sansarasaram Bhujagendraharam
Sadavasantam Hridayaravinde, Bhavam Bhavani Sahitam Namami

At the core of my academic odyssey, I express my heartfelt gratitude to the divine presence for being the unwavering source of strength, resilience, and guidance. In moments of challenge and triumph alike, the divine hand has provided the fortitude needed to navigate the complexities of this academic journey. The belief that there is a higher purpose and a guiding force has been a constant beacon, offering solace and inspiration during the trials and triumphs of my research endeavors.

I express immense gratitude to **Dr. Rajeev Kumar** and **Dr. Ashish Girdhar** for their invaluable guidance, mentorship, and expertise. Their unwavering support has been pivotal in shaping the direction and success of this research. Specifically, I extend my heartfelt appreciation to **Dr. Ashish Girdhar** for not only believing in my capability to start my Ph.D. but also for his continuous guidance and mentorship throughout this academic odyssey. His unwavering belief in my potential has been a driving force, setting the foundation for my research endeavors.

I want to especially acknowledge **Dr. Rajeev Kumar** for accepting me as his student during a tumultuous period in my life. I am grateful to **Dr. Rajeev Kumar** for helping me navigate the stormy seas when my boat was sinking, ensuring that I not only stayed afloat but also emerged stronger from the challenges faced. His understanding and support amidst the roller coaster of circumstances have been invaluable, shaping not only my academic pursuits but also fostering personal growth.

I express immense gratitude to my son, **Anmay**, for the myriad compromises he made while his mother was engrossed in the demands of academia. His patience, understanding, and unwavering support have been a source of inspiration, emphasizing the significance of family support in academic pursuits. To my husband, I extend my deepest appreciation for his unwavering emotional support. His encouragement, resilience, and belief in my

capabilities have been a constant anchor, sustaining me through the highs and lows of this journey.

To my family - your unwavering love and understanding have been my pillar of strength. Your constant encouragement fueled my determination to pursue this endeavor. In every challenge faced, your support has been a source of inspiration and motivation, making this academic journey richer and more meaningful. A heartfelt acknowledgment goes to my mother-in-law for her steadfast support during challenging times. Her presence, encouragement, and understanding have been a pillar of strength, demonstrating the importance of a supportive family network. Each of these individuals has played a unique and significant role in my academic journey, and I am profoundly thankful for their enduring contributions, which extend far beyond mere words.

My sincere appreciation extends to the members of the SRC and DRC for their insightful comments and pertinent suggestions during the research proposal phase. Special thanks to **Prof. Rajni Jindal**, **Prof. D.K. Lobiyal**, **Prof. Rajesh Kumar Bhatia**, **Prof. Vinod Kumar** for their continuous support and guidance. Their collective wisdom and constructive feedback have significantly contributed to the refinement and success of this research.

A heartfelt thank you to my friends who provided encouragement, understanding, and a supportive environment during the ups and downs of this academic journey. The camaraderie and shared experiences have added a valuable dimension to this academic pursuit, making the journey more enriching and memorable.

This thesis is a testament to the collective efforts and encouragement of these remarkable entities and individuals. I am profoundly grateful for their enduring contributions, which have significantly impacted both my academic and personal growth.

Place: Dubai Pallavi Ranjan

Date: May 29, 2024

List of Abbreviations

HSI Hyperspectral Imaging

HSIC Hyperspectral Image Classification

SVM Support Vector Machine

DL Deep Learning

ML Machine Learning

RF Random Forest

CNN Convolutional Neural Network

MRF Markov Random Field

RGB Red Green Blue

2D Two-Dimensional

3D Three-Dimensional

AE Autoencoders

AL Active Learning

AN Attention Networks

DBN Deep Brief Network

DML Deep Metric Learning

DA Domain Adaptation

EL Ensemble Learning

FSL Few Shot Learning

GAN Generative Adversarial Network

GBL Graph Based Learning

GCN Graph Convolutional Networks

LSTM Long Short Memory Network

ML Machine Learning

TL Transfer Learning

TN Transformer Network

Su Supervised

S-Su Semi-Supervised,

Un Unsupervised

OA Overall Accuracy

AA Average Accuracy

KA Kappa Coefficient

KC Kappa Coefficient

BO Botswana

HU Houston University

IP Indian Pines

KSC Kennedy Space Center

Pavia U Pavia University

PU Pavia University

PC Pavia Centre

SA Salinas

MLR Multinomial Logistic Regression

SAE Stacked Autoencoder

MLP Multi Layer Perceptron

RNN Recurrent Neural Network

DT Decision Trees

PCA Principal Component Analysis

BO Botswana,

HU Houston University,

IP Indian Pines,

KSC Kennedy Space Center,

PU Pavia University,

PC Pavia Centre,

SA Salinas

SA-A Salinas-A

TP True Positive

TN True Negative

FP False Positive

FN False Negative

PCA Principal Component Analysis

ICA Independent Component Analysis

LLE Local Linear Embedding

NKM Nonlinear Kernel Mapping

LDA Linear Discriminant Analysis

LR Linear Regression

KNN K-Nearest Neighbour

DWT Discrete Wavelet Transform

FCN Fully Connected Network

FSR Few-Shot Recognition

OSR One-Shot Recognition

ZSL Zero-Shot Learning

UL Unsupervised Learning

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

With the technological advancement in spectroscopy, the Earth observation fields have seen tremendous growth. This growth can be attributed mainly to the amalgamation of sensory capabilities of various beings, including *bumble bees* that perceive only a limited range of the electromagnetic spectrum—visible light from *(10-380 nm)*, *human eyes* extend into the ultraviolet range *380 to 700 nm*, *goldfish* delve into the infrared *(700 nm-1 mm)*, and many others. The various ways these beings perceive the electromagnetic spectrum are combined into one unified imaging system. [2, 3]. This integration becomes especially significant in remote sensing applications, where spectroscopy plays a fundamental role in capturing the electromagnetic radiation emanating from the Earth's surface with the help of a variety of imaging sensors, mounted on aircraft or spacecraft platforms. The sensor may include multispectral and hyperspectral sensors, where the former ones gather data in a limited number of non-contiguous wavebands, while the later captures images across contiguous narrow bands spanning the visible to near-infrared regions. Therefore, multispectral imaging systems excel in identifying prevalent ground-cover types and have diverse applications. However, their limitations arise from acquiring data in a limited number of non-contiguous spectral bands, potentially leading to the loss of intricate details in images captured with significant spectral gaps and low resolutions [4]. To enhance the capacity for discerning subtle spectral variations, hyperspectral imaging systems were developed and are being used. An overview of a sensor capturing a target area for subsequent processing is depicted in 1.1.

Hyperspectral imaging systems enable a more comprehensive analysis of spectral data by offering valuable insights into the composition of objects in the scene. This significantly helps in the classification of these objects by stratifying the pixels in the hyperspectral image (HSI) and makes it highly valuable for various practical applications such as environmental monitoring, military target detection, and precision agriculture.

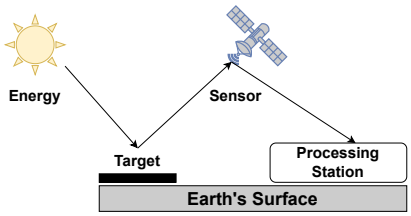
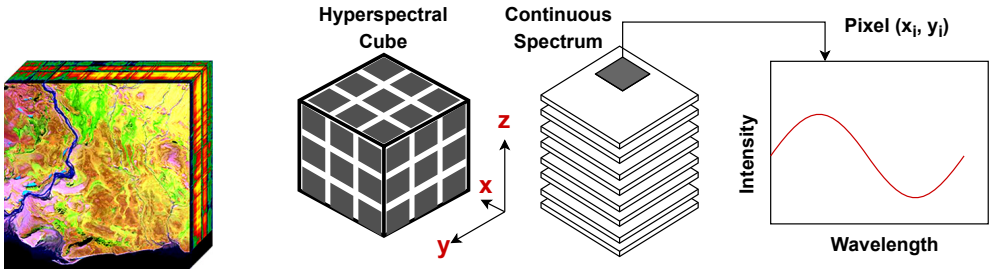


Figure 1.1: A sensor capturing a target for processing

1.1 Unravelling Hyperspectral Imaging (HSI)

The term "Hyperspectral" is a combination of "Hyper," meaning excessive, and "Spectral," referring to the number of spectral bands. In a metaphorical sense, hyperspectral sensors gather data in the form of a series of 'images.' Each of these images corresponds to a specific narrow wavelength range in the electromagnetic spectrum, commonly referred to as a spectral band. In an expanded spectrum, each channel generates an image encoded with grayscale levels, collectively forming a multidimensional entity known as an HSI cube. Figure 1.2a illustrates a typical hyperspectral dataset, where imaging and spectroscopy unite to yield an extensive reserve of spectral and spatial information about the scene [5]. Conceptually, a hyperspectral image can be visualized as a cube, with its length (x) and width (y) representing the spatial coverage (number of pixels) of the Two-Dimensional (2D) image at each wavelength, and its depth (z) symbolizing the number of spectral bands within the hyperspectral image, as depicted in 1.2b. In simpler terms, HSI seamlessly integrates digital imaging with spectroscopy, providing abundant spatial and spectral information for every pixel within an image. This technology helps us understand the Earth, other planets, and space better than regular colour cameras and also empowers the identification, measurement, and precise characterization of materials based on their unique chemical and physical properties [6, 7, 8], leading to the widespread adoption of HSI across various domain.



(a) (b)

Figure 1.2: (a) Hyperspectral Cube (Image Reference [1]) (b) HSI Cube showing a continuous spectrum of bands followed by spectrum for each pixel (x_i, y_i)

1.1.1 Transformative power of HSI across disciplines

Since the HSI holds abundant spatial and spectral information, it has been proven to be remarkably transformative in various fields. Some of the prominent fields where HSI has already demonstrated its transformative power are as follows:

-

- In *environmental monitoring*, HSI's unique spectral signatures have been instrumental in identifying substances for pollution monitoring and biodiversity assessment. For instance, a case study in [9] demonstrates how HSI is used to monitor harmful algal blooms and assess the impact of human activities on aquatic ecosystems, aiding in sustainable environmental management.

- *Precision agriculture* benefits from HSI's potential in mapping crop fields and monitoring crop health in real-world farming scenarios. Case studies like those in [10, 11, 12] showcase how HSI technology is utilized to optimize agricultural practices, minimize food waste, and detect pollutants during harvests, thereby enhancing agricultural sustainability and productivity.

- In *mineral exploration and mapping*, HSI is used in efficient resource exploration and mineral identification and facilitates the identification of subsurface minerals and ores indicative of a valuable mineral deposit suitable for mining. [6].

- *Military* benefits from enhanced situational awareness proving invaluable for target detection in defence and security in reconnaissance and surveillance missions through compact and high-resolution sensors. HSI proves effective in the early detection of metal corrosion on aircraft and naval vessels, leading to cost reduction and improved operational uptime for these valuable assets. [13, 14].

- In *biomedical imaging*, HSI is used to identify tissues and biomarkers for identifying tumour margins in real-time brain surgeries. HSI enables detailed examination of biological structures relevant to medical specialities such as pathology, oncology, and neurology for the detection of various cancers and diseases[15].

- In *remote sensing*, HSI integrates actionable insights for forestry, urban planning, and disaster management [16]. In forestry, HSI can be used to assess vegetation health and identify specific plant species. In urban planning, it aids in land-use classification and monitoring changes over time. For disaster management, HSI can contribute to assessing the impact of natural disasters, such as wildfires or floods, by analyzing changes in the landscape.

- In ensuring *food quality and safety*, HSI plays a pivotal role in controlling the quality of the food, ensuring that products meet regulatory standards and are safe for consumption enabling the identification of contaminants, spoilage, and adulteration in food items [17, 18, 19, 20].

However, one has to process and analyze the HSI dataset to extract valuable insights by mainly doing classification. Classification in hyperspectral imaging is of paramount importance as it allows for the categorization and identification of materials, objects, and/or land cover types present in the captured scenes. Accurate classification of hyperspectral images is crucial for enabling informed decision-making. Despite its significance, HSIC presents challenges due to the numerous spectral bands in the HSI dataset. To address these challenges and enhance accuracy, early HSIC stages relied on conventional image processing techniques. Although effective, these methods encountered difficulties in managing the intricate and high-dimensional characteristics of hyperspectral data. In the next sub-section, we will explore the evolution of HSIC methods, discussing the transition from conventional techniques to the integration of powerful Machine Learning models, ultimately leading to the rise of deep learning.

1.1.2 Evolution of HSIC methods

In its early stages, HSIC relied on conventional image processing techniques such as subspace projection approach [21], knowledge-based approaches [22], etc. While effective, these methods encountered difficulties in managing the intricate and high-dimensional characteristics of hyperspectral data, leading to the adoption of Machine Learning (ML) models like SVM [23, 24], Random Forest (RF) [25, 26, 27], and DT [28] for HSIC. While earlier ML models laid the groundwork for remote sensing applications, demonstrating good classification accuracy and versatility in handling various data types, they also exhibited disadvantages such as slow convergence, overfitting, manual feature extraction, and prolonged training and testing times. The complex nature of HSI data further challenges traditional ML models, dealing with inherently nonlinear relationships between the extracted spectral bands and corresponding materials [7].

To address the aforementioned limitations of traditional methods, deep learning-based HSIC models came into prominence because of their powerful feature extraction ability from images [7] and enhanced discrimination capacity [6, 29]. In no time, deep-learning models, including CNNs [30, 31, 32], Recurrent Neural Network (RNN)s [33, 34], AE [35, 36], Transfer Learning (TL) [37, 38], GAN [39, 40], etc. became integral to the HSIC landscape, excelling at automatically learning hierarchical features from hyperspectral data [41, 42, 33, 43, 44]. CNNs, specifically, brought spatial hierarchy learning to HSIC, ex-

tracting spatial features from hyperspectral images for better classification of objects, resulting in improved HSI accuracy and performance. However, challenges in interpretability led to exploration into specialized architectures such as Siamese CNN, attention networks, inception networks, and others to address constraints posed by conventional CNNs [27, 30, 45, 46, 47, 48, 49, 50, 51, 52]. Despite advancements, there is still a room for improving lightweight deep learning models for enhanced accuracy and faster convergence in HSIC. While existing models show good accuracy, further exploration is needed to address challenges related to limited labelled samples, the Hughes phenomenon, working effectively with unlabeled data or dealing with cross-domain data. Hughes phenomenon, also called as the curse of dimensionality exacerbates the challenges associated with high-dimensional data, such as HSI. In this context, increased dimensionality leads to computational complexities and sparse data distribution, making analysis and interpretation arduous. For instance, visualizing or comprehending data in high-dimensional space becomes challenging due to the diminishing relevance of distances between data points and the exponential increase in data required to adequately represent the space. In HSIC, the curse of dimensionality poses obstacles like overfitting and heightened computational demands, impairing classification accuracy and efficiency.

1.1.3 Problem Statement

Effectively classifying hyperspectral images encounters challenges due to the complex interaction between spectral and spatial data. The complexities lie in the varying significance of features and the accurate classification of diverse categories [53, 54]. The integration of supervised models becomes crucial for unlocking the potential of deep learning algorithms in hyperspectral image analysis. Leveraging carefully labelled datasets, these models discern complex relationships between spectral responses and various land cover types, ensuring precise classification and facilitating the extraction of critical features for comprehensive analysis in practical scenarios [55, 56, 57].

Also, to deal with the scarcity of labelled samples, the inclusion of semi-supervised learning models is essential. These models, relying on both labelled and unlabeled data, enhance classification accuracy significantly, even with limited labelled data [36, 58]. HSIC presents challenges due to inter-class and intra-class similarities. Utilizing 1DCNN, 2DCNN, or 3DCNN proves viable for classification, given HSIC's reliance on both spectral and spatial information. However, the computational demands in processing the voluminous and spectral dimensions of HSI can be burdensome, potentially hindering feature extraction and leading to underperformance [59]. One primary obstacle in HSI lies like the data itself—numerous narrow spectral bands coupled with limited labelled training data, giving rise to the Hughes Phenomenon or the curse of dimensionality [58, 60]. This phenomenon

occurs when the number of labelled training samples is significantly lower than the spectral bands present, resulting in a diminished predictive performance for supervised [59] and semi-supervised learning [61] methods in HSIC.

Therefore, this research first scrutinizes prior studies for cutting-edge deep learning models, benchmark datasets, and associated classification accuracies in HSIC, establishing a baseline understanding of the current state-of-the-art. Second, it involves developing an innovative supervised model tailored for HSIC to surpass existing benchmarks, integrating novel features and methodologies focused on enhancing classification accuracy. Third, it explores the potential of unlabeled HSI datasets using semi-supervised learning to extract valuable insights and improve classification accuracy, aiming to overcome limitations posed by limited labeled samples in HSIC.

1.2 Research Objectives

The adoption of deep learning technology for HSIC has been a breakthrough as the deep learning models have shown the excellent ability of automatic feature extraction with improved discrimination capacity. This has led to numerous works in the domain with the objective of improving classification accuracy and few to tackle the limitation of limited samples of labelled data and high complexity. However, the HSIC still faces the challenges as far as accuracy, cross-domain adaptation, convergence speed, limited labelled data, and working with unseen data are concerned. These challenges have motivated to develop efficient deep learning models for stratification of hyperspectral images using both supervised and semi-supervised approaches. Accordingly, the following objectives are proposed for this thesis work.

- **Objective 1**: To study and analyze various available deep learning models for the classification of Hyperspectral Images.
- **Objective 2**: To develop a novel lightweight deep learning algorithm using supervised learning that can extract discriminative features and enhance classification performance.
- **Objective 3**: To develop a novel semi-supervised deep learning algorithm that works on the problem of limited labelled samples.
- **Objective 4**: To experimentally evaluate and compare the proposed models with state-of-the-art models.

These research objectives might face the challenge of limited labelled data or computational efficiency. This will be solved by generating new data or optimizing the algorithms

to get better performance on HSI data. Particularly, the second objective seeks to develop a lightweight DL algorithm, addressing resource-intensive models to enhance computational efficiency in HSIC. However, the third objective focuses on developing a semi-supervised DL algorithm to tackle limited labelled data in HSIC. By leveraging both labelled and unlabeled data, this objective aims to boost classification accuracy and generalization, expanding the usability of deep learning in scenarios with sparse labelled samples. Following the analysis of the existing challenges and objectives outlined, this thesis endeavours to address the gaps in current deep learning approaches for HSIC classification.

1.3 Contributions of Thesis

This thesis contributes significantly to the field of Hyperspectral Image Classification (HSIC) by addressing the challenges posed by relationships in spectral and spatial data.

The key contributions of this thesis are as follows:

- The thesis undertakes a rigorous investigation and analysis of existing deep learning models employed for the classification of HSI. Through a detailed review of these models, the thesis aims to provide a holistic understanding of their capabilities, limitations, and applicability in the context of HSIC.

- The thesis endeavours to develop a pioneering deep-learning algorithm with a focus on lightweight architecture. This algorithm will leverage supervised learning techniques to extract highly discriminative features, thereby significantly enhancing the overall classification performance of HSI. By prioritizing improved classification accuracy, the proposed lightweight algorithm aims to contribute to the development of more streamlined and resource-efficient deep learning methodologies for HSI.

- Addressing the challenge of limited labeled samples, the thesis seeks to devise an innovative semi-supervised deep learning algorithm. This algorithm will be specifically tailored to tackle the complexities associated with the scarcity of labeled data in HSIC. By integrating semi-supervised learning techniques, the proposed algorithm aims to unlock the potential of unlabeled data, thereby expanding the scope and accuracy of HSIC models in real-world applications.

- The thesis aims to conduct comprehensive experimental evaluations and comparative analyses of the newly developed deep learning models with state-of-the-art methodologies in the field. This comparative analysis will contribute to the advancement and refinement of HSIC methodologies, paving the way for improved and more reliable classification techniques for hyperspectral data.

1.4 Alignmnent of Research Objectives with Publications

The table 1.1 below aligns research objectives with their corresponding publications.

1.5 Dissertation Organization

- Chapter 1: **Introduction to the HSI Field**

  This chapter serves as an introductory section encompassing the first chapter, providing an extensive introduction to the HSI field and the broader context within which the dissertation is developed. This chapter lays the groundwork for the subsequent discussions, establishing the fundamental background and significance of the research conducted.

- Chapter 2: **Literature Review and Theoretical Background**

  This chapter provides a detailed literature review, offering an in-depth overview of the frameworks proposed for HSIC. Additionally, this chapter furnishes the theoretical background underpinning the proposed methodologies, providing a comprehensive understanding of the theoretical foundations that guide the subsequent analysis and experiments.

- Chapter 3: **Supervised Deep Learning Model I**

  This chapter is dedicated to the exploration of the lightweight methods using supervised learning strategy. The proposed lightweight Xcep-Dense network addresses the challenge of extensive parameters in existing deep learning models, offering comparable classification accuracy with significantly fewer parameters.

- Chapter 4: **Supervised Deep Learning Model II**

  The proposed few shot classification model, based on a Siamese network with enhanced feature extraction, outperforms current models in the literature, achieving higher accuracy than most of the state-of-the-art models.

- Chapters 5 **Semi-Supervised Deep Learning Model I**

  This chapter presents a novel semi-supervised deep learning network for HSI classification, which incorporates an autoencoder, siamese and attention layers, working on both same domain and cross domain datasets.

- Chapter 6 **Semi-Supervised Deep Learning Model II**

  This chapter presents a comprehensive framework that leverages the synergistic potential of hybrid CNN-GAN-autoencoder models, text embeddings, and zero-shot

Table 1.1: Research Objectives and Corresponding Publications

Research Objective	Corresponding Publication
RO1: To study and analyze various available deep-learning models for the classification of Hyperspectral Images.	1. Pallavi Ranjan and Ashish Girdhar. A Comprehensive Systematic Review of Deep Learning Methods for Hyperspectral Images Classification. International Journal of Remote Sensing. Taylor and Francis. **Published**. IF : 3.531, ID: 2133579 https://doi.org/10.1080/01431161.2022.2133579
RO2: To develop a novel lightweight deep learning algorithm using supervised learning that can extract discriminative features and enhance classification performance. **RO4**: To experimentally evaluate and compare the proposed models with state-of-the-art models.	1. Pallavi Ranjan; Rajeev Kumar; Ashish Girdhar; Recent CNN Advancements For Stratification of Hyperspectral Images, Published at International Conference on Information Systems and Computer Networks (ISCON 2023), SCOPUS, **Published** March 03, 2023 https://ieeexplore.ieee.org/document/10112174 2. Pallavi Ranjan and Ashish Girdhar, Deep Siamese Network with Handcrafted Feature Extraction for Hyperspectral Image Classification. Multimedia Tools and Applications. IF: 3.6 Springer Nature. **Published** May 15, 2023 https://doi.org/10.1007/s11042-023-15444-4 3. Pallavi Ranjan and Ashish Girdhar. Xcep-Dense: a novel lightweight extreme inception model for hyperspectral image classification. International Journal of Remote Sensing. Taylor and Francis. **Published** Oct 14, 2022 ; IF : 3.531 https://doi.org/10.1080/01431161.2022.2130727
RO3: To develop a novel semi-supervised deep learning algorithm that works on the problem of limited labeled samples. **RO4**: To experimentally evaluate and compare the proposed models with state-of-the-art models.	1. Pallavi Ranjan; Rajeev Kumar; Ashish Girdhar; 3D-convolutional-autoencoder embedded Siamese-attention-network for classification of hyperspectral images. Neural Comput & Applic 36, 8335–8354 (2024). https://doi.org/10.1007/s00521-024-09527-y. 2. Pallavi Ranjan; Rajeev Kumar; Ashish Girdhar; Revolutionizing Hyperspectral Image Classification: Unifying Autoencoder- Enhanced GANs with Convolutional Neural Networks and Zero-Shot Learning. Submitted to Computers and Electrical Engineering (**Major Revisions Submitted**) 3. Pallavi Ranjan and Ashish Girdhar. A novel spectral-spatial Wasserstein GAN-GP integrated with convolutional LSTM for hyperspectral image classification. Submitted to Evolving Systems. Springer. (**Under Review, Submitted on Nov 17, 2022**) 4. Pallavi Ranjan and Ashish Girdhar, A Comparison of Deep Learning Algorithms Dealing With Limited Samples in Hyperspectral Image Classification, SCOPUS, Published and Best Paper Award at OPJU INTERNATIONAL TECHNOLOGY CONFERENCE On Emerging Technologies for Sustainable Development, **Published** February 13, 2023 https://ieeexplore.ieee.org/document/10114005 5. Pallavi Ranjan; Rajeev Kumar; Ashish Girdhar; Unlocking the Potential of Unlabeled Data: Semi-Supervised Learning for Stratification of Hyperspectral Images," 2023 OITS International Conference on Information Technology (OCIT), Raipur, India, 2023, pp. 938-943, doi: https://ieeexplore.ieee.org/document/10430513.

learning for advanced hyperspectral image classification. The proposed model not only addresses the challenges of limited labelled data but also enhances feature representations through unsupervised and semi-supervised learning paradigms.

- Chapter 7: **Semi-Supervised Deep Learning Model III**

  This chapter introduces an approach for effectively classifying hyperspectral images. By merging AE and GCN, we achieved highly promising outcomes when tested against various standard datasets. This strategic fusion of spectral and spatial insights enabled our model to thrive, especially in scenarios where spatial associations played a pivotal role, ultimately leading to highly accurate classifications.

- Chapter 8: **Conclusion and Future Research Directions**

  This chapter serves as the concluding segment of the dissertation, offering a comprehensive summary of the most significant findings and insights derived from the research. Further, most prominent future research directions, highlighting potential areas for further exploration and development in the field of hyperspectral image classification, are also discussed.

CHAPTER 2

LITERATURE REVIEW

This chapter is dedicated to studying and analysing the recent works in the domain of hyperspectral image classification, which has undergone a paradigm shift with the emergence of advanced deep learning techniques. For this, a systematic and comprehensive literature review has been done to provide an exhaustive overview of the existing body of knowledge, elucidating crucial developments and trends in the realm of deep learning models specifically designed for hyperspectral image classification. The review has been done in two phases. In the first phase, a meta-analysis has been conducted to furnish a comprehensive understanding of the current landscape. Next, the subsequent sections meticulously delve into pertinent studies, unravelling key findings that not only inform but also inspire the proposed novel models within this dynamic and evolving field.

2.1 Meta-Analysis

The objective of the meta-analysis was twofold: first, to synthesize and systematically review the existing literature on HSIC utilizing deep learning approaches, and second, to extract overarching trends, patterns, and insights from the collective body of research. This meta-analysis aims to provide a comprehensive overview of the methodologies, findings, challenges, commonalities, variations and gaps reported across diverse studies, enhancing the understanding of the current state-of-the-art in HSIC. This section is divided as follows: The initial subsection (2.1.1) centres on elucidating the intricacies of the data collection process. Subsequently, the second subsection introduces the research questions (as outlined in 2.1.2). Following this, subsection 2.1.3 encompasses a discussion on benchmark datasets, while subsection 2.1.4 delineates the various performance metrics employed in the analysis.

2.1.1 Data-Collection

To initiate data collection, the Scopus database was searched to create a collection of similar articles focussed on the targeted area i.e., 'Hyperspectral Image Classification using Deep Learning'. The process is depicted in Figure 2.1.

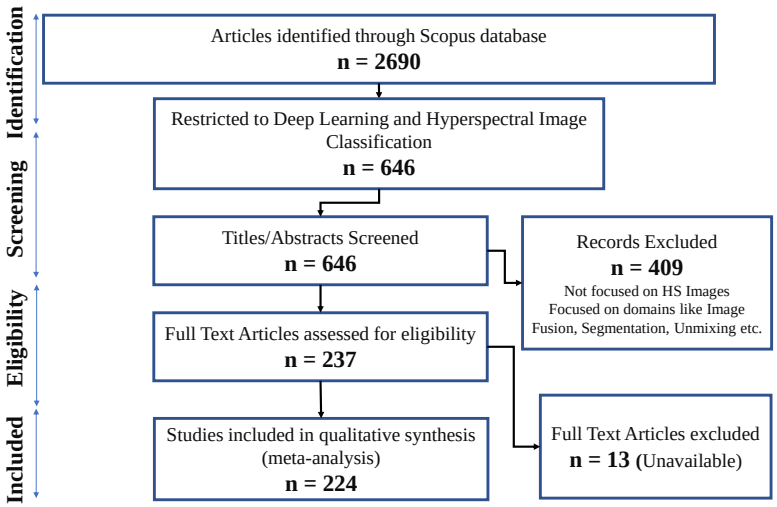


Figure 2.1: Preferred Reporting items for systematic reviews and meta-analyses for this study

- In step **1**, a title/abstract/keyword search was conducted in the Scopus database using the terms 'Deep Learning' AND 'Hyperspectral Image Classification'. Searching was restricted to the quality journals in this field, namely Remote Sensing, IEEE Transactions On Geoscience And Remote Sensing, IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing, IEEE Access, International Journal Of Remote Sensing, ISPRS Journal Of Photogrammetry And Remote Sensing, Neurocomputing, IEEE Geoscience And Remote Sensing Letters, and International Geoscience And Remote Sensing Symposium. Table 2.1 shows the inclusion criteria for the articles included in the survey.

- In step **2a**, a total of 2690 records were retrieved. Each title in the initial sheet was screened, and the articles focussed on multispectral data, lidar data, image unmixing, segmentation, object detection, and others were removed from this database. This number went down to 646, which contained all the articles restricted to DL and HSIC.

- In another filtration step **2b**, abstracts of each article were reviewed, resulting in the removal of 409 articles related to band selection, scene classification, image fusion, core neural networks, literature surveys, and duplicate content. The survey exclusively included empirical studies containing experimentation.

S. No	Criterion	Inclusions	Exclusions
1	Perform Search	*Keywords*: Deep Learning AND Hyperspectral Image Classification *Focused Journals*: • Remote Sensing • IEEE Transactions On Geoscience And Remote Sensing • IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing • IEEE Access • IEEE Geoscience, And Remote Sensing Letters • International Geoscience And Remote Sensing Symposium • Neurocomputing • International Journal Of Remote Sensing • ISPRS Journal Of Photogrammetry And Remote Sensing	Restricted to the keywords and journals published before 2019
2a	Screening of Titles	Articles restricted to Deep Learning and Hyperspectral Image Classification	Articles related to multispectral data, LIDAR data, image unmixing, segmentation, object detection, and others
2b	Screening Abstracts	Only empirical studies containing experimentation work based on deep learning models for hyperspectral image classification	409 articles related to band selection, scene classification, image fusion, core neural networks, literature surveys, and duplicate articles were removed
3	Final list of Articles	Empirical studies/articles with full text available • YEAR 2019 onwards = [195 Articles] = 88% • YEAR 2016-2018 = [28 Articles] = 12%	Articles with limited access/survey

Table 2.1: Inclusion and Exclusion Criteria

- In step **3**, the remaining 237 articles were assessed, with 13 being excluded due to limited access. The final number of articles that were thoroughly screened was 224. Subsequently, each study was compared on performance parameters including the problem statement, the solution proposed, the dataset used, contributions, shortcomings, future work, DL models used to improve the accuracy, performance assessment of proposed models in the literature using overall accuracy, average accuracy, and kappa coefficient.

Based on the inclusion-exclusion criteria outlined in Table 2.1 and using preferred reporting items for systematic reviews and meta-analysis, 224 articles were listed. The articles retrieved using the keywords deep learning and hyperspectral image classification included empirical research and experimentation. The investigation and comparison of each article were made using the proposed deep learning model, datasets utilized, training

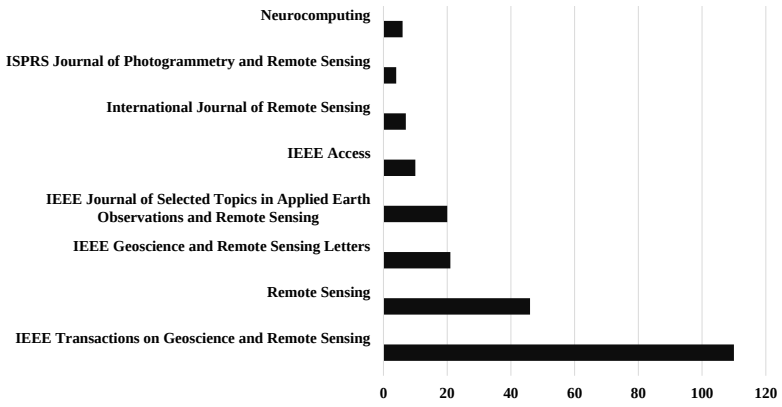


Figure 2.2: Number of articles included from respective journals

ratio, accuracy attained, contributions, shortcomings, and future work. For a quick analysis, Figure 2.2 shows the names of journals and available relevant papers in the respective journals.

2.1.2 Proposed Research Questions

The objective behind mainly focusing on HSI is that they contain hundreds or thousands of bands, making them difficult to interpret. Deep Learning gained popularity in remote sensing because it can hierarchically learn representative and discriminative features. As a result, the ultimate goal of this work is to look at existing deep learning models and their newly proposed variants for hyperspectral image classification, and problems existing in current literature, and analyze the accuracy associated with DL models. The objective is to find out the answer to the below research questions :

- **RQ1**. What are the most widely used HSI datasets and what performance metrics are used to assess the performance?
- **RQ2**. What is the performance of existing supervised, semi-supervised, and unsupervised DL models in classifying hyperspectral images?
- **RQ3**. Which DL model gives the highest classification accuracy for HSI classification?
- **RQ4**. What are the significant challenges encountered by existing researchers?

To look into these research questions closely, in forthcoming sections, we begin by looking at benchmark HSI datasets in section 2.1.3 and performance metrics in section 2.1.4. The next section 2.2 provides an elaborated literature on the use of various deep

learning models in HSIC. We will examine various models employed for HSIC including additions for model improvement, insights into training ratios, datasets used, and the results obtained. Each section is dedicated to a specific type of model, providing an in-depth look into their characteristics. Section 2.2.2 focuses on models built using CNN, Section 2.2.3 delves into models utilizing AE as feature extractors, Section 2.2.4 covers graph-based learning, Section 2.2.5 explores ensemble networks, Section 2.2.6 delves into generative adversarial networks, Section 2.2.7 focuses on long short-term memory networks and the final section, Section 2.2.8, encapsulates miscellaneous models, including attention networks, transformer networks, and few-shot learning, among others.

2.1.3 Benchmark HSI Datasets

This section provides an in-depth examination of commonly utilized hyperspectral datasets, highlighting their unique characteristics. Within the literature, prominent datasets include Indian Pines, Pavia University, Salinas, Pavia Center, and Kennedy Space Center. The distribution of these frequently employed hyperspectral datasets in the literature is depicted in Figure **??**.

Table 2.2: Brief of Hyperspectral Datasets

Dataset	Device	Year	Device Type	Place	Pixel Size	Classes	Samples
Indian Pines	AVIRIS	1992	Airborne	Indiana	145×145	16	10,249
Salinas	AVIRIS	1992	Airborne	California	512×217	16	54,129
KSC	AVIRIS	1996	Airborne	Florida	512 X 614	13	5,211
Pavia University	ROSIS	2001	Airborne	Northern Italy	610×340	9	42,776
Pavia Centre	ROSIS	2001	Airborne	Northern Italy	1096×492	9	7,456
Botswana	Hyperion	2001	Satellite	Okavango delta	1476×256	14	3,248
Houston University	CASI	2017	Airborne	Houston	601×2384	20	504,712

AVIRIS Mode – Airborne; Bands – 224; Range - 0.36-2.45; Width – 10; Ground Sample Distance: 20;
ROSIS - Mode – Airborne; Bands 115; Range - 0.43-0.86; Width – 4; Ground Sample Distance: 1.3;
Hyperion Mode - Satellite; Bands - 220; Range - 0.40-2.50; Width – 10; Ground Sample Distance: 30;
CASI Mode - Airborne ; Bands - 144; Range - 0.36-1.05; Width - 2.4; Ground Sample Distance: 2.5

- **Indian Pines (IP)**: This dataset was acquired using the AVIRIS sensor above the Indian Pines test site located in Northwestern Indiana. It comprises a grid of 145x145 pixels and encompasses 224 spectral reflectance bands within the wavelength range of 0.4–2.5×10^{-6} meters. The provided ground truth is categorized into sixteen classes and the bands were later reduced to 200.

- **Pavia University (PU) and Pavia Centre (PC)**: These two datasets were captured by the ROSIS sensor during an aerial survey conducted over Pavia in northern Italy. The

Table 2.3: Sample Distribution for IP

No	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-Trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

Pavia Centre dataset comprises 102 spectral bands, while the Pavia University dataset has 103. The Pavia Centre image is 1096×1096 pixels, and the Pavia University image is 610×610 pixels. The geometric resolution is 1.3 meters, and both images' ground truths are classified into nine distinct classes each.

Table 2.4: Sample Distribution for PU

No	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted Metal Sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

Table 2.5: Sample Distribution for PC

No	Class	Samples
1	Water	824
2	Trees	820
3	Asphalt	816
4	Self-Blocking Bricks	808
5	Bitumen	808
6	Tiles	1260
7	Shadows	476
8	Meadows	824
9	Bare-Soil	820

- **Salinas (SA) and Salinas-A (SA-A)**: The imagery was acquired using the 224-band AVIRIS sensor over Salinas Valley, California, having a high spatial resolution of 3.7-meter pixels. The covered area spans 512×217 samples. The ground truth for the Salinas dataset comprises 16 distinct classes. A subsection of the Salinas image, referred to as Salinas-A, is commonly utilized. This subscene encompasses dimensions of 86×83 pixels and is annotated with information from six distinct classes.

- **Kennedy Space Centre (KSC)** : The Kennedy Space Center dataset involves the classification of wetland vegetation at the Kennedy Space Center in Florida through the utilization of hyperspectral imagery, containing 13 classes.

Table 2.6: Sample Distribution for SA

No	Class	Samples
1	Brocoli green weeds 1	2009
2	Brocoli green weeds 2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807

Table 2.7: Sample Distribution for SA-A

No	Class	Samples
1	Brocoli_green_weeds_1	391
2	Corn_senesced_green_weeds	1343
3	Lettuce_romaine_4wk	616
4	Lettuce_romaine_5wk	1525
5	Lettuce_romaine_6wk	674
6	Lettuce_romaine_7wk	799

Table 2.8: Distribution for KSC

No	Class	Samples
1	Scrub	347
2	Willow Swamp	243
3	CP Hammock	256
4	Slash Pine	252
5	Fallow_smooth	2678
6	Oak	161
7	Hardwood	229
8	Swamp	105
9	Graminoid Marsh	390
10	Spartina Marsh	520
11	Cattail Marsh	404
12	Salt Marsh	419
13	Mud Flash	503
14	Water	927

2.1.4 Performance Metrics

It has been found in our analysis of the existing works that Overall Accuracy (OA), Average Accuracy (Average Accuracy (AA)), and Kappa Coefficient (Kappa Coefficient (KC)) are the three commonly used evaluation metrics to evaluate and comparatively analyze the performance of the proposed model with the state-of-the-art models. These metrics are described in details as below :

- **Overall Accuracy**: The OA metric quantifies the percentage of accurately classified pixels within the complete HSI dataset, such that $OA = \frac{C'}{C}$, where C' is the total number of samples classified correctly. Therefore, the OA offers a comprehensive evaluation of the model's classification performance across all the classes and can be computed using the equation 2.1 where True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) represent the abbreviations for true positives, true negatives, false positives, and false negatives, respectively.

$$\text{OA} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{2.1}$$

- **Average Accuracy**: The AA, also known as mean class accuracy, determines the average classification accuracy for each specific class within the HSI dataset. By calculating the average accuracy per class, this metric offers valuable insights into the model's performance at a class-specific level, enabling a more detailed evaluation of its classification capabilities. AA can be computed using equation 2.2, detailed as follows:

$$\text{AA} = \frac{A_1 + A_2 + A_3 + \ldots + A_n}{n} \tag{2.2}$$

- **Kappa Coffecient**: The KC, also known as Cohen's kappa, is a statistical metric used to evaluate the agreement between predicted and true class labels while considering the possibility of the agreement by chance alone. This measure takes into account both the classification accuracy and the potential agreement that could occur randomly and can be computed using equation Equation 2.3, given below.

$$\text{KC} = \frac{OA - P_c}{1 - P_c} \tag{2.3}$$

where P_c denotes the speculative probability of possibility and can be calculated using Equation 2.4:

$$P_c = \frac{(TP + FP)(TP + FN) + (FN + TN)(FP + TN))}{(TP + FP + TN + FN)^2} \tag{2.4}$$

These metrics were used to evaluate the performance of various deep learning models on benchmark HSI datasets. The performance evaluation of various studies is presented in the next section 2.2.

2.2 Overview of Deep Learning Approaches for HSIC

Deep learning algorithms have been at the forefront of the evolution of HSIC, addressing the need for more accurate and efficient methods [62]. Therefore, this section delves into a detailed examination of the diverse deep-learning approaches that have emerged in the pursuit of more robust and precise HSIC outcomes. Figure 2.3 depicts the data showing the number of articles published corresponding to different types of DL models and the year when the study was published.

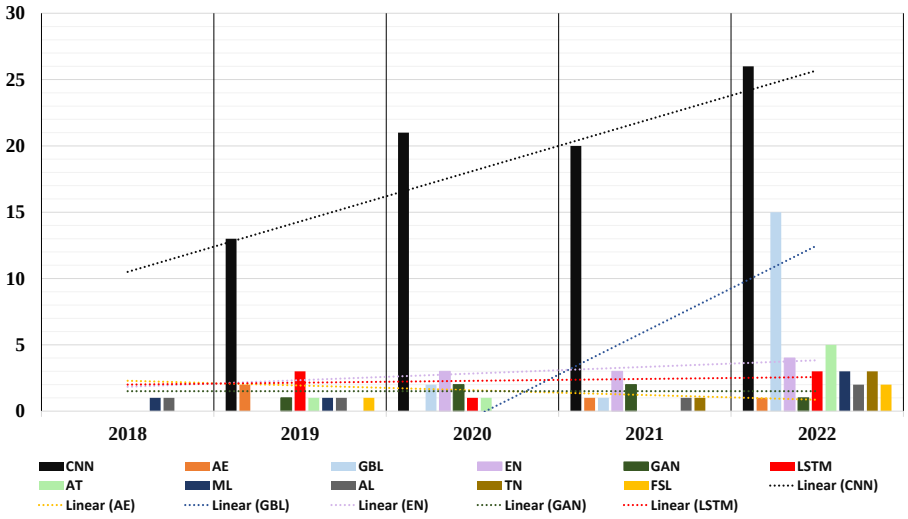


Figure 2.3: Trend Analysis for publications respective to DL models and corresponding year

2.2.1 Criteria for Selection

The research studies were selected based on various criteria:

- **Performance on benchmark datasets:** This chapter highights the widespread use of benchmark HSI datasets such as IP, PU, SA, KSC for evaluating model performance. Works that demonstrate competitive or leading performance on one or more of these datasets were selected for comparison, as they represent relevant benchmarks for the proposed models.
- **Methodological relevance:** Works that employ similar DL architectures or techniques as the proposed models, such as CNN, AE, GCN, or GAN could be chosen

for comparison. This methodological alignment facilitates a meaningful analysis of the strengths, weaknesses, and unique contributions of the proposed models relative to existing approaches.

- **Recency and impact:** More recent works, particularly those published in high-impact journals or conferences in the field of remote sensing or HSI analysis, are prioritized for comparison. These works represent the cutting-edge developments and state-of-the-art approaches in the field, providing a relevant benchmark for assessing the contributions of the proposed models.

- **Diversity of approaches:** To provide a comprehensive comparison, works representing diverse methodological approaches (e.g., supervised, semi-supervised, unsupervised) or focusing on different aspects of HSIC (e.g., data augmentation, transfer learning, attention mechanisms) are selected. This diversity ensures a well-rounded analysis and highlights the unique strengths and contributions of the proposed models in relation to various existing approaches.

By considering a combination of these criteria, the selection of state-of-the-art works for comparison are tailored to effectively position the proposed models within the current research landscape, highlight their novelty, and demonstrate their potential contributions to advancing the field of HSIC.

2.2.2 Review of studies based on CNN

The investigation into Convolutional Neural Networks (CNN) within the domain of Hyperspectral Image (HSI) stratification has spawned a multitude of innovative methodologies. This section presents a meticulous literature review that scrutinizes the diverse spectrum of deep learning models having CNN as the base model. The analysis touches upon the additional components integrated to augment model performance, the datasets employed, achieved accuracies in terms of OA, AA, KC, and the distinctive contributions and limitations of each study.

Tables 2.9, 2.10, 2.11, and 2.12 serve as compendiums of case studies leveraging CNN as the foundational model for classification, elucidating proposed enhancements, and documenting efforts to improve classification accuracy. Each table provides insights into the strengths and limitations of the respective studies. Evaluation metrics, including OA, AA, KC form the basis for assessing the efficacy of each proposed model on specific datasets.

Table 2.9: Articles employing CNN as the base model for classification (Year: 2022)

S. No	Ref	Model	Addition	Training Ratio	Datasets and Results	Contributions	Shortcomings
1	[63]	CNN	3-D Gabor Filters	**PU:** 0.20 **HU:** 0.20	**Pavia U:** OA = 94.78; AA = 96.42; KC = 93; **Houston U:** OA = 74.37; AA = 87.15; KA = 69	1. 3-D Gabor wavelet is better than 2D for feature extraction. 2. 3-D Gabor Filters reduce parameters, making the network lightweight.	Parameters and time can be further reduced for optimization.
2	[64]	CNN	3-D-ANAS: 3-D Asymmetric Neural Architecture Search	**PU:**0.006 **PC:** 0.04	**Pavia U:** OA = 97.92; AA = 98.62; KC = 97.25; **Pavia C:** OA = 99.5; AA = 98.74; KC = 99.3	1. Resolved patch pixel classification duplication. 2. Improved interference speed.	1. Augmentation strategy used for increased accuracy. 2. More effective for datasets with additional spectral bands and categories.
3	[65]	CNN	Incremental PCA	**PU:** 0.6 **IP:** 0.6 **SA:** 0.6	**Pavia U:** OA = 98.4; AA = 97.89; KC = 97.89; **Indian Pines:** OA = 97.75; AA = 94.54; KC = 97.54; **Salinas:** OA = 98.06, AA = 98.8; KC = 97.85	1. Better accuracy. 2. Lesser computational and convergence time. 3. Use of incremental PCA to reduce band duplicity.	Too high number of training samples taken.
4	[66]	CNN	Depthwise Convolutions	**PU:** 0.01 **HU:** 0.2	**Pavia U:** OA = 97.78; AA = 95.81; KC = 96.79; **Houston U:** OA = 87.17; AA = 88.94; KC = 86.09	1. Alternative to spatial convolutions. 2. Lightweight network with fewer parameters.	Longer processing time for convergence. Better classification with dimensionality reduction methods.
5	[67]	CNN	Superpixel Segmentation or Undirected Graph Clustering	**PU:** 0.01 **IP:** 0.07 **SA:**0.001 **HU:** 0.05	**Pavia U:** OA = 99.86; AA = 99.77; KC = 99.64; **Indian Pines:** OA = 98.13; AA = 99.01; KC = 97.85; **Salinas:** OA: 99.31, AA = 99.27; KC = 99.23; **Houston U:** OA = 97.52; AA = 97.89; KC = 97.32	1. Accurate classification of misclassified classes. 2. Small training time due to small patch size.	1. Relies on generated features for classification. 2. Time cost increases with dataset size.
6	[68]	CNN	Step Activation Quantization Acceleration	**PC:** 0.24 **SA:** 0.06	**Pavia C:** OA = 99.12; AA = 97.84; **Salinas:** OA = 93.32; AA = 96.02	1. Computationally less exhaustive. 2. Use of quantization bits for balanced accuracy and speed.	Classification accuracy could be improved.
7	[69]	CNN	Adaptive Hash Attention Mechanism and a Lower Triangular Network	**IP:** 0.05 **PU:** 0.05 **BOT:**0.05	**Indian Pines:** OA = 97.86; AA = 97.89, KC = 97.56; **Pavia U:** OA = 99.64; AA = 99.51; KC = 99.52; **Salinas:** OA = 99.97; AA = 99.95; KC = 99.96; **Botswana:** OA = 98.9; AA = 98.86; KC = 98.8	1. Works with imbalanced datasets. 2. Skip connections aid feature reuse and discrimination. 3. Reduces overfitting with band padding and dropout.	Use of joint attention mechanism for spectral and spatial attention.

8	[70]	CNN	Spatial Pyramid Pooling	**PU:** 0.004 **SA:** 0.005 **HU:**0.002	**Pavia U:** OA = 95.6; AA = 94.5; KC = 94.2; **Salinas:** OA = 99.7; AA = 99.7; KC = 99.7; **Houston University:** OA = 89.2; AA = 89.6; KC = 88.3;	1. Works well with limited samples. 2. Less time in training and efficient inference.	Spatial constraint method used for HIS image classification.
9	[71]	CNN	Feature Extractor	**PU:**0.001 **IP:**0.007 **SA:**0.001 **KSC:**0.01	**Pavia U:** OA = 82.9; AA = 92.7; KC = 78.7; **Indian Pines:** OA =75.8; AA = 85.9; KC = 72.7; **Salinas:** OA = 94.8; AA = 97; KC = 94.2; **KSC:** OA = 97.3; AA = 96.3; KC = 97	1. Effective for imbalanced class distributions. 2. Alleviates overfitting. 3. Uses ResSE for feature extraction. 4. Reduces bias and misclassification.	PLM alone cannot handle imbalanced class distribution.
10	[72]	CNN	Attention Mechanism + Graph Convolutional Network	**SA:**0.01 **PU:**0.01	**Salinas:** OA = 97.34; AA = 97.22; KC = 97.04; **Pavia U:** OA = 92.23; AA = 83.21; KC = 89.65	1. Addresses accuracy with less training data. 2. Better Feature Extraction for improved performance.	1. Attention module could be more targeted. 2. Synthetic data addition for small dataset. 3. Cross-domain training/testing could be explored.
11	[73]	CNN	Feature Extractor, Feature Fusion	**PU:**0.002 **IP:**0.015 **KSC:**0.024	**Pavia U:** OA = 98.93; AA = 99.24; KC = 98.59; **Indian Pines:** OA = 96.3; AA = 97.93; KC = 95.68; **KSC:** OA = 97.26; AA = 95.6; KC = 96.71	1. Improves classification performance by adding unsupervised samples to a few supervised samples. 2. Shared Feature Extractor. 3. Good for a limited and noisy sample problem.	1. Moderate Run Time. 2. Unlabeled data played an important role in improving the classification performance. 3. Accuracy decreases with a high increase in cluster numbers or a high increase in spatial size.
12	[74]	CNN	PCA + Augmentation	**IP:** 0.01 **PU:** 0.001	**Indian Pines:** OA = 83.68; AA = 81.08; KC = 81.25; **Pavia U:** OA = 98.25; AA = 96.98; KC = 97.68	1. Reduces computational complexity with lesser processing time. 2. PCA and Augmentation to deal with a small dataset. 3. Use of L2 regularization and dropout to alleviate overfitting.	1. Accuracy for Indian Pines could be increased. 2. Processing time improved after incorporating 2xNVIDIA GeForce RTX 2080 Ti, NVLink.
13	[75]	CNN	Attention Mechanism	**PU:**0.02 **IP:**0.10 **SA:**0.02	**Pavia U:** OA = 97.58; AA = 98.53; KC = 96.79; **Indian Pines:** OA = 96.55; AA = 96.62; KC = 96.07; **Salinas:** OA = 96.91; AA = 98.73; KC = 96.55	1. Alleviates the HSI rotation problem, Good classification performance. 2. Spectral attention module suppresses redundant bands. 3. Cross Spatial Attention generates feature rotation independent.	Shows poor performance near the edges of the Indian Pines dataset.

14	[76]	CNN	Attention Mechanism + Residual Network	**SA:**0.04 **PU:**0.06 **IP:**0.11	**Salinas:** OA = 91.26; AA = 94.22; KC = 88.5; **Pavia U:** OA = 93.33; AA = 89.61; KC = 92.3; **Indian Pines:** OA = 82.38; AA = 80.35; KC = 79.1	1. Performs pixel-to-pixel classification with attention modules. 2. Ability to discriminate features better. 3. Stable Classification performance.	Still has the scope of improving classification accuracy.
15	[77]	CNN	EMP + EPF extended morphological profiles, Edge preserving filtering	**HU:**0.10 **PU:** 0.10	**Houston U:** OA = 95.58; AA = 95.48; KC = 95.2; **Pavia U:** OA = 96.15; AA = 97.14; KC = 94.9	Use of mutual teaching and controlled random sampling to improve the classification performance.	Takes a long time because of multiple iterations.
16	[78]	CNN	Feature Fusion + Attention	**IP:** 0.1 **PU:** 0.1 **KSC:** 0.1	**Indian Pines:** OA = 85.76; AA = 81.08; KC = 83.76; **Pavia U:** OA = 94.88; AA = 93.02; KC = 93.21; **KSC:** OA = 91.76; AA = 87.56; KC = 90.82	1. Attention module to extract informative features. 2. Ultra-lightweight attention module to further enhance the feature extraction.	1. CNN misclassifying some classes. 2. Overall time costlier.
17	[79]	CNN	Spectral and Spatial Feature Extraction	**KSC:** 0.05 **PU:** 0.004 **SA:** 0.06	**KSC:** OA = 91.6; AA = 89.92; KC = 91; **Pavia U:** OA = 90.77; AA = 94.05; KC = 88; **Salinas:** OA = 88.8; AA = 93.52; KC = 88	1. Addresses the problems of noisy labels. 2. Reduces confusing features and increases classification of misclassified labels; works well with corrupted datasets too. 3. Robust Loss function for better convergence.	Needs a few perfect samples for the proposed network to work.
18	[80]	CNN	Feature Transformer, Attentive Transformer	**IP:** 0.10 **PU:** 0.04 **SA:** 0.06	**Indian Pines:** OA = 96.36; AA = 97.77; KC = 96; **Pavia U:** OA = 97.86; AA =96.28; KC = 96; **Salinas:** OA = 98.36; AA = 98.93; KC = 98	Applies unsupervised pre-training and supervised classification.	Higher Computational complexity.
19	[81]	CNN	Attention Mechanism + Features Grouped Network (FGN)	**IP:** 0.10 **PU:** 0.05 **KSC:** 0.10	**Indian Pines:** OA = 99.79; AA = 99.77; KC = 99.76; **Pavia U:** OA = 99.96; AA = 99.97; KC = 99.94; **KSC:** OA = 99.96; AA = 99.94; KC = 99.96	1. Attention mechanism to generate the discriminative features. 2. Feature Group Network to group the features sequentially according to spectral dimension. 3. Good classification performance.	Does not deal with the sample scarcity problem.
20	[82]	CNN	Attention Mechanism	**PU:** 0.01 **IP:** 0.05 **PC:** 0.02	**Pavia U:** OA = 97.4; KC = 96.5; **Indian Pines:** OA = 97.2; KC = 96.9; **Pavia C:** OA = 99.7; KC = 99.4	1. Identifies the feature associations with the homogeneous mask to reduce heavy computations. 2. Homogeneous Attention Network to extract discriminative features in mixed pixels.	Causes heavy computational load without the homogeneous mask.

21	[83]	CNN	Attention Mechanism + Multiclass Focal Loss + Depth-wise Convolutions	**IP**: 0.05 **PU**: 0.01 **SA**: 0.01	**Indian Pines:** OA = 96.51; AA = 93.3; KC = 96.02; **Pavia U:** OA = 96.48; AA = 94.25; KC = 95.34; **Salinas:** OA = 97.18; AA = 98.44; KC = 96.87	1. Use of spatial-spectral attention mechanism with depth-wise separable convolutions to make the network lightweight **2. Decreased complexity and high classification accuracy**	1. More focused on spatial feature extraction, the Use of spectral information could enhance the performance **2. Attention block works better when the number of samples is severely limited.**
22	[84]	CNN	3-D Squeeze and-Excitation residual (3DSERes) blocks	**IP:** 0.07 **PU:** 0.10	**Indian Pines:** OA = 93.99; AA = 94.55; KC = 93.09; **Pavia U:** OA = 91.48; AA = 93.97; KSC = 88.81	1. Constructs multi-dimensional samples in nine directions to alleviate the misclassification problem 2. Use of 3D Squeeze and Excitation blocks to explore spectral and spatial information for extracting discriminative features	1. Due to multi-direction patches involved, it takes a long computational time with satisfactory classification performance
23	[85]	CNN	Capsule Networks + Attention Mechanism	**KSC:** 0.05 **IP:** 0.10 **PU:** 0.05	**KSC:** OA = 97.73; AA = 95.71; KC = 97.47; **Indian Pines:** OA = 92.27; AA = 82.4; KC = 91.21; **Pavia U:** OA = 95.69; AA = 94.32; KC = 94.19; **Houston University:** OA = 89.05; AA = 90.78; KC = 88.11	1. Better feature discrimination 2. Reduced computational load with primary capsule network 3. Reliable network with a lesser number of parameters with a capacity to work with limited data	1. Does not work well for imbalanced class distribution
24	[86]	CNN	Dilated Convolutions + Self-Supervised Contextual Feature Learning	**IP:** 0.008 **SA:** 0.001 **PU:** 0.001**HU:** 0.005	**Indian Pines:** OA = 82.69; AA = 88.75; KC = 80.37; **Salinas:** OA = 94.73; AA = 97.08; KC = 94.14; **Pavia U:** OA = 85.06; AA = 83.48; KC = 79.8; **Houston U:** OA = 80.89; AA = 83.56; KC = 79.39	1. Fewer floating-point operations per second 2. Faster computational speed 3. The proposed feature extraction network is made up of multiple plug-and-play dilated convolutional blocks	1. Increasing the number of multiscale patches increases the overall accuracy with an increase in floating point operations, making the network heavier
25	[87]	CNN	Spatial Spectral Similarity Measurement + Knowledge Distillation	**IP:** 0.02 **PU:** 0.002	**Indian Pines:** OA = 84.3; AA = 87.8; KC = 79.9; **Pavia U:** OA = 86.8; AA = 88.8; KC = 83.4	1. Similarity measurement using spectral and spatial distance with 3D transformation and adaptive soft label that labels the unlabeled samples 2. Takes similar time to run as the state-of-the-art models under the same environment	1. Works well with limited samples but still has much scope for improvement in improving the classification accuracy

26	[88]	CNN	Residual Network + Stochastic Depth Training	**IP:** 0.20 **PU:** 0.10 **SA:** 0.10	**Indian Pines:** OA = 75.11; AA = 80.94; KC = 71.56; **Pavia U:** OA = 91.56; AA = 93.13; KC = 89.02; **Salinas:** OA = 92.53; AA = 95.96; KC = 91.67	1. Effective feature extraction using spectral and spatial modules 2. Stochastic depth training to improve the efficiency of training	1. Time Costlier 2. Low classification accuracy for the Indian Pines dataset

Table 2.10: Articles employing CNN as the base model for classification (Year: 2021)

S. No	Ref	Model	Addition	Training Ratio	Datasets and Results	Contributions	Shortcomings
27	[89]	CNN	Residual Learning + Involution block	**PU:** 0.10 **HU:** 0.18 **SA:** 0.01	**Pavia U**: OA = 96.4; AA = 95.8; KC = 95.2; **Houston U:** OA = 86.5; AA = 88.6; KC = 85.4; **Salinas:** OA = 96.7; AA = 98.6; KC = 96.3	1. Able to capture long-range spatial interactions 2. Good classification accuracy (3.5% higher) with lesser parameters (3.23×) with the use of involution block in comparison to nonuse of involution block	1. Much room for improvement in classification accuracy, especially for Hyrank and Houston University dataset
28	[90]	CNN	Attention Mechanism	**IP:** 0.03 **KSC:** 0.03 **SA:** 0.05	**Indian Pines**: OA = 97.37; AA = 97.13; KC = 97; **KSC**: OA = 98.64; AA = 97.74; KC = 98.48; **Salinas**: OA = 98.35; AA = 98.87; KC = 98.17	1. Feature extraction module to extract discriminative features 2. Suitable for small sample training 3. Spectral and spatial attention module help in increasing the classification performance.	1. High computational efficiency 2. Ensuring Time efficiency
29	[91]	CNN	Spatial–Spectral Schroedinger eigenmaps + Dual Channel Convolution	**IP:** 0.007 **PU:** 0.001 **SA:** 0.001	**Indian Pines**: OA = 74.78; AA = 84.85; KC = 72; **Pavia U**: OA = 82.3; AA = 87.27; KC = 78; **Salinas**: OA = 88.61; AA = 93.77; KC = 87	1. Spatial–Spectral Schroedinger eigenmaps to reduce the parameters 2. Dual Channel Convolution and Bi channel fusion to extract discriminative features 3. Works well with smaller datasets	1. Scope of improving classification performance
30	[92]	CNN	Wide sliding window and subsampling network	**PU:** 0.20 **KSC:** 0.20 **SA:** 0.20	**Pavia U**: OA = 99.19; AA = 98.51; KC = 98.93; **KSC**: OA = 99.87; AA = 99.71; KC = 99.86; **Salinas**: OA = 99.67; AA = 99.63; KC = 99.63	1. Wide sliding windows help to learn the higher level discriminative features 2. Alleviates overfitting and reduces the computational load 3. Great classification accuracy	1. Identifying the ideal patch size is a challenge

31	[93]	CNN	Convolutional Encoder (CAE)	**IP:** 0.10 **PU:** 0.10	**Indian Pines**: OA = 96.17; AA = 95.29; KC = 95.63; **Pavia U**: OA = 98.65; AA = 98.01; KC = 98.21	1. The proposed 3D CAE is piled up with convolutional and deconvolutional layers that help in feature extraction 2. Training can be done in an unsupervised manner	1. Causes misclassification in a few classes 2. Encoded layers slightly affect the value of OA and K
32	[94]	CNN	Adaptive Routing + Capsule Network with powered activation regularization	**PU:** 0.27 **SA:** 0.30	**Pavia U** : OA = 99.51; KC = 99; **Salinas**: OS =94.52; KC = 93	1. Powered activation regularization to extract discriminative features 2. Alleviates overfitting and gradient vanishing 3. Significantly reduced time in comparison to Capsule network	1. Slight decrease in OA with deeper architecture 2. Higher computational efficiency in comparison to capsule network
33	[95]	CNN	Attention Mechanism	**IP:** 0.10 **PU:** 0.02 **SA:** 0.02	**Indian Pines**: OA = 98.1; AA = 96.16; KC =97.84; **Pavia U**: OA = 98.97; AA = 98.32; KC = 97.64; **Salinas**: OA = 98.18; AA = 98.76; KC = 97.97	1. Center attention module to extract discriminative features 2. Reduced parameters in the network 3. Increased computational efficiency	1. Misclassified samples, especially at the boundary
34	[96]	CNN	Particle Swarm Optimization + SuperNet	**PU:** 0.004 **SA:** 0.004 **IP:** 0.01 **KSC:** 0.04	**Pavia U** : OA =93.36; AA = 88.39; KC = 92.32; **Salinas**: OA = 96.64; AA = 97.32; KC = 96.23; **Indian Pines** : OA = 89.32; AA = 74.42; KC =87.58; **KSC** : OA = 97.56; AA = 96.41; KC = 97.28	1. Use of particle swarm optimization to obtain the optimal architecture and iterations 2. Reduced search time with Super Net 3. Better convergence with reduced complexity in comparison to state-of-the-art handcrafted methods	1. Classification accuracy can be improved for the Indian Pines dataset
35	[52]	CNN	Capsule Network + Octave convolution	**DFC13:** 0.01 **DFC14**: 0.01	**GRSS DFC 2013** : OA = 99.59; AA = 99.62; KC = 99.56; **GRSS DFC 2014** : OA = 99.37; AA = 95; KC = 96.7	1. Octave convolution to reduce the larger number of parameters in the capsule network and high memory resource consumption involved 2. Enhanced feature discrimination capability 3. Alleviates overfitting	1. PCA did not affect the performance of the method 2. It could have been applied to benchmark hyperspectral datasets too
36	[49]	CNN	Siamese Network + Extended Morphological Profiles + Spectral Spatial Fusion	**PU:** 0.001 **KSC:** 0.01	**Pavia U** : OA = 85.81; AA = 85.44; KC = 81.36; **KSC** : OA = 90.8; AA = 89.63; KC = 89.75	1. Dual path-based processing with similarity learning 2. Adversarial training and augmentation for improving the classification performance	1. Average Accuracy of the Pavia U dataset is slightly lower

37	[37]	CNN	Transfer Learning + Super pixel Pooling	**IP:** 0.03 **PU:** 0.004 **SA:** 0.006	**Indian Pines** : OA = 94.45; AA = 96.43; KC = 93.44; **Pavia U** : OA = 93.18; AA = 93.78; KC = 92.36; **Salinas** : OA = 95.99; AA = 95.97; KC = 95.46	1. Superpixel pooling to deal with the problem of limited samples 2. Upsampling and Down sampling to help preserve the spatial and spectral features 3. Transfer Learning introduced in the model shortens the training time.	1. Misclassification of some classes like Trees and Shadows
38	[97]	CNN	Attention Mechanism + Residual Network	**PU:** 0.10 **KSC:** 0.20 **IP:** 0.20	**Pavia U** : OA = 99.82; AA = 99.59; KC = 99.71; **KSC** : OA = 99.81; AA = 99.74; KC = 99.52; **Indian Pines** : OA = 99.37; AA = 99.45; KC =99.61	1. Smooth and fit classification, no salt and pepper noise 2. Fast convergence 3. Good classification performance	1. Tuning the learning rate parameter has a low impact on the Pavia U dataset and more impact on the Indian Pines dataset 2. Misclassification in Indian Pines dataset for a few classes
39	[98]	CNN	Two streams Residual Network	**IP:** 0.10 **PU:** 0.10 **KSC:** 0.10	**Indian Pines** : OA = 98.7; AA = 98.71; KC = 98.52; **Pavia U** : OA = 99.86; AA = 99.77; KC = 99.82; **KSC** : OA = 99.48; AA = 99.04; KC = 99.42	1. Reduced parameters with skip connections added to the network 2. Works well even for imbalanced class distribution	1. Second best overall accuracy in comparison to the state-of-the-art models 2. Deeper network leads to overfitting
40	[99]	CNN	Dilated convolutions	**IP:** 0.10 **PU:** 0.01 **SA:** 0.01	**Indian Pines** : OA = 98.95; AA = 98.49; KC = 98.81; **Pavia U** : OA = 99.16, AA = 98.72, KC = 98.89; **Salinas** : OA = 99.41; AA = 99.45; KC = 99.34	1. Dilated Convolution kernels to make the network lightweight 2. Efficient feature extraction and alleviation of overfitting using a multiscale residual network to generate informative features with lower computational cost	1. Determining the perfect spatial size to increase the classification performance
41	[100]	CNN	Attention Mechanism + Residual Network	**PC:** 0.008 **HU:** 0.17	**Pavia C** : OA = 99.8; AA = 99.29; KC = 99.72; **Houston U** : OA = 83.61; AA = 79.1; KC = 79.27	1. Extracting multiscale features at a granular level, Attention network to further improve discriminability 2. Double branch structure with different convolutional kernels	1. Low Accuracy achieved for the Houston U dataset
42	[101]	CNN	Mixed Link Blocks + Dual-Path Architecture	**IP:** 0.07 **PU:** 0.09 **HU:** 0.2	**Indian Pines** : OA = 97.2; AA = 98.32; KC = 96.79; **Pavia U** : OA = 96.12; AA = 95.9; KC = 94.78; **Houston U** : OA = 86.43; AA = 88.54; KC = 85.32	1. Feature usage and feature identification that helps in reducing the time taken in duplicate features learning 2. Improved flow of information with shifted additions	1. Low accuracy achieved for the Houston U dataset 2. More parameters involved in the network compared to the state-of-the-art models

43	[102]	CNN	Ghost Module + Ghost Bottleneck	**IP:** 0.03 **PU:** 0.05 **HU:** 0.10	**Indian Pines** : OA = 88.31; AA = 78.77; KC = 86.7; **Pavia U** : OA = 92.83; AA = 91.37; KC = 90.2; **Houston U** : OA = 87.87; AA = 89.35; KC = 86.84	1. A less computer-intensive and memory demanding network in comparison to traditional CNN 2. Computationally efficient classification method	1. Classification accuracy could be better for Indian Pines and Houston University dataset 2. Takes more epochs
44	[103]	CNN	Similarity Learning + Maximum Margin Ranking Loss	**IP:** 0.07 **SA:** 0.001 **PU:** 0.001	**Indian Pines** : OA = 96.02; AA = 94.49; KC =95.46; **Salinas** : OA = 98.27; AA = 98.67; KC = 98.08; **Pavia U** : OA = 99.13; AA = 98.75, KC = 98.85	1. Deep similarity network handles the scarcely available samples 2. To gain hold of discriminative features, maximum margin ranking loss and cross-entropy loss are introduced 3. Two branch networks to enhance the classification	1. Misclassification of a few classes in the Indian Pines dataset 2. Takes more training time than traditional CNN
45	[104]	CNN	Receptive Field + Selective Kernel Networks	**IP:** 0.03 **PU:** 0.03 **HU:** 0.03	**Indian Pines** : OA = 81.73; AA = 71.4; KC = 79.2; **Pavia U** : OA = 90.66; AA =88.09; KC = 87.34; **Houston U** : OA = 88.28; AA = 88.87; KC = 87.28	1. A non-linear attention mechanism included that combines information 2. Better connected regions in output with clear boundaries	1. Classification accuracy can be improved
46	[105]	CNN	Residual Network + Dilated Convolutions + Multiple Spectral Resolution	**BOT:** 0.10 **IP:** 0.10 **PU:** 0.10	**Botswana** : OA = 98.8; AA= 98.89; KC =98.7; **Indian Pines** : OA = 98.1; AA = 96.4; KC = 97.84; **Pavia U** : OA = 99.96; AA = 99.13; KC =99.96	1. More dimension paid to the spectral module 2. Better spectral information extraction and analysis 3. Better performance under small patch size	1. Lack of spectral information 2. Incorporating CNN gives second-best results for a few datasets 3. Insensitive to the change in patch size

Table 2.11: Articles employing CNN as the base model for classification (Year: 2020)

S. No	Ref	Model	Addition	Training Ratio	Datasets and Results	Contributions	Shortcomings
47	[30]	CNN	Siamese Network + Spatial-Spectral Pyramid Pooling + Transfer learning	**IP:** 0.19 **PU:** 0.04 **SA:** 0.05	**Indian Pines :** OA = 99.17; KC = 99; **Pavia U :** OA = 99.52; KC = 99.3; **Salinas :** OA = 98.15; KC = 97.9	1. Extraction of fixed-length multi-level 3D features using pyramid pooling 2. Siamese network with the capability of exploring both spectral and spatial features with varying 3D samples or scales 3. Use of transfer learning across datasets	1. The use of transfer learning is challenging when the dataset trained is high resolution and the test is a low resolution 2. Time costlier

48	[106] CNN	Cubic Convolution	**IP:** 0.16 **PU:** 0.04 **SA:** 0.06	**Indian Pines** : OA = 99.4; AA = 99.34; KC = 99.27; **Pavia U** : OA = 99.88; AA = 99.7; KC = 99.53; **Salinas**: OA = 98.93; AA = 99.4; KC = 98.8; **Botswana** : OA = 99.67; AA = 99.71; KC = 99.63	1. PCA to reduce dimensions and 1DCNN to remove redundant bands 2. Convolutions are generated on three different sides for more flexibility in updating parameters 3. Smaller kernel size than 3DCNN for good classification accuracy and less training time	1. Noise remains at the edges 2. Convergence speed can be accelerated
49	[107] CNN	Attention mechanism + Squeeze-and-Excitation Network (SEN) + Residual Network	**PU:** 0.1 **IP:** 0.20 **KSC:** 0.20	**Pavia U** : OA = 99.89; AA = 99.84; KC = 99.86; **Indian Pines** : OA = 99.72; AA = 99.56; KC = 99.69; **KSC**: OA = 99.72; AA = 99.56; KC = 99.69	1. Residual Network to deal with the decreasing accuracy and SEN to expose the dependencies within spectral information 2. Feature fusion to join the feature cubes	1. Computationally expensive
50	[108] CNN	Dense Network + Early Exit Strategy + Adaptive Spectral unmixing	**IP:** 0.05 **SA:** 0.02 **KSC:** 0.01 **PU:** 0.01	**Indian Pines** : OA = 96.34; AA =96.56; KC =95.83; **Salinas** : OA = 98.92; AA = 99.27; KC = 98.8; **KSC** : OA = 92.82; AA = 89.08; KC = 93.09; **Pavia U** : OA = 98.64; AA = 97.97; KC = 98.2	1. An end-to-end network based on CNN and an early exit strategy for reduction in resources 2. Enhanced feature learning capability with fewer 3D convolutions and 2D convolutions, obtaining the feature maps	1. Did not perform well on Houston Dataset due to imbalanced class distribution 2. Augmentation or adversarial networks can be used to deal with the problem of limited data.
51	[109] CNN	Attention mechanism + Octave Convolution	**IP:** 0.3 **PU:** 0.4	**Indian Pines** : OA = 99.68; AA = 99.45; KC = 99.62; **Pavia U** : OA = 99.76; AA = 99.66; KC = 99.67	1. Octave Convolution to capture diverse features and reduce redundant bands 2. Attention network to explore discriminative features	1. Use of semi-supervised training to work on the problem of limited datasets
52	[110] CNN	Multilayer Fusion + Spatial attention mechanism	**IP:** 0.08 **PU:** 0.08 **SA:** 0.08 **HU**: 0.08	**Indian Pines** : OA = 97.38; AA = 98.42; KC = 96.86; **Pavia U** : OA = 99.17; AA = 99.33; KC = 98.89; **Salinas** : OA = 98.55; AA = 99.41; KC = 98.38; **Houston U** : OA = 87.89; AA = 92.23; KC = 86.04	1. Spectral and Spatial feature maps explored with 3D and 2D CNN 2. Multilayer fusion to use complementary features 3. No misclassification of classes identified wrong by other state-of-the-art networks	1. Consumes moderate training time on Pavia U dataset but more on Salinas and Houston University 2. More network parameters and floating point operations involved

53	[111] CNN	Squeeze-and Excitation (SE)	**IP:** 0.06 **PU:** 0.01 **SA:** 0.01	**Indian Pines** : OA = 98.09; AA = 96.75; KC = 96.26; **Pavia U** : OA = 99.05; AA = 98.82; KC = 98.43; **Salinas** : OA = 99.63; AA = 99.09; KC = 98.99	1. An end-to-end network that better extracts small patches in spectral, local, and global spatial information parallelly based on inter-channel connections and SE. 2. Combination of shallow and deep networks to find the optimum content 3. Computationally efficient	1. Overall accuracy decreases as the network gets deeper, causing overfitting
54	[112] CNN	Dilated Convolution	**IP:** 0.10 **PU:** 0.05	**Indian Pines** : OA = 97.61; AA = 96.31; KC = 97.27; **Pavia U** : OA = 98.43; AA = 98.17; KC = 97.9	1. Dilated convolution to overcome the challenge of resolution loss 2. An end-to-end network	1. Removing convolutional layers may lack representative ability, and increasing the number of neurons may cause overfitting
55	[113] CNN	Residual network + Dense connections	**IP:** 0.15 **PU:** 0.15 **SA:** 0.15 **PC:** 0.15	**Indian Pines** : OA = 98.74; AA = 98.73; KC = 98.25; **Pavia U** : OA = 99.62; AA = 99.53; KC = 99.5; **Salinas** : OA = 99.69; AA = 99.61; KC = 99.66; **Pavia C** : OA = 99.94; AA = 99.83; KC = 99.89	1. Dual branch residual network to learn the spectral and spatial features 2. Dense connections between neurons for better correlations 3. Residual structure to alleviate the problem of overfitting	1. Labelled samples for training could be replaced by unlabeled samples to further work on the problem of shortage of data.
56	[114] CNN	Feature Extraction + SVM	**IP:** 0.10 **PU:** 0.05	**Indian Pines** : OA = 95.58; AA = 95; KC = 94.97; **Pavia U** : OA = 98.48; AA = 97.56; KC = 97.98	1. Hierarchically built network that works with smaller samples. 2. Adopts raw images and applies feature extraction and classification	1. Proposed network could be further improved by adding dimensionality reduction and optimization techniques
57	[115] CNN	GAN + Collaborative Learning + Hard Attention Module + Conv LSTM	**IP:** 0.05 **PU:** 0.05	**Indian Pines** : OA = 97.4; AA = 95.2; KC = 97; **Pavia U** : OA = 99.2; AA = 98.6; KC = 99.2	1. Optimized GAN to alleviate the issue of overfitting 2. Joint attention module to better generate the discriminative spectral, spatial features 3. A convolutional LSTM layer in discriminator to extract joint spatial-spectral information	1. More principal-components lead to increased computational complexity and longer time to train.

58	[116] CNN	Transfer Learning	**SA:** 0.06 **PU:** 0.04 **KSC:** 0.5 **IP:** 0.31	**Salinas** : OA = 94.7; AA = 92.37; KC = 93.62; **Pavia U** : OA = 94.25; AA = 87.86; KC = 92.36; **KSC** : OA = 98.31; AA = 97.47; KC = 98.12; **Indian Pines** : OA = 90.86; AA =74.78; KC = 89.05	1. Training and Testing between two heterogeneous datasets to solve the limited sample availability problem 2. Attention mechanism for the feature maps reweighting and increasing classification accuracy 3. Central Kernel Alignment for similarity measurements 4. Consumes less training time and good accuracy	1. Taking ResNet as a pre-trained model gives bad results on the Pavia U dataset
59	[38] CNN	Transfer learning	**BOT:** 0.15 **HU:** 0.10	**Botswana** : OA = 99.65; AA =99.67; KC = 99.62; **Houston U**: OA = 99.45; AA= 99.4; KC = 99.35	1. Reduced computational time and parameters while having good classification performance 2. Transfer learning with band selection method performed on different datasets having the same number of bands	1. Finding hyperspectral datasets with the same bands is a constraint as there is already a shortage of datasets
60	[117] CNN	Feature Relation Map Learning	**IP:** 0.10 **SA:** 0.10 **PU:** 0.10	**Indian Pines** : OA = 97.3; KC = 96.9; **Salinas** : OA = 98.19; KC = 98; **Pavia U** : OA = 98.98; KC =97.3;	1. Identifies the relationships between features under a specific mapping function 2. The identified relationships can be portrayed as a picture 3. New features are learned from the emerged relationships using feature relation map learning 4. No use of dimensionality reduction, exhibit full use of bands	1. For accurate classification, it is essential to find a suitable classifier, especially when the training samples are limited
61	[118] CNN	Capsule Network + Markov Random Field (MRF)	**IP:** 0.10 **SA:** 0.10 **PU:** 0.10	**Indian Pines** : OA = 98.52; AA = 98.41; KC =98.32; **Salinas** : OA = 99.74; AA = 98.68; KC = 95.82; **Pavia U** : OA = 99.84; AA = 94.65; KC = 95.02	1. Enhanced connection of capsule network overcoming the overfitting problem called conv-caps with Markov random field 2. Caps MRF can better use the features, and MRF helps to achieve better convergence 3. Works well with limited or noisy samples under boisterous conditions	1. Using spatially disjoint samples may affect the spectral-based method

62	[119] CNN	Active Learning + Markov Random Field (MRF)	**IP:** 0.05 **PU:** 0.01 **PC:** 0.005	**Indian Pines** : OA = 94.28; AA = 89.79; **Pavia U** : OA = 98.17; AA = 96.4; **Pavia C** : OA = 99.15; AA = 97.45	1. Proposed ensemble the network has CNN that extracts spectral-spatial discriminative features, active learning to reduce the labeling cost, and MRF to understand the spatial correlations 2. Good accuracy and smooth classification maps	1. Labeling of samples could be replaced by unsupervised learning as a future work
63	[120] CNN	Transfer Learning + Markov Random Fields (MRF)	**IP:** 0.04 **PU:** 0.04	**Indian Pines** : OA = 93.89; AA = 89.77; KC = 92.93; **Pavia U** : OA = 91.79; AA = 88.67; KC = 91.64	1. Transfer learning is adopted to reduce the training time, the Bayesian framework to reduce the time, MRF to make use of spatial information	1. Average accuracies of both datasets were slightly lower
64	[121] CNN	Knowledge Distillation	**IP:** 0.2 **HU:** 0.4 **PU:** 0.2	**Indian Pines** : OA = 96.4; AA = 96.1; KC = 96; **Houston U** : OA = 94.7; AA = 94.4; KC = 94.2; **Pavia U** : OA = 99.3; AA = 99.1; KC = 99.1	1. Knowledge Distillation using a teacher/student system for a compact CNN with satisfactory classification performance. 2. The trained network has high accuracy with fewer parameters	1. Results for Indian Pines are satisfactory; however, for Pavia U, accuracy decreases by 2% while compared with the state-of-the-art models
65	[122] CNN	Attention Mechanism: Spatial Attention and Channel wise attention	**IP:** 0.3 **PU:** 0.2 **KSC:** 0.2	**Indian Pines** : OA = 99.52; AA = 99.22; KC = 99.53; **Pavia U** : OA = 100; AA = 99.99; KC = 99.99; **KSC** : OA =100; AA = 100; KC = 100	1. The structure of channel-wise attention followed by spatial attention helps to sort the unimportant information from a significant part 2. Residual connection helps in better convergence, and the group-wise attention module helps in reducing the possibility of losing important information	1. Number of parameters is high 2. Computationally expensive
66	[123] CNN	Attention Mechanism + Dense Connectivity	**PU:** 0.2 **IP:** 0.2	**Pavia U** : OA = 99.97; AA = 99.96; KC = 99.97; **Indian Pines** : OA = 99.29; AA = 99.07; KC = 99.19	1. A broader and deeper network based on CNN that helps to extract discriminative spectral and spatial features 2. Attention mechanism to take hold of discriminative information optimized with augmentation, regularization, and batch normalization.	1. The size of the spatial cube gives different results on different datasets. It has to be optimized. 2. Secondly, the number of training samples needs to be higher for the Indian Pines dataset to achieve high accuracy

67	[124]	CNN	Attention mechanism + Fully connected network	**IP:** 0.7 **BOT:** 0.7 **SA:** 0.7	**Indian Pines** : OA = 97.31; KC = 99.85; **Botswana** : OA = 100; KC = 100; **Salinas** : OA = 100; KC = 99.99	1. The proposed network is applied in three parts. Firstly, 2D and 3D CNN are combined for feature extraction. Secondly, deeper features are generated for cross-domain data. Thirdly, the attention module is integrated	1. Optimization approaches could result in better convergence

Table 2.11: Articles employing CNN as the base model for classification (Year: 2020)

S. No	Ref	Model	Addition for Model Improvement	Training Ratio	Datasets and Results	Contributions	Shortcomings
68	[125]	CNN	Centre Loss + Cross Entropy + Feature Extraction	**IP:** 0.02, **SA:** 0.003	**Indian Pines:** OA = 94.55, AA = 94.44, KC = 93.77; **KSC:** OA = 98.26, AA = 97.48, KC = 98; **Salinas:** OA = 96.13, AA = 97.37, KC = 95.7	1. The proposed network can learn deep as well as refined spectral-spatial features 2. The 3D kernel is broken down into three 1D kernels to reduce parameters and thereby alleviate overfitting 3. Centre loss function to improve the classification accuracy	1. Much high computational requirement 2. Use of the centre loss function increases the cost
69	[46]	CNN	Capsule Network	**IP:** 0.15, **PU:** 0.15	**Indian Pines:** OA = 98.72, AA = 99.41, KC = 98.54; **Pavia U:** OA = 99.97, AA = 99.97, KC = 99.96	1. Based on spectral-spatial capsule units helping to discover discriminative features to reduce overfitting and faster convergence. 2. Works well with limited data and provides relevant and complete information about spectral bands.	1. In future work, this could be topped up by the use of semi-supervised learning strategies
70	[126]	CNN	Transfer Learning	**PU:** 0.03, **IP:** 0.14, **KSC:** 0.07	Pavia U: OA = 99.4, AA = 99.68, KC = 99.2; **Indian Pines:** OA = 98.87, AA = 99.45, KC = 98.68; **KSC:** OA = 98.22, AA = 97.87, KC = 98.02	1. Reduced the computational cost of networks based on CNN and Logistic regression by making them lightweight with fewer parameters 2. Use of Transfer Learning to alleviate the problem of overfitting by transferring knowledge across different domains and different sensors	1. The choice of one pre-trained network for all three datasets is a challenge 2. Homologous datasets may not always show better results than heterologous datasets.

71	[127] CNN	Dense Network + Multi-scale Filter bank + Feature Extraction	**IP:** 0.15, **PU:** 0.05, **SA:** 0.05	**Indian Pines:** OA =99.81, AA = 99.78, KC = 99.78; **Pavia U:** OA = 99.97, AA = 99.98, KC = 99.6; **Salinas:** OA = 99.99, AA = 99.99, KC = 99.99	1. To alleviate the problem of overfitting and extract informative features, Dense connections are both in the fusion part and dual-channel part 2. 2D replaces 3D filters	1. Training and Testing time could be further reduced
72	[128] CNN	Capsule Network + Dynamic Routing	**IP:** 0.3, **PU:** 0.3, **SA:** 0.3	**Indian Pines:** OA =99.86, AA = 99.92, KC=99.04; **Pavia U:** OA = 99.99, AA = 99.98, KC = 99.98; **Salinas:** OA = 99.98, AA = 99.98, KC = 99.98	1. Hybrid Capsule network to deal with low training samples 2. Dimensionality reduction to remove redundant bands 3. Nested cross-validation to find the best value of parameters	1. Higher number of principal components increases the computational cost. Moreover, dynamic routing increases the cost as well.
73	[129] CNN	Robust PCA + Low-Rank Subspace Estimation (LRSE)	**PU:** 0.10, **IP:** 0.10	**Pavia U:** OA = 99.42, AA = 99.08, KC = 99.23; **Indian Pines:** OA =99.47, AA = 99.83, KC = 99.4	1. Robust PCA to fetch the low-rank representation of data and sparse data for image denoising 2. Pointwise convolution for concatenating the two branches	1. RPCA needs appropriate LRSE to achieve good classification performance
74	[130] CNN	Multiscale features + CNN	**PU:** 0.42, **SA:** 0.06	**Pavia U:** OA = 99.13, KC = 98.86; **Salinas:** OA = 99.46, KC = 99.4	1. The proposed network takes multiscale features in each scale as the input 2. The deep features are extracted with 1D CNN	1. Challenge to customize the parameters according to the dataset.
75	[123] CNN	Spatial Residual Block + Spectral Feature Learning	**IP:** 0.20, **PU:** 0.10, **SA:** 0.10	**Indian Pines:** OA =99.68, AA = 99.74, KC = 99.64; **Pavia U:** OA = 99.62, AA = 99.87, KC = 99.89; **Salinas:** OA = 99.86, AA = 99.91, KC = 99.84	1. Spatial block to extract spatial features connected with residual block to avoid the accuracy from falling 2. Spectral feature learning has few trainable parameters to deal with small sample availability 3. Feature fusion to integrate features better	1. The acquisition of labeled samples is time-consuming and labor-intensive
76	[83] CNN	Attention mechanism	**IP:** 0.1, **PU:** 0.02, **SA:** 0.02	**Indian Pines:** OA =95.49, AA = 94.17, KC = 94.85; **Pavia U:** OA = 98.02, AA = 96.9, KC = 97.37; **Salinas:** OA = 96.81, AA = 98.33, KC = 96.54	1. An end-to-end network containing a spectral-spatial unit made of convolutions that helps to find informative features with an attention module to focus on the essential areas	1. The attention module makes the network end up with a longer training time

77	[131] CNN	Residual Network	**IP:** 0.15, **SA:** 0.15, **PU:** 0.15	**Indian Pines:** OA = 99.4, AA = 98.98, KC = 99.31; **Salinas:** OA = 99.97, AA = 99.98, KC = 99.96; **Pavia U:** OA = 99.94, AA = 99.89, KC = 99.92	1. A spectral-based unit using 1D CNN and a spatial-based unit using 2D CNN after being passed through principal component analysis as dimensionality reduction. Finally, spatial and spectral information is fused to create an input data block focused on excellent classification	1. Computationally expensive 2. Parallel implementation could have reduced the time 3. Approaches like GAN or augmentation to produce more data
78	[132] CNN	Attention mechanism + Preactivation mechanism	**IP:** 0.20, **PU:** 0.10, **SA: 0.10**	**Indian Pines:** OA =99.67, AA = 99.37, KC = 99.62; **Pavia U:** OA = 99.92, AA = 99.87, KC = 99.9; **Salinas:** OA = 99.9, AA = 99.93, KC = 99.89	1. Optimization strategies, including residual connections, and batch normalization, are proposed to alleviate overfitting. 2. Attention block with preactivation 3. Performs great even with imbalanced class distribution	1. Longer Computational time to train and test
79	[133] CNN	Conditional Random Field (CRF) + Feature Learning + Self-Supervision	**IP:** 0.02, SA: 0.01, **PU:** 0.004	**Indian Pines:** OA = 83.01, AA = 87.88, KC = 81.9; **Salinas:** OA = 96.82, AA = 96.79, KC = 96.6; **Pavia U:** OA = 85.76, AA = 82.52, KC = 80.8	1. The proposed novel network integrates subpixel, pixel and superpixel information using self-supervised learning 2. CRF gets feedback from the network and feeds it back to the network to enhance the learning process	1. A lot of room left for improving the classification accuracy and optimizing it to perform better.
80	[134] CNN	Residual Network + Visual Attention	**IP:** 0.15, **PU:** 0.10, **SA:** 0.10, **HU:** 0.23	**Indian Pines:** OA = 98.75, AA = 97.05, KC = 98.58; **Pavia U:** OA = 99.86, AA = 99.76, KC = 99.82; **Salinas:** OA = 99.85, AA = 99.83, KC = 99.83; **Houston U:** OA = 88.71, AA = 90.09, KC = 87.73	1. Visual Attention network to improve the feature extraction process to extract representative and informative features 2. Works well with limited data 3. Satisfactory classification results 4. Great feature extraction capabilities	1. Parameter optimization is required 2. Classification accuracy for the Houston U dataset could be improved

This section embarks on a journey through the evolution of HSIC leveraging CNNs from **2019** to **2024**. Researchers have continuously expanded the horizons of HSIC with a diverse array of techniques. In **2019**, dense networks were introduced, enhancing feature extraction and mitigating overfitting by establishing direct connections between all layers. The year **2020** witnessed the integration of 3D-ANAS, Incremental Principal Component Analysis (PCA), and Superpixel Segmentation, enriching feature extraction capabilities and potentially improving the capture of spatial and spectral information. Notably, the adoption of 3D Gabor Filters facilitated the extraction of directional features, augmenting the representation of spatial and spectral characteristics. Additionally, a notable shift

towards depth-wise convolutions and specialized feature extractors emerged, aiming to enhance feature learning efficiency and effectiveness. In **2021**, attention mechanisms took centre stage, allowing selective focus on informative features and potentially improving classification performance. By **2022**, depth-wise convolutions became prominent, reducing computational cost and enhancing efficiency. Spatial pyramid pooling techniques were also employed to aggregate features at various spatial scales, capturing multi-scale information effectively. Moreover, the utilization of specialized feature extractors tailored to specific tasks or datasets gained traction, further optimizing feature learning. As the journey progressed into **2023**, exploration into vision transformers and novel attention mechanisms intensified. Vision transformers offered promise in capturing long-range dependencies in spectral information, especially beneficial with limited labels. Novel attention mechanisms honed in on specific regions or features within hyperspectral data, potentially enhancing classification accuracy. The early stages of **2024** witnessed the emergence of GANs, offering the ability to generate synthetic HSI data and address the challenge of limited training data. Throughout these years, training and optimization strategies have evolved in tandem with model advancements. Semi-supervised learning techniques, augmentation strategies, and the integration of residual networks have been instrumental in leveraging unlabeled data, mitigating overfitting, and addressing the vanishing gradient problem, respectively. CNNs have become one of the leading techniques for HSI due to their remarkable ability to exploit both spatial and spectral information. This dominance is reflected in the prevalence of keywords like CNN, deep learning, hyperspectral images, classification, and feature extraction when exploring relevant research through keyword cloud charts as shown in Figure 2.4.

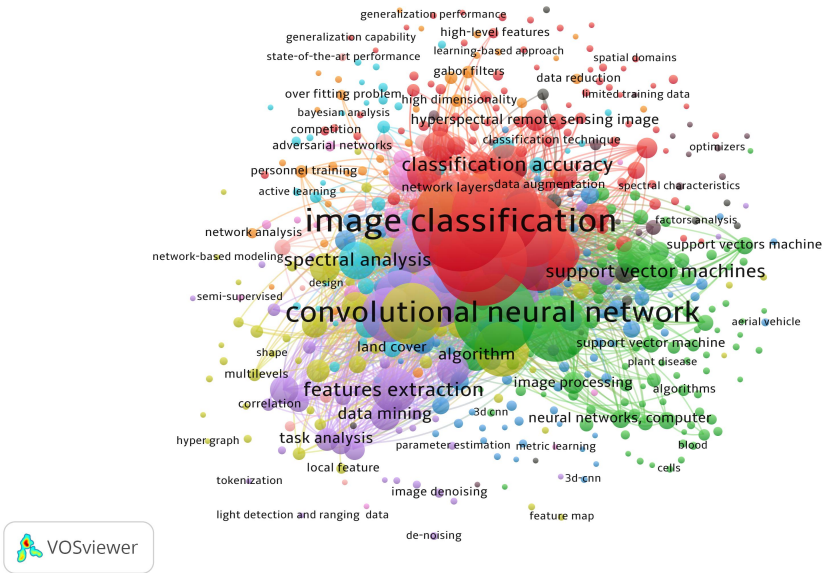


Figure 2.4: Keywords encountered in studies on CNN and HSIC

Figure 2.5 shows the co-authorship diagram with the extent of interdisciplinary research by illustrating connections between researchers in HSI. It visualizes the collaborative relationships among researchers in HSI which helps in identifying how researchers are connected and who collaborates with whom.

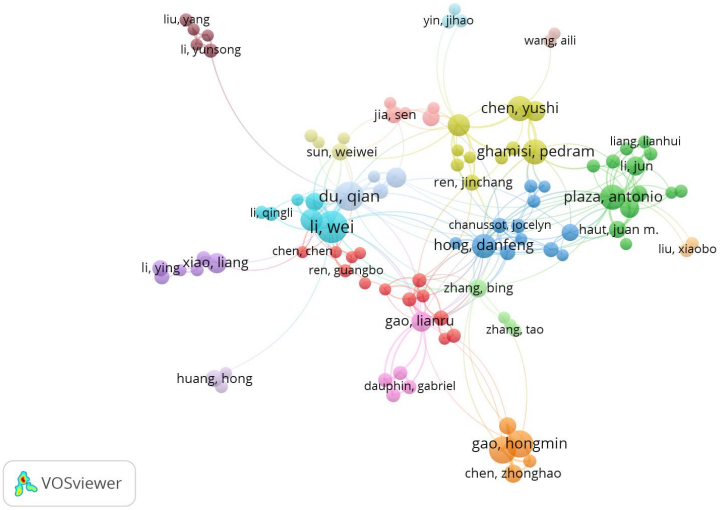


Figure 2.5: Most popular studies using CNN in HSIC

Furthermore, this analysis unfolds in a nuanced exploration of publications within a two-tier framework: initial classification by OA, followed by categorization based on the addressed problem. Consequently, this presentation encapsulates not only the findings of the authors but also dissects their contributions and shortcomings, providing a comprehensive panorama of the evolving landscape in the forthcoming sections.

Models based on CNN achieving more than 90% Overall Accuracy

In this section, we outline several key challenges encountered in HSIC and highlight recent research efforts aimed at overcoming these obstacles. Each challenge represents a crucial aspect of HSIC while using CNN models, ranging from imbalanced class distributions to limited labeled samples, noise and more. Researchers have developed innovative approaches and techniques to tackle these challenges, aiming to enhance the accuracy, convergence, feature extraction of HSIC. Through this section, we look into the diverse array of challenges faced by practitioners who have used CNN for HSIC and explore the cutting-edge solutions proposed to address them.

Challenge : Addressing Imbalanced Classes and Spatial Resolution Loss

- Wu et al. [122] introduced a channel and spatial attention network. This approach effectively combats information loss and enhances context extraction in datasets with

imbalanced classes. However, the computational complexity of separate attention mechanisms for both channels and spatial dimensions needs further investigation.

- Yang et al. [124] proposed a fusion architecture combining 2D and 3D CNNs with data fusion for spectral-spatial information integration. While this method improves classification accuracy, it raises concerns regarding the high computational cost and convergence time associated with such complex networks.

- Bai et al. [127] achieved remarkable results by leveraging spectral-spatial capsules to streamline complexity. This method offers a promising alternative to traditional spectral-spatial networks, but further research is needed to address potential limitations in handling highly diverse or noisy data.

Challenge : Gradient Vanishing and Overfitting

- Khodadadzadeh et al. [128] introduced dynamic routing and hyperparameter optimization to address gradient vanishing in capsule networks. This approach improves performance, particularly with limited training data. However, the effectiveness of this method compared to alternative gradient handling techniques in capsule networks requires further evaluation.

- Addressing overfitting and gradient updating inefficiencies in CNN models, Jia et al. [63] proposed a multilayer, multi-branch architecture incorporating 3D Gabor filters for enhanced feature extraction. Tackling the same challenges, Gao et al. [88] introduced a stochastic depth residual network, shortening training time by randomly skipping some residual blocks.

Challenge : Small Sample Datasets

- Khotimah et al. [98] proposed a two-stream residual network for efficient training with small datasets. While offering a solution, this approach might not be optimal for all types of HSI data, and exploring its generalizability is crucial.

- Yue et al. [70] and Liu et al. [38] introduced techniques using spatial pyramid pooling and shuffled group CNNs, respectively, for handling small sample datasets. These methods offer efficient training strategies but require further investigation regarding their effectiveness with highly complex or imbalanced datasets.

- Given the prevalence of small sample datasets, He et al. [116] introduced heterogeneous transfer learning, ensuring consistency across datasets by aligning the number of channels supplied into pre-trained CNNs with an attention module. In a complementary effort to address small sample datasets, Liu et al. [72] developed a multidimensional CNN with an attention module, incorporating both 3D and 2D con-

volutional layers to extract spectral-spatial features and analyze spectral and spatial properties at an abstract level.

- To tackle small sample sets, Jia et al. [91] proposed a lightweight CNN with Schrodinger eigenmaps for dimensionality reduction and dual-channel convolution for improved classification. Addressing limited labelled samples, Yue et al. [87] introduced adaptive knowledge distillation using semi-supervised learning. Yu et al. (2024) proposed an end-to-end network based on CNN, pooling, and dropout layers, addressing minor training set problems.

Challenge : Processing Multiscale Features

- Gao et al. [99] proposed a hierarchical shrinkage network to address the resource intensity of processing multiscale features. While offering a more efficient solution, this method might lead to a trade-off in terms of feature extraction accuracy compared to more complex architectures.

- Siamese CNNs with adaptive pooling introduced by Rao et al. [30], offer solutions for limited sample availability and generalization when dealing with multiscale features. However, the effectiveness of this approach compared to alternative architectures for multiscale feature processing needs further evaluation.

Challenge : Limited Labeled Samples and Noise

- Ge et al. [69] introduced spectral and spatial attention modules to enhance classification with limited labelled data and noise. This approach offers promising results, but a comparative analysis with other noise reduction techniques is necessary to assess its overall effectiveness.

- Zhang et al. [135], Qing et al. [97], Zhong et al. [136], and Xu et al. [109] all proposed various network architectures (dense connections, multi-stage models, spectral-spatial residual networks, and multiple spectral 3D CNNs) to address limited labelled samples. While these methods offer solutions, a comprehensive evaluation comparing their performance and generalizability across different datasets is crucial.

- Addressing limited labelled samples, Huang and Chen (2024) introduced a dual-path similarity-based network using adversarial networks for spatial and spectral information. Pan et al. [137] proposed a semi-supervised multi-grained network correlating spectral and spatial information with minimal hyperparameters. Xu et al. [79] utilized noise during data collection, employing a dual-channel residual network and a noise-resilient loss function that even incorporates mislabeled samples

Challenge : Convergence Rate and Sensor-Specific Models

- Wang et al. [113], Haut et al. [134], and Xu et al. [109] all proposed methods to improve the convergence rate for spectral-spatial information processing. While these approaches offer faster training, a trade-off between convergence speed and classification accuracy might exist, requiring further investigation.

- Zhang et al. [64] proposed a pixel-to-pixel processing structure to address the limitations of sensor-specific, patch-based models. This method offers a more generalizable solution, but its effectiveness compared to patch-based approaches with sensor-specific optimizations needs to be evaluated across various sensor types.

- While two-dimensional CNN models serve as viable classification tools, the utilization of three-dimensional CNNs, although capable of leveraging spectral information in HSI cubes, often struggles with perfect feature map classification. To address this challenge, Ahmad et al. [65] proposed a methodology involving the breakdown of the HSI cube into 3D patches, constructing 3D feature maps with fewer parameters for improved computational efficiency. Similarly, Cui et al. [83] introduced MobileNetV3, incorporating depth-wise convolutions to reduce computational time without compromising classification performance.

- To further optimize computing resources while maintaining accuracy, Paoletti et al. [138] suggested a spectral-spatial information-based 3D network capable of effectively handling border areas in images, particularly constructed utilizing graphics processing units. Additionally, Meng et al. [66] proposed a lightweight CNN architecture that significantly reduced network parameters and computing time using multiply-accumulate operations while enhancing classification performance. Chang et al. [74] consolidated CNN architectures, harnessing the advantages of both 2D and 3D CNNs to learn spatial characteristics and spectral bands, thereby addressing data redundancy and insufficient spatial resolution while retaining processing speed and classification accuracy.

- Efficient HSI classification often requires minimizing computing costs. Leveraging the lightweight nature of the Ghost network, Paoletti et al. [102] incorporated a ghost module into CNN, significantly reducing network parameters and runtime.

Challenge : Efficient Feature Extraction

- Researchers have conducted diverse feature extraction tests to enhance classification performance. Using a semi-supervised approach, Wei et al. [73] extracted features, while Wang et al. [139] employed dimension reduction with PCA and 1DCNN to eliminate redundant information from HSI. In a similar vein, Fang et al. [108] focused on discriminative samples and implemented an early exiting strategy, effectively reducing computing costs for easy samples. Jiao et al. [139] further refined

feature extraction by retrieving discriminative features through the excavation of multiscale spatial information, integrating a pre-trained network for spectral feature fusion through a weighted approach.

- To enhance spatial-spectral information utilization, Xi et al. [84] introduced a network with 3D Squeeze and Excitation residual blocks, exploring both low-level and high-level spatial-spectral information. Complementarily, Zou et al. [76] devised a spectral-spatial information network addressing information leakage through bidirectional attention maps for improved and interactive feature maps.

2.2.3 Review of Studies based on Autoencoders

Autoencoders, an integral component of deep learning architectures, have emerged as powerful tools for feature extraction tasks. Comprising encoder and decoder networks, autoencoders aim to efficiently compress high-dimensional input data into a lower-dimensional representation, while maintaining the essential features necessary for accurate reconstruction. This process is fundamental in various domains, including computer vision, natural language processing, and signal processing.

$$x \rightarrow g_e(b, W; x) \rightarrow c \rightarrow g_d(\hat{b}, \hat{W}; c) \rightarrow \hat{x} \tag{2.5}$$

Conventionally, autoencoders consist of an encoder network responsible for compressing the input data and a decoder network tasked with reconstructing the original input from the compressed representation. This architecture is typically represented as a left-to-right flow, with the encoder preceding the decoder. Mathematically, the encoding process can be represented as $x \rightarrow g_e(b, W; x) \rightarrow c$, where x denotes the input vector, g_e represents the encoding function parameterized by weights W, and c represents the encoded representation. The decoder network, denoted by g_d, then operates on the encoded representation to produce the reconstructed output $\hat{x}$. In the collected data, a smaller subset of studies were identified that integrated autoencoders into classification pipelines, thereby showcasing the potential of autoencoders for enhancing classification performance.

Table 2.13: Articles with autoencoders (AE, SAE, SCAE, and CAE) as feature extractors fed to classification.

S. No	Ref	Feature Extraction Module	Addition to improvise the proposed model	Training Ratio	Datasets Used and Results	Contributions	Shortcomings

Continued on next page

Table 2.13: Articles with autoencoders (AE, SAE, SCAE, and CAE) as feature extractors fed to classification. (Continued)

1	[140]AE + SSCL	Encoders + Contrastive Loss + PCA + EMP + Data Augmentation	**SA:** 0.10 **PU:** 0.10 **BOT:** 0.10	**Salinas**: OA = 99.70; AA = 99.67; KC = 99.67; **Pavia U**: OA = 97.21; AA = 96.00; KC = 96.30; **Botswana**: OA = 97.96; AA = 98.02; KC = 97.70	1. This study proposed a classification algorithm based on contrastive learning using self-supervised learning 2. The first stage of training involves learning from unlabeled samples to construct positive and negative sample pairs 3. The pre-trained model uses a small number of labeled features to fine-tune the network	1. The effect of neighborhood size is sensitive to datasets 2. The Botswana dataset is larger in area and sparse 3. Choosing an optimal batch size is a challenge
2	[141]SAE + SVM	PCA + Distance Transform Image + EMAP + SAE	**SA:** 0.10 **PU:** 0.10 **SU:** 0.10	**Salinas**: OA = 97.93; AA = 98.88; KC = 98; **Pavia U**: OA = 99.34 ; AA = 99.11 KC = 99; **Surrey**: OA = 94.31; AA = 93.98; KC = 92	1. A spatial-spectral feature extraction model for pixel-wise classification 2. Distance transform values to add weights to target pixels 3. EMAP to further integrate geometric information	1. SAE's weight and the model's hyperparameter are restricted to every dataset 2. Applying this model to a new dataset means fine-tuning the model again
3	[142]SAE + MLR	CNN	**SA:** 0.10 **PU:** 0.10 **KSC:** 0.10 **IP:** 0.10	**Salinas**: OA = 99.91; AA = 99.87; KC = 99.68; **Pavia U**: OA = 99.88; AA = 99.54; KC = 99.71; **KSC**: OA = 98.47; AA = 97.51; KC = 97.65; **Indian Pines**: OA = 98.18 ; AA = 97.80; KC = 97.89	1. The proposed model combines CNN and SAE and extracts spectral-spatial features jointly 2. To improve classification accuracy, adaptive weight generation using CNN, depending on spatial contextures 3. MLR used for the classification process	1. Choosing the size of features is sensitive to the dataset 2. A deeper SAE network may lead to overfitting
4	[143]Softmax + CD-SAE	LFDR+ Diversity Regularization	**PU:** 0.04 **IP:** 0.24 **SA:** 0.06	**Pavia U**: OA = 97.59; AA = 97.66; KC = 96.86; **Indian Pines**: OA = 95.81; AA = 97.38; KC = 95.30; **Salinas**: OA= 96.07; AA=97.56; KC=96.78;	1. Proposed compact and discriminative stacked auto encoder having two stages 2. Incorporated Local Fisher discriminant regularization to effectively learn the pixels 3. Diversity regularization to balance feature dimensionality and feature representation.	1. Run time increases as the network size or number of samples increase

Continued on next page

Table 2.13: Articles with autoencoders (AE, SAE, SCAE, and CAE) as feature extractors fed to classification. (Continued)

5	[36]	Softmax + SAE	Linear Prediction Based Band Generation + MRF	**IP:** 0.10 **PU:** 0.05 **SA:** 0.05	**Indian Pines**: OA = 98.39; AA = 98.14; KC = 98.2; **Pavia U**: OA = 99.20; AA = 98.26; KC = 98.9; **Salinas**: OA = 98.53; AA = 99.00; KC =98.0	1. Fine Tuned samples as training samples 2. More confidential generation of training samples that improves the generalization performance 3. Better than traditional spectral-spatial models as it fuses information at the decision level	1. Can be optimized for limited sample data 2. The use of active learning can be done to guide the selection of unlabeled samples better 3. Improving the computing time

AE Models achieving more than 96% Overall Accuracy

This observation highlights the relatively underexplored area of utilizing autoencoders for feature extraction in classification tasks. Despite the limited number of studies, the findings from these investigations provide valuable insights into the effectiveness of autoencoder-based approaches and their impact on classification accuracy. The studies related to autoencoders are presented in Table 2.13 depicting a summary of the contributions and shortcomings of each study, including autoencoders [140], [144], stacked autoencoders [141], [142], [36], discriminant autoencoders [143], and convolutional stacked autoencoders [145], employed with classification models such as SVM [141], [144], and Multinomial Logistic Regression (MLR) [142]. Most studies in the survey employing autoencoders achieved a classification accuracy of more than 96%, depicted in Figure 2.6.

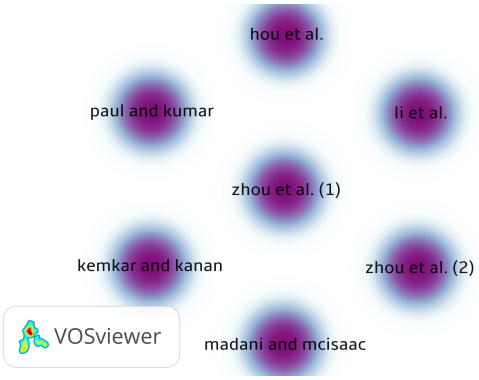


Figure 2.6: Density Visualization of studies having OA more than 96% with AE for HSIC

- Hou et al. [140]: Addressing the challenge of **limited labelled sets**, Hou et al. revolutionized the approach by blending enhanced unlabeled examples with contrastive learning in the SSCL (Self-supervised Contrastive Learning) network. Their innovative method includes a deep residual network encoder that accepts enhanced data to significantly minimize computational requirements.

- Madani and McIsaac [141]: The work of Madani and McIsaac focuses on **feature extraction**, proposing a spatial and spectral information-based method. Their approach

involves utilizing a stacked encoder with a sparse AE and introducing an innovative Stacked Autoencoder (SAE) pretraining stage, showcasing advancements in feature extraction techniques.

- Li et al. [142]: Crafting an effective fusion of spectral and spatial information, Li et al. introduced an adaptive feature learning approach integrating CNN and AE within a single framework. Their method flexibly incorporates spatial information, resulting in improved feature discriminability without the **need for labelled data**.

- Zhou et al. [36]: Zhou et al. introduced a semi-supervised AE with a dual-training strategy on spectral and spatial data. Their approach, involving two independent Stacked AEs processing spectral-spatial data, demonstrates exceptional generalization performance by leveraging the diversity of samples for increased classification accuracy.

- Kemkar and Kanan [145]: Kemkar and Kanan proposed self-taught learning for classification, leveraging a vast amount of unlabeled data to extract distinctive features, showcasing the potential of unsupervised learning in **enhancing classification accuracy**.

- Zhou et al. [143]: Zhou et al. unveiled a low-dimensional feature space through a compact and discriminative SAE empowered by LFDR (local fisher discriminant regularization), achieving substantial between-class separation and **enhancing classification accuracy**.

- Paul and Kumar [144]: Navigating complexity with spectral segmentation, AE, and morphological profiles, Paul and Kumar introduced a mutual information-based segmented AE to **minimize algorithm time**. Their approach synergizes SAEs and morphological profiles for enhanced feature extraction across diverse datasets.

2.2.4 Review of Studies based on Graph-Based Learning

CNNs have been pivotal in computer vision, excelling in tasks like image classification and object recognition. However, extending their capabilities to non-grid data structures, like graphs, has become increasingly important. GCN achieve this, offering a powerful framework for learning from graph-structured data. In hyperspectral image classification, where data is represented as hypercubes with multiple spectral bands, GCNs have emerged as a promising solution. By treating pixels or regions as nodes and leveraging spectral-spatial relationships, GCNs effectively capture discriminative representations for classification. This section provides a focused review of GCNs for hyperspectral image classification, bridging

the gap between traditional CNNs and graph-based approaches. By exploring their architecture, strengths, and applications, we aim to inspire further research in leveraging GCNs for analyzing HSI data.

Table 2.14: Articles with Graph Convolutional Networks as the base network

S. No	Ref Model	Addition	Training Ratio	Results	Contributions	Shortcomings
1	[146]GCN	SGC + DNAGSI	**SA:** 0.01 **IP:** 0.10 **PU:** 0.01	**Salinas**: OA = 99.70; AA = 99.44; KC = 99.66; **Indian Pines**: OA = 98.36; AA = 97.79; KC = 98.12; **Pavia U**: OA = 99.30; AA = 98.73; KC = 99.07	1. A novel graph-based method that dynamically generates a graph structure and adaptively merges the newly generated graphs with an existing graph 2. To generate high-level feature representations, use of initial residual and identity mapping 3. To increase interclass distance and reduce intraclass distance, use of joint loss with centre loss 4. Performs better than CNN-based methods	1. Spends more time in the training stage 2. Parameters increase with the depth of the network 3. Exploring an adaptive hyperparameter selection is challenging
2	[147]GCN	Siamese Network	**IP:** 0.05 **PU:** 0.006 **SA:** 0.008 **HU:** 0.03	**Indian Pines**: OA = 96.98; AA = 95.24; KC = 96.51; **Pavia U**: OA = 95.55; AA = 94.78; KC = 94.28; **Salinas**: OA = 99.43; AA = 99.54; KC = 99.37; **Houston U**: OA = 96.05; AA = 96.04; KC = 95.73	1. An automatic graph convolutional network (GCN) models the synergy of high-order tensors. 2. Incorporated with Siamese network with a novel labeling approach to computing the similarity between the features 3. GCN and Siamese are collaborated and jointly trained 4. Good classification accuracy and does not suffer from a wide range of misclassification	1. Classification performance of Houston University dataset can be increased
3	[148]GCN	MUDA + Tangential graph + Neighborhood Structure Graph	**PU:** 0.01 **HU:** 0.06	**Pavia U**: OA = 83.0; AA = 87.3; KC = 78.1; **Houston U**: OA = 92.6; AA = 92.4; KC =92.0; **Urban**: OA = 95.1; AA = 94.9; KC = 93.9	1. Constructs an interclass and intraclass structure graph by making use of radial neighborhood information 2. Using the tangential information, constructs a tangential graph for interclass and intraclass samples. 3. Gaussian distribution to improve the scatter of distributed features	1. It takes more time to run the model because of the time taken to build neighborhood structure graphs, tangential graphs, and Gaussian weighted model to extract discriminative features 2. Accuracy can be improved for all datasets

Continued on next page

Table 2.14: Articles with Graph Convolutional Networks as the base network (Continued)

S. No	Ref Model	Addition	Training Ratio	Results	Contributions	Shortcomings
4	[149]GCN	Sa-GCN + Se-GCN + Attention mechanism	**IP:** 0.05 **PU:** 0.006	**Indian Pines**: OA = 94.84; AA = 96.43; KC = 94.10; **Pavia U** : OA = 96.45; AA = 96.40; KC = 95.30	1. The proposed network contains two subnetworks with spectral GCN and spatial GCN to extract the spectral and spatial features and suppress noise 2. Graph cross-attention fusion by merging attention mechanism	1. Due to spectral-spatial features involved, it consumes more time than other methods
5	[150]GCN	Point Graph + Distribution Graph + Drop edge + Cutout	**SA:** 0.001 **IP:** 0.008 **PU:** 0.001 **HU:** 0.005	**Salinas**: OA = 87.33; AA = 88.23; KC = 86.49; **Indian Pines** : OA = 77.46; AA = 77.79; KC = 77.63; **Pavia U**: OA = 79.86; AA = 79.74; KC = 77.09; **Houston U** : OA = 63.52; AA = 63.60; KC = 61.51	1. Dual GCN, Point graph and Distribution graph 2. Reduces the need for labeled samples 3. Alleviates overfitting using drop edge 4. Regularization using improved cutout with the multiscale operation	1. Higher training time 2. Accuracy can be highly improved for all datasets
6	[151]GCN	Dual GCN	**IP:** 0.05 **PU:** 0.006 **SA:** 0.009 **HU:** 0.03	**Indian Pines**: OA = 94.16; AA = 96.41; KC = 93.34; **Pavia U**: OA = 93.24; AA = 93.76; KC = 91.14; **Salinas**: OA = 97.61; AA = 96.94; KC = 97.34; **Houston U**: OA = 91.72; AA = 92.52; KC = 91.03	1. Proposed a network with dual GCN across GCN branches that considers multiscale spatial information to refine the graph information 2. Enhanced feature representation capability by fusing information from two branches 3. Accelerated convolution operation	1. Misclassification in case of noisy pixels in a few classes 2. Convergence is slower for the Salinas dataset 3. Higher computational complexity 4. Sensitive neighborhood size 5. Accuracy can be further increased
7	[152]GCN	SR Graph + Co-Propagation	**KSC:** 0.10 **BOT:** 0.12 **SV:** 0.10	**KSC**: OA = 99.38; AA = 99.20; KC = 99.30; **Botswana**: OA = 99.65; AA = 99.66; KC = 99.61; **Salinas-V** :OA = 97.16; AA = 98.60; KC = 96.83	1. Use of label propagation to generate labels for unlabeled classes 2. SR spectral model is regularized using spectral statistical information to enhance discrimination further 3. SR spatial model is regularized using superpixel block constraint 4. The models have collaborated	1. Time costlier 2. High computation cost 3. Co-propagation decreases class-specific accuracy 4. Fine-tuning is required
8	[153]GCN	Poisson Learning	**IP:** 0.007 **PU:** 0.001 **SA:** 0.001	**Indian Pines**: OA =86.91; AA = 87.02; KC = 83.32; **Salinas** : OA = 96.75; AA = 96.47; KC = 96.38; **Pavia U** : OA = 86.74; AA = 89.37; KC = 84.05	1. Use of Poisson learning for label propagation 2. Construction of spectral and spatial graphs and fusion using iterative label propagation 3. Feedback strategy to update the fused graph in run time	1. Accuracy can be increased, especially for Indian Pines and Pavia U 2. Computational time is comparable but can be decreased further

Continued on next page

Table 2.14: Articles with Graph Convolutional Networks as the base network (Continued)

S. No	Ref Model	Addition	Training Ratio	Results	Contributions	Shortcomings
9	[154]GCN	Enhanced super-pixel segmentation	**IP:** 0.04 **PU:** 0.006 **HU:** 0.18	**Indian Pines**: OA =95.87; AA = 97.45; KC = 95.27; **Pavia U** : OA = 98.4; AA = 99.11; KC = 97.98; **Houston U**: OA = 88.57; AA = 89.92; KC = 87.62	1. An end-to-end network with an enhanced process of super-pixel segmentation 2. Use of multiscale spectral-spatial feature extractor 3. Mix hop super pixel-based GCN model to integrate information 4. Pixel-wise classification process causes lesser misclassification than CNN	1. Fewer parameters but a hundred times higher floating point operations required 2. Accuracy for the Houston dataset is low
10	[155]GNN	Edge Convolution	**IP:** 0.04 **PU:** 0.005 **KSC:** 0.06	**Indian Pines**: OA = 93.12; AA = 95.78; KC = 92.14; **Pavia U:** OA = 95.31; AA = 96.28; KC = 93.84; **KSC**: OA = 98.61; AA = 97.81; KC = 98.56	1. Graph Neural Network (GNN) added with Edge convolution for message passing to extract the informative features fully 2. Fusion of superpixel and pixel level feature fusion to avoid loss of resolution 3. Smaller computational cost	1. Accuracy can be further increased for Indian Pines and Pavia U
11	[156]GCN	Similarity Measurement + Attention Mechanism	**IP:** 0.10 **PU:** 0.06 **SA:** 0.05	**Indian Pines**: OA = 98.61; AA = 99.01; KC = 98.41; **Pavia U**: OA = 99.44; AA = 99.28; KC = 99.26; **Salinas**: OA = 99.04; AA = 99.39; KC = 98.93	1. Design of a novel similarity measurement method for associating features 2. Aggregation of bands with similar spectra using an attention mechanism 3. Deep GCN for feature extraction using dense connections and dilated convolutions	1. High computational complexity 2. An end-to-end network could be proposed
12	[157]GCN	Hypergraph Learning	**HP:** 0.07 **KSC:** 0.07 **BOT:** 0.05 **PU:** 0.006	**Indian Pines**: OA = 96.25; AA = 97.02; KC = 95.87; **KSC**: OA = 97.53; AA = 97.24; KC = 97.29; **Botswana**: OA = 98.78; AA = 98.89; KC = 98.72	1. Integration of hypergraph learning into graph convolutional networks 2. Hypergraph modeling of the relationship between samples and labels 3. Higher-order proximity preserved by hypergraph structure 4. Adaptive selection of neighbors with hypergraph information	1. Accuracy improvement is needed for Indian Pines 2. Sensitive to hyperparameter settings 3. Requires further testing on large datasets
13	[158]GCN	Multi-Graph Attention	**IP:** 0.06 **PU:** 0.01 **BOT:** 0.10	**Indian Pines**: OA = 95.79; AA = 96.73; KC = 94.96; **Botswana**: OA = 97.22; AA = 97.15; KC = 96.68; **Houston U**: OA = 94.41; AA = 95.11; KC = 93.38	1. Construction of multi-graphs to capture the global and local structural information of hyperspectral data 2. Graph attention mechanism to adaptively assign different weights to different graphs 3. Integration of multi-graphs and attention mechanism for feature learning	1. Time-consuming during the training stage 2. Complexity increases with the number of graphs

Continued on next page

Table 2.14: Articles with Graph Convolutional Networks as the base network (Continued)

S. No	Ref Model	Addition	Training Ratio	Results	Contributions	Shortcomings
14	[159]GAT	Superpixel Based Graph	**IP:** 0.05 **BOT:** 0.08 **HU:** 0.05	**Indian Pines**: OA = 97.15; AA = 97.69; KC = 96.48; **Botswana**: OA = 99.44; AA = 99.42; KC = 99.35; **Houston U**: OA = 96.18; AA = 96.46; KC = 95.84	1. Integration of superpixel-based graph to capture local contextual information 2. Graph attention mechanism for better feature learning 3. Reduction of computational complexity by using superpixels	1. Improvement needed in classification accuracy for the Houston dataset
15	[160]GAT	Structure Similarity	**IP:** 0.06 **PU:** 0.01 **HU:** 0.05	**Indian Pines**: OA = 94.82; AA = 96.34; KC = 93.80; **Houston U**: OA = 93.78; AA = 94.02; KC = 93.47; **Urban**: OA = 97.63; AA = 97.76; KC = 97.57	1. Introduction of a structure similarity measurement method for graph attention networks 2. Consideration of the spatial structure information of hyperspectral images 3. Enhanced feature extraction using the attention mechanism	1. Accuracy improvement is required for Indian Pines and Houston U 2. Higher computational cost
16	[161]GAT	Graph Capsule	**IP:** 0.05 **BOT:** 0.08 **HU:** 0.05	**Indian Pines**: OA = 97.23; AA = 97.68; KC = 96.68; **Botswana**: OA = 99.22; AA = 99.26; KC = 99.19; **Houston U**: OA = 96.02; AA = 96.31; KC = 95.71	1. Incorporation of graph capsules into graph attention networks 2. Enhanced feature extraction using capsule networks 3. Improvement of graph capsules for capturing hierarchical features	1. Accuracy improvement needed for the Houston dataset
17	[162]GAT	Adaptive Edge Learning	**IP:** 0.04 **PU:** 0.008 **SA:** 0.005	**Indian Pines**: OA = 95.64; AA = 96.71; KC = 94.57; **Salinas**: OA = 97.91; AA = 97.66; KC = 97.29; **Urban**: OA = 95.23; AA = 95.42; KC = 94.85	1. Adaptive edge learning for refining the graph structure 2. Attention mechanism to capture informative features 3. Improved feature extraction using graph attention networks	1. Computational complexity is relatively high 2. Fine-tuning is required for optimal performance
18	[163]GAT	Spatial-Spectral Attention	**IP:** 0.05 **HU:** 0.08	**Indian Pines**: OA = 96.84; AA = 97.33; KC = 95.81; **Houston U**: OA = 92.85; AA = 93.23; KC = 92.23; **Urban**: OA = 94.15; AA = 94.32; KC = 93.79	1. Spatial-spectral attention mechanism for capturing both spatial and spectral information 2. Hierarchical feature learning using graph attention networks 3. Improved representation of hyperspectral data	1. Accuracy improvement is needed for the Houston dataset 2. Time-consuming during the training stage

Studies with GCN Models achieving more than 96% Overall Accuracy

This part presented herein addresses several key challenges encountered in HSI classification and explores innovative solutions leveraging GCNs. From enhancing classification accuracy to dealing with unlabelled data and addressing computational complexities, researchers have devised a myriad of approaches to overcome these obstacles mentioned below. The network connection of these studies is visualized in Figure 2.7.

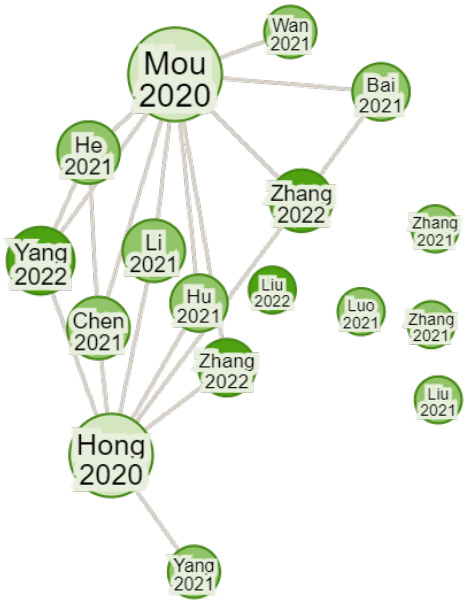


Figure 2.7: Density Visualization of studies having OA more than 96% with GCN for HSIC

Challenge: Enhancing Classification Accuracy

- In addressing the challenge of enhancing classification accuracy in hyperspectral image classification, researchers have explored various strategies to overcome the inherent limitations of graph-based methods. Yang et al. [146] introduced the Super Pixel-Graph Construction (SGC) method, leveraging adaptive graph construction to enhance discriminative ability. However, while SGC addresses the issue of discriminative ability, it may still encounter challenges related to scalability and computational complexity due to its reliance on graph construction.

- To address similar challenges differently, Chen et al. [147] proposed Auto-GCN, aiming to streamline the process of updating graphs. By integrating semi-supervised Siamese networks through novel labelling approaches, Auto-GCN reduces manual efforts and enables joint training for more meaningful feature extraction. Nonetheless, the reliance on semi-supervised learning may introduce challenges related to the availability and quality of labelled data, potentially limiting its applicability in scenarios with limited annotated samples.

- To further capitalize on the inherent association between spectral bands in hyperspectral images, Yang et al. [149] pioneered a dual-branch network architecture. By integrating spatial and spectral Graph Convolutional Networks (GCNs) with a cross-attention fusion module, they achieved enhanced differentiation of information and improved classification accuracy. Despite its effectiveness, this approach may face

challenges related to computational complexity and model interpretability, particularly in scenarios with large-scale hyperspectral datasets.

- Building upon this understanding, Wan et al. [151] addressed the limitations of traditional GCNs in adequately considering the spatial context. Their proposed dual interactive GCN incorporates contextual spatial details to enrich graph information, leading to improved overall performance in hyperspectral image classification tasks. However, the dual interactive GCN approach may introduce challenges related to model training and optimization, particularly in scenarios with complex spatial relationships and heterogeneous spectral characteristics.

Challenge: Dealing with Unlabelled Data

- Dealing with unlabeled data presents a significant challenge in hyperspectral image classification. Zhang et al. [64] tackled this by incorporating sparse representation for graphs in both spectral and spatial dimensions, enhancing classification through label propagation. Additionally, to address performance degradation due to limited labelled samples, Zhang et al. [154] introduced differentiable superpixel segmentation and refined pixel boundaries before classification. Similarly, Liu et al. [157] proposed a novel approach surpassing traditional GCN-based methods by integrating feature fusion and hypergraph structures, showcasing the effectiveness of various fusion strategies. These studies collectively highlight innovative approaches to mitigate challenges posed by limited labelled data, fostering advancements in hyperspectral image classification.

Challenge: Massive Computations

- Massive computations pose a significant challenge in GCN-based hyperspectral image classification. Liu et al. [72] proposed multilayer graphs based on progressive superpixel merging and un-pooling features to adapt quickly to pixel-wise classification.

- Zhang et al. [163] introduced a feature spectra-spatial extraction method, utilizing graph convolution to fuse both types of information. By incorporating graph pooling to reduce duplicate features, these approaches preserve important information while minimizing unimportant patches, thereby improving computational efficiency.

- Hong et al. [160] introduced minibatch GCNs to reduce computational costs, enabling the training of large-scale GCNs in smaller batches. Additionally, Mou et al. [161] proposed an end-to-end semi-supervised network based on non-local GCN, streamlining the annotation process while maintaining high classification accuracy. These advancements underscore ongoing efforts to overcome computational barriers and improve the efficiency of hyperspectral image analysis techniques.

Challenge: Feature Discrimination

- Lastly, addressing the challenge of identifying discriminative aspects within hyperspectral data, Luo et al. [148] introduced the multi-structural unified discriminative embedding (MUDA) method. By analyzing neighbourhood, tangential, and statistical features associated with each pixel, MUDA effectively shapes interclass and intraclass samples using neighbourhood and tangential structure graphs, leading to more accurate classification outcomes.

2.2.5 Review of Studies based on Ensemble Networks

Ensemble Networks leverage the diversity of multiple base classifiers to improve classification accuracy and reliability in hyperspectral image analysis. By combining predictions from individual models, Ensemble Networks mitigate the limitations of individual classifiers and exploit the complementary strengths of diverse classifiers to achieve superior classification performance. This ensemble approach enables more robust classification decisions by capturing a broader range of spectral and spatial patterns present in hyperspectral data. The application of Ensemble Networks in hyperspectral image classification has gained increasing attention due to their effectiveness in handling the unique characteristics of hyperspectral data. Ensemble Networks have been shown to outperform traditional single-model approaches by harnessing the collective intelligence of diverse classifiers, including support vector machines, decision trees, neural networks, and spectral-spatial classifiers. In this section, we provide a review of Ensemble Networks used for hyperspectral image classification, focusing on their training ratio, results, contributions and shortcomings. Ensemble learning combines several models, such as classifiers or experts, to tackle a specific computational intelligence issue. Several ensemble models are combined in the literature to improve the classification accuracy and maintain the HSI computational time. Table 2.15 includes the models with improvisations made to each.

Table 2.15: Articles with Ensemble Network

S. No	Ref	Model Used	Training Ratio	Results	Contributions	Shortcomings
1	[164]	EMAP + Gabor + GLCM + NPE+ LFDA	**IP:** 0.10 **PU:** 0.01 **KSC:** 0.05	**Indian Pines**: OA = 98.46; AA = 98.39; KC = 98.24; **Pavia U**: OA = 99.48; AA = 99.62;KC = 99.32; **KSC**: OA = 96.81; AA =96.26; KC = 96.45	1. Multiview fusion using EMAP (extended multi-attribute profiles), Gabor filters and gray level cooccurrence matrix (GLCM) 2. LFDA to remove the redundant features and NPE to preserve the local manifold structure 3. Better than existing methods due to feature stacking	Computational time can be calculated and reduced.

Continued on next page

Table 2.15: Articles with Ensemble Network (Continued)

S. No	Ref	Model Used	Training Ratio	Results	Contributions	Shortcomings
2	[71]	Capsule Network + GAN + Multiscale Convolution	**PU:** 0.008 **SA:** 0.006 **KSC:** 0.06	**Pavia U**: OA = 97.98; AA = 98.01; KC = 97.32; **Salinas** : OA = 93.87; AA = 97.54; KC = 93.20; **KSC** : OA = 98.20; AA = 97.23; KC = 97.99	1. Capsule Network integrated with GAN to alleviate the gradient vanishing problem caused by conventional GANs 2. Octave convolution and Multiscale convolution to reduce the parameters 3. Multiscale fusion of spectral and spatial information	1. Higher training time than other state-of-the-art networks 2. Capsule Networks increase the computational cost
3	[165]	Bi-LSTM + Attention mechanism + Global-Local Pooling	**IP:** 0.10 **PU:** 0.34 **SA:** 0.02	**Indian Pines**: OA= 99.15; AA = 99.36; KC = 99.03; **Pavia U:** OA = 99.26; AA = 99.03; KC =99.01; **Salinas**: OA = 99.37; AA= 99.57; KC = 99.3	1. Integrated Bidirectional LSTM for spectral features and global-local CNN for spatial features 2. To counteract the effect of interfering information in the patch, the GL-PF module evaluates the correlation between alternative pooling algorithms.	1. The number of principal components and neighbourhood size is sensitive to each dataset
4	[166]	ARMA + GCN + Context Aware Learning + Cross Entropy	**PU:** 0.006 **SA:** 0.008 **HU:** 0.03	**Pavia U**: OA =97.71; AA = 97.56; KC = 98.24; **Salinas:** OA = 97.96; AA = 98; KC = 97.43; **Houston U:** OA = 93.04; AA = 92.68; KC = 92.36	1. To handle the computational cost involved in GCN, integrated autoregressive moving average filter to reduce the calculations 2. Solves over smoothing problem involved in GCN 3. An end-to-end model trained with context-aware learning to extract local information	1. Difficult feature extraction with ARMA if CAL is omitted 2. Computational complexity increases as the number of segmentations increase
5	[167]	CNN + Label Smoothing + Transfer Learning	**IP:** 0.02 **KSC:** 0.04 **HU:** 0.01	**Indian Pines**: OA = 91.88; AA = 77.37; KC = 90.28; **KSC**: OA = 99.27; AA = 98.87; KC = 99.19; **Houston U**: OA = 88.33; AA = 88.1; KC = 87.39	1. A network that joins CNN, Transfer learning, and ensemble networks by taking advantage of all three of them 2. An improved label smoothing technique to improve the generalization of the current approach	1. Classification Accuracy can be significantly increased 2. Improved Label Smoothing showed minimal improvement in accuracy
6	[168]	CNN + Ensemble Learning + Enhanced Random Feature Subspace	**IP:** 0.05 **PU:** 0.05 **SA:** 0.05	**Indian Pines:** OA = 97.57; AA = 96.23 KC = 97.23; **Pavia U:** OA = 98.48; AA = 98.37; KC = 97.98; **Salinas**: OA = 99.34; AA = 99.3; KC = 99.27	1. This study proposed a network using enhanced feature random subspace that works on imbalanced class distribution problem 2. Data enhancement using random subsampling technique 3. Use of ensemble techniques like bagging, boosting, and stacking	1. Computational time increases with the size of the ensemble, and features get bigger
7	[169]	Capsule Network + GAN+CNN +Augmentation +Conv LSTM	**IP:** 0.01 **HU:** 0.009	**Indian Pines**: OA= 72.31; AA = 81.67; KC = 68.74; **Houston U :** OA = 70.69; AA = 72.26; KC = 68.25	1. GAN to generate synthetic data for data augmentation 2. Discriminator embedded with capsule network and convolutional extended short tern memory model 3. Stable training using sparse constraint	1. Poor accuracy can be significantly increased

Continued on next page

Table 2.15: Articles with Ensemble Network (Continued)

S. No	Ref	Model Used	Training Ratio	Results	Contributions	Shortcomings
8	[170]	Few-shot learning + Quadruplet loss + dense and dilated convolution	**Trained on HU, KSC, BOT**	**Salinas**: OA = 95.58; AA = 95.85; KC =95.46; **Indian Pines**: OA = 82.77; AA = 82.65; KC = 82.59; **Pavia U**: OA = 94.57; AA = 94.71; KC = 94.42	1. This study proposed a quadruplet network that works on increasing the inter-class distance and reducing the intra-class distance 2. Use of dense and dilated convolution to extract the discriminative features 3. Proposal of the quadruplet loss function	1. Computing time to train the network could be further reduced 2. Accuracy could be better for the Indian Pines dataset
9	[171]	CNN + Transfer Learning	**IP:** 0.08 **PU:** 0.003	**Indian Pines**: OA = 94.65; AA = 96.62; KC = 94; **Pavia U**: OA = 97.86; AA = 98.13; KC = 97	1. Proposed an artificial labelling method of unsupervised training that works based on spatial location 2. Analyses clustering phenomenon and effect of noise addition in classes	1. Computing Time could be reduced
10	[172]	FCN + Patch Free Global Learning + Feature Fusion	**PU:** 0.04 **SA:** 0.06 **HU:** 0.18	**Pavia U**: OA = 99.81; AA= 99.83; KC = 99.74; **Salinas**: OA =99.92; AA = 99.91; KC = 99.91; **Houston U**: OA = 86.41; AA = 88.44; KC = 85.55	1. The proposed network includes a sampling technique, FCN based on encoder and decoder 2. Integrated FreeNet to increase the accuracy 3. A lateral connection between encoder and decoder for the fusion of spatial data	1. Lower accuracy for the Houston U dataset

EN achieving more than 95% Overall Accuracy

Figure 2.8: Network Diagram of studies having OA more than 96% with Ensemble for HSIC

Challenge: Limited Labelled Data

- Addressing the challenge of limited data availability, He et al. [167] employed transfer learning and an improved label smoothing technique to enhance the performance of CNNs. While their approach achieved impressive results exceeding 99% accuracy for the KSC dataset, it struggled to achieve comparable performance for IP and

Houston University (HU) datasets, highlighting the limitations of transfer learning in datasets with varying characteristics.

- Masarczyk et al. [171] also tackled the limited sample problem by introducing dense and dilated convolutions to extract discriminative features and employing quadruplet loss to enhance accuracy. While their approach shows promise in capturing complex spatial patterns, the effectiveness of the proposed method may vary depending on the dataset's characteristics and the selection of hyperparameters.

- Wang et al. [173] addressed the limited sample problem by leveraging GANs for data augmentation, generating synthetic samples to augment the dataset. Their proposed network, which incorporated convolutional LSTM and capsule networks within the GAN discriminator, showcases the potential of synthetic data generation for enhancing model performance. However, the reliance on generated data introduces challenges related to dataset distribution and the generalization of the model to real-world scenarios.

Challenge: Class Imbalance

- To mitigate the class imbalance problem, Lv et al. [168] introduced augmentation techniques and an enhanced feature subspace, constructing an ensemble learning model with CNNs serving as subnetworks for classification. Despite their efforts, the model's performance may still be hindered by the inherent challenges posed by imbalanced datasets, necessitating further exploration of robust techniques to address class imbalances effectively.

Challenge: Overfitting

- Building upon these efforts, Wang et al. [71] focused on alleviating overfitting and exploiting spectral-spatial information by integrating capsule networks and GANs. Their approach, which fused spectral and spatial information using contexture relations, demonstrated improved efficiency and stability by incorporating capsule networks within the GAN discriminator. Despite these advancements, the model's performance may be influenced by the choice of hyperparameters and the complexity of the spectral-spatial relationship representation.

2.2.6 Review of Studies based on Generative Adversarial Networks

Generative Adversarial Networks (GANs) have emerged as a promising approach for addressing challenges in hyperspectral image classification. By harnessing the power of adversarial training, GANs enable the generation of synthetic hyperspectral images that closely resemble real-world data, thereby facilitating data augmentation and improving

classification performance. In this section, we explore literature involving GANs for synthetic data generation in HSIC domain.

Table 2.16: Articles with GAN as the base network

S. No	Ref	Model for Data Gen	Model Used	Training Ratio	Results	Contributions	Shortcomings
1	[174]	GAN	3D Patch Extractor + 3D Discriminator +3D Classifier	**IP:** 0.05 **PU:** 0.05 **KSC:** 0.05	**Indian Pines**: OA = 86.96; AA = 78.72; KC = 85.17; **KSC**: OA = 95.31; AA = 92.26; KC = 94.78; **Pavia U**: OA = 93.9; AA = 93.29; KC = 91.86	1. Proposed a network based on GAN for class imbalance problems. This network generates samples using over sampling strategy with the use of a generator and feature mapping unit 2. A discriminator that differentiates real and fake samples Uses a low amount of training samples	1. Accuracy can be increased 2. Poor generalization of the KSC dataset 3. Performance of the proposed dataset improves with a higher degree of imbalance
2	[175]	GAN	Adaptive weighting feature fusion	**IP:** 0.10 **PU:** 0.01	**Indian Pines**: OA = 97.83; AA = 98.30; KC = 97.53; **Pavia U**: OA = 98.68; AA = 98.81; KC = 98.12	1. Proposed network with discriminator containing adaptive spectral-spatial combination pattern 2. Four sets of filter banks for performance improvement 3. Center loss and Mean minimization loss encapsulated	1. Bigger Kernel size causes overfitting 2. Yields best accuracy with no unlabeled samples
3	[176]	GAN	Domain Adaptation + Cross Dataset	**Trained on PU**	**Botswana**: OA: 89.65;AA = 87.14; KC = 82.35; **Pavia C**: OA = 88.25; AA = 87.93; KC = 84.92; **Salinas**: OA = 86.55; AA = 87.3; KC = 81.96	1. Proposed a network based on GAN that encapsulates variational encoder that helps to learn data across domains 2. Two alignment strategies to align different domains 3. A GAN with generator and discriminator that trains on source dataset and tests on target data	1. Accuracy can be increased for all three datasets, including Botswana, Pavia C, and Salinas 2. Takes longer training time
4	[115]	GAN	Attention mechanism + Convolutional LSTM	**IP:** 0.05 **PU:** 0.03	**Indian Pines**: OA = 97.4; AA = 95.2; KC = 97; **Pavia U**: OA = 99.2; AA = 98.6; KC = 99.2	1. Proposed a convolutional GAN to alleviate the overfitting issue 2. Generator generates samples using competitive and collaborative learning 3. Attention module to generate discriminative features	1. Sampling strategies to reduce the overlap between training and testing sets 2. Determining the best values of the number and position of modules is a challenge
5	[177]	GAN	Active Learning	**IP:** 0.02 **PU:** 0.006	**Indian Pines**: OA = 91.41; AA = 93.2; KC = 90.2; **Pavia U**: OA = 93.47; AA = 88.97; KC = 91.34	1. Proposed a network based on GAN and active learning able to work with limited samples 2. Integrates Acquisition heuristic for better feature discrimination that is adversarially learned with high-level features	1. Overall Accuracy could be improved

Continued on next page

Table 2.16: Articles with GAN as the base network (Continued)

S. No	Ref	Model for Data Gen	Model Used	Training Ratio	Results	Contributions	Shortcomings
6	[178]	GAN	Adversarial Learning + CNN + Multiclass Spectral Spatial	**IP:** 0.05 **PU:** 0.03 **KSC:** 0.01	**Indian Pines**: OA = 95.6; AA = 91.4; KC = 95;**Pavia U**: OA = 99.2; AA = 98.4; KC = 98.9;**Salinas**: OA = 99.1; AA = 99.3; KC = 99	1. An end-to-end network for multiclass classification with unified sample generation, feature extraction, and classification 2. Uses adversarial learning to enhance the discrimination further, works on a small sample problem	1. Training Time could be reduced 2. Unable to deal with a class imbalance distribution problem

GAN with overall accuracy between 99-100%

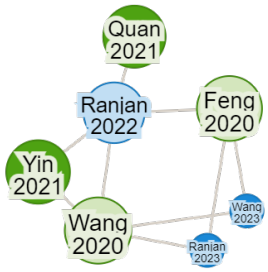


Figure 2.9: Network Diagram depicting studies achieving OA over 95% using GAN for HSIC - *(Current works highlighted in green, future works in blue)*

Challenge: Enhance Classification

- Feng et al. [115] integrated a complex attention module into the generator, aiming to eliminate noisy aspects in samples. They extended the network by incorporating a convolutional LSTM into the discriminator to extract spectral-spatial features.
- Yin et al. [34] introduced a novel network architecture with a discriminator to generate improved samples, enhancing information discrimination and classification by considering both spectral and spatial information.
- Wang and Ren [177] introduced adversarial active learning, establishing an acquisition heuristic to guide the selection of informative samples for training. This approach optimizes the learning process by focusing on the most informative data points, contributing to overall classification performance.

Challenge: Class Imbalance

- To tackle the issue of class imbalance, Quan et al. [174] proposed a comprehensive network architecture comprising a generator, discriminator, and classifier. This architecture effectively addressed class imbalance distribution by generating samples, classifying them as real or fake, and further categorizing them into several classes. Additionally, Feng et al. [115] addressed redundant spectral bands and limited samples by generating synthetic data and employing semi-supervised training with various fusion strategies.

2.2.7 Review of studies based on LSTM

LSTM, a specialized form of RNN, plays a crucial role in addressing short-term memory challenges and is widely utilized in diverse fields, including natural language processing, speech recognition, and HSI classification. Singh and Singh [179] introduced a novel framework focusing on the spectral properties of HSI, utilizing LSTM as a classifier. This approach excels in handling sequential problems, showcasing the effectiveness of LSTM in capturing temporal dependencies within hyperspectral data. Further advancements in utilizing LSTM for HSI classification were made by integrating traditional low-level features to extract sequential features from central samples [180]. This integration enhances LSTM's ability to learn high-level semantic features, thereby significantly improving classification accuracy. However, concerns regarding Convolutional LSTM's excessive number of parameters and high storage requirements were addressed by Hu et al. [181] within a unified framework, emphasizing the importance of optimization techniques in model design and implementation. Yin et al. [34] presented an innovative approach by optimizing parameters for two feature extractors: Bi-LSTM and 3D CNN. The Bi-LSTM extractor focuses on capturing spectral features to understand relationships between spectral bands, while the 3D CNN extractor emphasizes spatial-spectral features. Both extractors undergo optimization using a carefully designed loss function, highlighting the significance of feature extraction methods in enhancing HSI classification performance. Building on this, Ma et al. [182] proposed a method that goes beyond spectral data alone by employing pixel-wise and block-wise similarity measurements to select sequence candidates. This approach provides a distinctive perspective on leveraging LSTM for enhanced HSI classification, emphasizing the importance of considering spatial relationships between pixels. Table 2.17 summarizes models with LSTM.

Table 2.17: Articles with LSTM as base network

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
1	[183]	LSTM	Attention Mechanism + Auto-correlation mechanism	**PU:** 0.05 **IP:** 0.10 **SA:** 0. 05 **HU:** 0.10	**Pavia U**: OA = 99.35; AA= 99.14; KC = 98.87; **Indian Pines**: OA =97.62; AA = 96.93; KC = 97.2; **Salinas**: OA = 99.42; AA = 99.23;KC =98.9; **Houston U**: OA = 95.26; AA = 96.69; KC = 93.79	1. LSTM takes spectral bands as input values along with spatial information employed by non-local diverse regions. 2. Attention mechanism is integrated to enhance discrimination and reduce the number of parameters 3. An autocorrelation mechanism enhances the bands' discriminative information and suppresses the bands' weak information.	1. Dimensionality reduction technique could have reduced the bands
2	[184]	LSTM	Attention Network	**SA:** 0.10 **PC:** 0.10 **PU:** 0.10	**Salinas**: OA = 98.12; AA =98.88; KC =97.91; **Pavia C**: OA= 99.73; AA= 99.2; KC =99.62; **Pavia U**: OA = 98.71;AA= 97.63; KC= 98.29	1. Explored the bi-directional correlation of hyperspectral pixels using bidirectional LSTM 2. An attention mechanism to explore the spatial features 3. Reduces redundancy and highlights discriminative features	1. LSTM uses more parameters and floating point operations than CNN
3	[185]	LSTM	Pixel level scanning + Multi scanning strategy	**IP:** 0.10 **PU:** 0.10 **SA:** 0.20	**Indian Pines**: OA= 95.38; AA =95.45;KC=95; **Pavia U**: OA= 99.18;AA= 98.81; KC =98; **Salinas**: OA =97.92; AA =97.67; KC = 97	1. The multi-scanning strategy used by RNN takes fewer parameters than the most groundbreaking model, CNN 2. Converts and image into sequential data with different scanning directions 3. Good classification accuracy, lesser parameters	1. Multi directions for the multi-scanning need to be finalized 2. Spectral-spatial models lead to higher accuracy
4	[186]	LSTM	PCA + Attention mechanism + Multiscale convolution + Softmax	**IP:** 0.15 **PU:** 0.10 **KSC:** 0.10	**Indian Pines**: OA= 88.25; AA =89.48; KC=86.5; **Pavia U**: OA = 95.43; AA =94.83; KC = 93.92; **Salinas**: OA =95.93; AA =97.8; KC =96.66	1. PCA to reduce the dimensions and attention mechanism to enhance the feature selection 2. Multiscale convolution to enhance feature mining 3. Bi LSTM for feature integration 4. Softmax for multiclass classification	1. Indian pines has the most imbalance class distribution problem; the accuracy can be increased
5	[182]	LSTM	Pixel Matching +Block Matching + Spatial Similarity Measurements	**SA:** 0.06 **PU:** 0.04	**Salinas**: OA= 90.63; AA =93.95;KC=89.55; **Pavia U**: OA= 96.2; AA =94.65; KC =94.91	1. Proposed model uses LSTM for constructing sequential features from a single image 2. Pixel matching and block matching to select sequence candidates 3. Investigates Euclidean distance and spectral angle mapping	1. Pixel matching method is sensitive 2. Over-smoothing problem 3. Cross-domain research could be a topic for future

Continued on next page

Table 2.17: Articles with LSTM as base network (Continued)

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
6	[187]	LSTM	Gated Recurrent Unit	**IP:** 0.07 **PU:** 0.09	**Indian Pines**: OA = 91.79; AA = 95.94; KC =90.62; **Pavia U**: OA =90.3; AA = 87.97; KC = 86.26	1. Gated Recurrent Unit to eliminate the redundant information and to learn complementary information from the non-adjacent bands 2. Takes advantage of both spectral and spatial information by integrating them	1. Due to the spatial feature not being fully accessible, accuracy for both datasets is not very high
7	[33]	LSTM	PCA + Spatial LSTM + Softmax + Decision Fusion strategy	**IP:** 0.10 **PU:** 0.09 **KSC:** 0.09	**Indian Pines:** OA =95; AA =91.69; KC=94.29; **Pavia U**: OA =98.48; AA = 98.51;KC =97.56; **KSC**: OA =97.89; AA = 97.28; KC =97.65	1. LSTM to address the problem of spectral feature extraction and spatial feature extraction 2. Patch centered at the pixel was used for the spatial feature extraction	1. Works well when fed both spectral and spatial information

2.2.8 Other Prominent Studies Encountered

In the survey, other encountered models with fewer published articles include attention networks, machine learning models (such as random forest, support vector machines, K-nearest neighbour), active learning, transformer networks, and few-shot classification models like Siamese, deep metric learning, deep belief network, extreme learning machine, and domain adaptation. Notably, attention networks demonstrated the highest overall accuracy (97%), followed by the transformer network (96%) and machine learning and active learning models (95%). Few studies addressed deep metric learning, deep brief learning, extreme learning machines, and domain adaptation, reflecting the accuracy of each study rather than an average overall accuracy. Additionally, models such as Deep-Q network, Low-Rank learning, Shared Subspace learning, CliqueNet, Sparse representation classifier, and Kernel space representation with Brownian descriptor were encountered with minimal data. Figure 2.10 illustrates the learning modes utilized by each deep learning model, revealing a predominant preference for supervised learning, followed by semi-supervised learning. Unsupervised learning, although less popular, has gained traction with advancements like Generative Adversarial Networks (GAN), facilitating the generation of synthetic data to mitigate challenges posed by small labelled datasets. Tables 2.18, 2.19, 2.20, 2.21, and 2.22 encompass the literature on attention networks, machine learning models, active learning, transformer networks, few-shot learning, and deep metric learning, respectively.

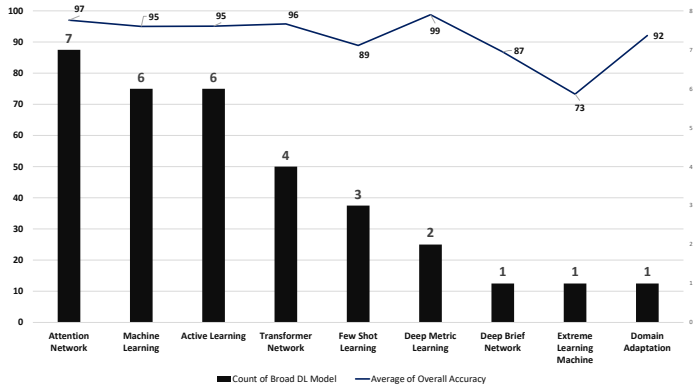


Figure 2.10: Number of other prominent models (in black) encountered in the survey with their average overall accuracy

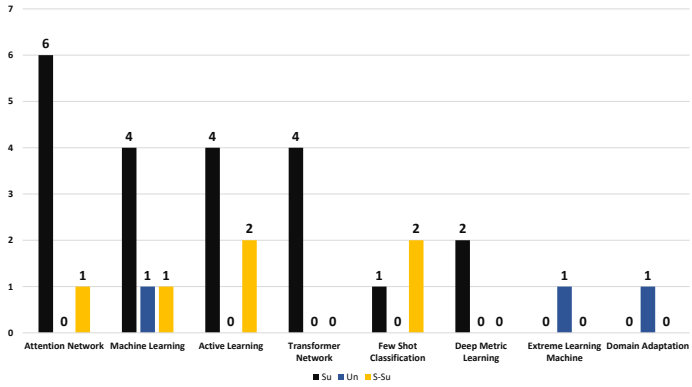


Figure 2.11: Most used training modes used in the models (Supervised/Unsupervised/Semi-Supervised)

Table 2.18: Articles with Attention Networks

S. No	Ref	Additions to model	Training Ratio	Results	Contributions	Shortcomings
1	[188]	Band Attention + Spatial Attention +Feature Fusion	**HU:** 0.05 **KSC:** 0.05	**Houston U**: OA= 89.13; AA= 90.31; KC = 88.36; **KSC**: OA = 98.6; AA = 98.02; KC = 98.42	1. The proposed network reduces interfering pixels to extract more discriminative features 2. Band attention and spatial attention to reducing the duplicate features	1. Spectral and spatial extraction individually does not result in good performance
2	[189]	3D Spatial and Spectral convolution + Dense Connection	**IP:** 0.19 **PU:** 0.19 **HU:** 0.06	**Indian Pines**: OA = 99.67; AA = 99.55; KC = 99.62; **Pavia U**: OA = 99.81; AA = 99.75; KC = 99.75; **Houston U**: OA = 96.64; AA = 95.43; KC = 96.03	1. Dense network to extract the features comprehensively and avoid overfitting 2. 3D Spectral and 3D Spatial convolution replaced by the 3D convolution 3. Attention mechanism to further improve discrimination	1. Needs more computational power 2. Higher classification accuracy with lesser training samples can be achieved in the future

Continued on next page

Table 2.18: Articles with Attention Networks (Continued)

S. No	Ref	Additions to model	Training Ratio	Results	Contributions	Shortcomings
3	[190]	Multiway Spatial attention and spectral attention	**PU:** 0.03 **KSC:** 0.05 **PC:** 0.08	**Pavia U**: OA =98.03; AA =97.51; KC= 97.39; **KSC**:OA =98.4; AA = 97.22; KC =98.21; **Pavia C**: OA = 98.08; AA= 93.93;KC =97.29	1. The proposed network contains a spectral, spatial, and semantic module that filters out unimportant information 2. Focuses on features with informative areas 3. Multiscale semantic module to help classify features	1. Great accuracy for other datasets, but classification accuracy for the newly explored HUOSHAOYUN mineral dataset can be increased
4	[191]	Spectral Feature Fusion + CNN	**IP:** 0.05 **PU:** 0.01 **SA:** 0.01	**Indian Pines**: OA = 92.51; AA = 94.22; KC =91.02; **Pavia U**: OA = 97.09; AA =96.51;KC= 96.15; **Salinas**: OA =96.16; AA=98.48; KC=95.72	1. Novel spectral feature fusion combined with Groupwise spectral classifier 2. Dual Channel attention to enhance discrimination 3. Joint loss function to enhance classification 4. Works well for limited data	1. Accuracy can be increased for the Indian Pines dataset 2. Involves much more parameters due to the dual attention module
5	[192]	Grouped convolutions + Spatial attention + Spectral Partition	**IP:** 0.05 **PU:** 0.005 **SA:** 0.005	**Indian Pines**: OA = 98.29; AA= 97.85; KC = 98.05; **Pavia U**: OA= 97.92; AA= 96.95; KC = 97.23; **Salinas**: OA = 97.46; AA = 98.45; KC = 97.17	1. The proposed network consists of spectral partition, feature extraction, fusion and classification stage 2. A novel spatial attention module to maintain performance while choosing a bigger patch size	1. The choice of patch size is sensitive to the dataset 2. Larger path size includes interfering pixels as well
6	[193]	Spectral bidirectional RNN + Spatial attention CNN	**PC:** 0.12 **PU:** 0.02 **IP:** 0.10	**Pavia C**: 99.69; 98.31; 99.18; **Pavia U**: 99.24; 98.07; 98.17; **Indian Pines**: 99.67; 99.08; 98.37	1. Spectral bidirectional attention and spatial attention convolution to extract spatial-spectral features 2. Extracts homogeneous discriminative features	1. Misclassification in few classes where classes have similar features in the spectral or spatial domain
7	[45]	Channel wise attention + Spatial Attention	**IP:** 0.03 **PU:** 0.04 **SA:** 0. 005 **BOT:** 0.012	**Indian Pines**: 95.38; 96.47; 94.74; **Pavia U**: 96; 96.45; 94.67; **Salinas**: 97.51; 98; 97.23; **Botswana**: 96.24; 96.74; 95.93	1. An end-to-end network with no need for feature engineering 2. Double Branch Dual-Attention mechanism that extracts discriminative features 3. Works well with limited data 4. Lesser time complexity	1. Accuracy can be further increased 2. Training Time can be reduced

Table 2.19: Articles with Machine Learning Models

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
1	[194]	Random Forest + Markov Random Field	Superpixel segmentation + Regularization + Fusion	**IP:**0.12 **SA:**0.012 **HU:**0.012	**Indian Pines**: OA = 94.97; AA = 86.85; KC = 94.25; **Salinas**: OA = 97.6; AA = 97; KC = 97.33; **Houston U:** OA = 84.94; AA = 86.53; KC = 83.68	1. The proposed network combines super pixel segmentation and correlation partitioning 2. Random Forest with Markov Random field fused as a classifier	1. Accuracy for the Houston U dataset can be increased 2. High computational complexity

Continued on next page

Table 2.19: Articles with Machine Learning Models (Continued)

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
2	[195]	SVM	PCA+ Spectral-Spatial feature extraction Multiscale Fusion	**IP:**0.02 **KSC:**0.016	**Indian Pines**: OA = 96.11; AA= 96.13; KC = 95.6; **KSC**: OA = 94.74; AA = 94.74; KC = 94.2	1. Combines PCA, segmented PCA, and two-dimensional spectrum analysis fusion method for feature extraction and classification jointly. 2. SVM is used as a classifier to evaluate the performance	1. High Computational complexity due to a large number of embedded windows
3	[196]	MLR/SVM	Data Augmentation	**IP:**0.015 **PU:**0.004	**Indian Pines**: OA = 87.76; AA= 86.12; KC = 86.2; **Pavia U:** OA = 85.85; AA = 86.69; KC = 83.99	1. A data augmentation method that studies the effectiveness of both labelled and unlabeled samples 2. Feature extraction method that works with small datasets	1. Feature extraction method can combine spatial information to enhance the classification accuracy 2. Classification accuracy is low
4	[197]	SVM with composite kernel (SVM-CK)	Spectral-Spatial Graph + Novel Distance Metric	**IP:**0.12 **PU:**0.016 **SA:**0.013	**Indian Pines**: OA = 98.18; AA=98.67; KC = 97.9; **Pavia U**: OA = 97.66; AA =97.92; KC = 96.9;**Salinas**: OA =98.9; AA= 99.27; KC = 98.9	1. Clustering hyperspectral data into clusters using superpixel segmentation 2. A novel distance metric was introduced to calculate the pixel similarity 3. A graph that preserves local information fed into SVM-CK for classification	1. Few misclassifications occur due to spatial information involved in graph construction 2. Computation cost is nearly lower and can be lowest by optimizing the existing algorithm
5	[198]	Watershed Classifier/ RF/SVM	Triplet Loss	**IP:**0.10 **PU:**0.10 **KSC:**0.10	**Indian Pines**: OA = 82.00; AA= 77.36; KC = 79.41; **Pavia U**: OA =94.19; AA =92.38; KC =92.29;**KSC**: OA =81.27 ; AA = 72.90 ; KC = 79.09	1. The proposed novel algorithm uses fewer parameters 2. The proposed network can be used for both supervised and semi-supervised learning without any modifications 3. Can be extended to classifiers like random forest and SVM as well	1. Deep learning models in the same paper provided better results than classifiers like SVM or RF
6	[199]	KNN and Guided Filer	Spectral-Spatial Information	**IP:**0.05 **PU:**0.05 **KSC:**0.05 **SA:**0.05	**Indian Pines**: OA = 98.76; AA =97.73; KC = 97.41; **Pavia U:** OA = 99.26; AA = 99.03; KC = 98.95;**KSC**: OA =99.76; AA = 99.63; KC = 99.73; **Salinas**: OA = 99.9; AA= 99.85; KC = 99.89	1. KNN with a guided filter that can extract spatial information and enhance discriminative ability by using an edge-preserving filter 2. Dimensionality reduction to reduce the dimensions and consider relevant bands	1. Selection of filters is sensitive to datasets 2. Other classification models can be tried

Table 2.20: Articles with Active Learning

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
1	[200]	Active Learning	Candidate Set + Pseudo Labels + Clustering	**IP:**0.019 **SA:**0.003	**Indian Pines**: OA = 67.8; AA = 65.7; KC = 64.2; **Salinas**: OA = 85.6; AA= 89.5; KC = 84	1. Cluster-inspired active learning for limited labeled samples 2. Formation of the candidate set using modified clustering with fast search 3. K-means clustering-based pseudo-labeling scheme for unlabeled data	1. Extremely low classification accuracy. A large room for improvement left in improving the classification accuracy.
2	[201]	Active Learning	Multiview Learning + Representation Learning+ LOCO Active Learning	**SA:**0.005 **BOT:**0.09 **KSC:**0.05	**Salinas**: OA = 88.47; AA =92.28; KC = 87.13; **Botswana**: OA = 92.14; AA =92.25; KC = 91.49; **KSC**: OA = 90.61; AA = 86.22; KC = 89.52	1. Proposed "leave one class out" (LOCO) strategy for limited samples 2. A sample selection strategy for ranking the training contribution of candidate set and training set 3. Reduces the time cost by decreasing the target set	1. Run time is lower but still higher than state-of-the-art models. This run time could be reduced. 2. Use of semi-supervised learning to reduce sample costs.
3	[202]	Active Learning	Densely Connected Block	**PU:**0.01 **IP:**0.05 **SA:**0.009	**Pavia U**: OA = 99.28; AA =98.07; KC = 99.05; **Indian Pines**: OA = 99.75; AA =99.74; KC = 99.71; **Salinas**: OA = 99.67; AA =99.38; KC = 99.63	1. The proposed deep, densely connected network is derived from DenseNet121 2. Integrated active learning to add selected samples to reduce labelling costs and improve classification accuracy. 3. Works for a small sample set	1. To enhance the accuracy, the classes with fewer samples in the Indian Pines dataset were removed 2. Training samples could be further reduced
4	[62]	Active Learning	Multiple Feature Representation + Stacked Sparse Autoencoder(SSAE) + Transfer Learning	**PU:**0.20 **PC:**0.20 **SA:**0.20	**Pavia U**: OA =99.61; AA = 99.61; KC = 99.48; **Pavia C**: OA = 99.86; AA = 99.62; KC = 99.8; **Salinas**: OA = 98.61; AA = 99.27; KC = 99.45	1. Hierarchical stacked sparse encoder to extract the spectral-spatial features 2. Fine-tuning of SSAE using limited labeled samples using active learning 3. Works on cross datasets and intra-image	1. Works even better when land covers are the same (for instance, Pavia C and Pavia U) and not for datasets with different land covers (Indian Pines and Salinas) 2. Parameters of hierarchical SSAE and active learning could be optimized 3. Can be applied to data across sensors in the future

Continued on next page

Table 2.20: Articles with Active Learning (Continued)

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
5	[59]	Active Learning	Gabor Filtering + Morphological Profiles + Overall Error Probability	**IP:**0.04 **PU:**0.01 **KSC:**0.07	**Indian Pines**: OA = 99.5; AA =99.63; KC = 99.03; **Pavia U**: OA = 99.84; AA =99.93; KC = 99.79; **KSC**: OA =99.53; AA = 99.69; KC = 99.47	1. Feature-driven active learning to deal with small labelled datasets 2. Overall error probability and Fisher ratio to assess the discriminative ability of feature space	1. Active learning method integrated with the Gabor filter gives a few misclassifications. 2.

Table 2.21: Articles with Transformer Network (TN)

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
1	[203]	TN	Local Contextual Information + Sequence Data + Skip Fusion + Transformer	**IP**: 0.07 **PU**: 0.09	**Indian Pines:** OA = 81.76; AA = 87.81; KC = 79.19; **Pavia U:** OA = 91.07; AA = 90.2; KC = 88.05	1. The proposed network is based on transformers that can learn local spectral data in a sequential form from the neighbouring pixels. 2. Cross-layer skip connections to add soft residuals across layers.	1. High network complexity could be reduced while maintaining classification performance. 2. Accuracy can be improved. 3. Paying more attention to the connected and skipped encoders.
2	[204]	TN	CNN + Semantic Features + Spectral-Spatial Tokenization + Transformer	**IP:** 0.09 **PU:** 0.05	**Indian Pines:** OA = 97.47; AA = 96.57; KC = 97.11; **Pavia U:** OA = 99.21; AA = 98.69; KC = 99.15	1. The proposed transformer-based network can extract spectral-spatial features and high-level semantic features. 2. Gaussian tokenizer for feature transformation into semantic tokens.	1. Classification accuracy is highly dependent on setting the right patch size and taking the correct number of tokens, sensitive to each dataset. 2. Poor semantic modeling for Houston U dataset due to sample points being discrete.
3	[205]	TN	Factorized architecture search + Spatial Attention + Spectral-Spatial Transformer	**IP:** 0.019 **PU:** 0.005	**Indian Pines:** OA = 94.39; AA = 92.77; KC = 93.58; **KSC:** OA = 97.3; AA = 94.97; KC = 96.99; **Pavia U:** OA = 98.02; AA = 96.67; KC = 97.37	1. Incorporated transformer blocks with spatial attention and spectral association modules. 2. Focuses on the operations involved layer-wise and sequential orders block-wise to increase the classification performance further.	1. Training time can be further reduced. 2. Could be combined to be an end-to-end network.
4	[206]	TN	Attention mechanism + Long distance dependence	**SA:** 0.20 **IP:** 0.20 **PU:** 0.20	**Salinas:** OA = 99.91; AA = 99.63; KC = 99.81; **Indian Pines:** OA = 99.22; AA = 99.08; KC = 98.85; **Pavia U:** OA = 99.64; AA = 99.67; KC = 99.47	1. The proposed network incorporates a spectral attention module and a self-attention module. 2. Multiple encoders connected to the multi-residual structure to avoid information loss. 3. Solves the problem of long-distance dependence.	1. The proposed network can be customized to perform on the imbalance class distribution problem.

Table 2.22: Articles with Few-Shot Learning

S. No	Ref	Model	Additions to model	Training Ratio	Results	Contributions	Shortcomings
1	[207]	FSL	Active Learning + Prototypical Learning + Adversarial autoencoder	**IP:**0.004 **PU:**0.0001	**Indian Pines**: OA = 62.18; AA = 70.72; KC = 57.04; **Pavia U**: OA = 69.84; AA = 68.01; KC =59.85	1. Improved prototypical learning proposed 2. Active learning incorporated with the proposed network for few-shot classification	1. Classification accuracy can be highly improved 2. Network takes an extremely high time to train the network
2	[208]	FSL	Attention mechanism + Residual Network + Feature-wise transformation + Meta learning	**PU:**0.005 **IP:**0.03 **SA:**0.007	**Pavia U**: OA = 91.77; AA = 87.95; KC = 89.21; **Indian Pines**: OA = 85.21; AA = 77.44; KC = 83.31; **Salinas**: OA = 96.57; AA =97.08; KC = 96.18	1. The proposed network can handle cross-domain datasets 2. Residual learning to effectively discriminate the features 3. Efficient Channel Attention module reduces parameters and decreases the computational complexity	1. Classification accuracy can be further improved 2. Especially for datasets like Indian pines with imbalanced class distribution, classification accuracy could have been better
3	[209]	FSL	3DCNN + Residual Learning	**PU:**0.009 **PU:**0.030 **IP:**0.04 **SA:**0.007	**Pavia U:** OA= 98.62; AA = 98.54; KC =98.17; **Pavia C**: OA =99.61; AA =99.19; KC = 99.45; **Indian Pines**: OA = 98.35; AA =99.03; KC = 98.07; **Salinas**: OA = 97.81; AA =99.25; KC =97.56	1. Proposed a network based on FSL that can increase the intra-class distance and decrease the inter-class distance 2. Residual learning to better extract discriminative features	1. Classes with fewer samples in the Indian Pines dataset could have been included 2. Can be generalized to other datasets as future work 3. The network is sensitive to parameters like episodes, learning rate, classes, and query number.

2.3 Discussion and Overall Analysis

This section introduces three graphs to assess the performance of deep learning models in HSI classification. Figures 2.12 and 2.13 are interconnected as they illustrate the same deep-learning models. Figure 2.12 highlights CNN as the predominant deep learning model, representing more than half of the publications in the survey list. Most models utilizing CNN as the foundation achieved an accuracy of over 90%. The proportion of models with less than 90% accuracy was notably lower. CNN, with its multiple convolutional filters and efficacy in classifying HSI, is considered an ideal choice for hyperspectral image classification. However, its application to hyperspectral images presents challenges due to strongly correlated bands and limited training samples. Recent CNN-based models address these challenges by mitigating overfitting and reducing network parameters to enhance computational efficiency.

Figure 2.13 provides a closer examination of the same set of models. The horizontal axis illustrates the frequency distribution of OA, while the vertical axis indicates the number of articles falling within each OA range. The linear trendline indicates that graph convolutional networks consistently achieved the highest OA, followed by autoencoders,

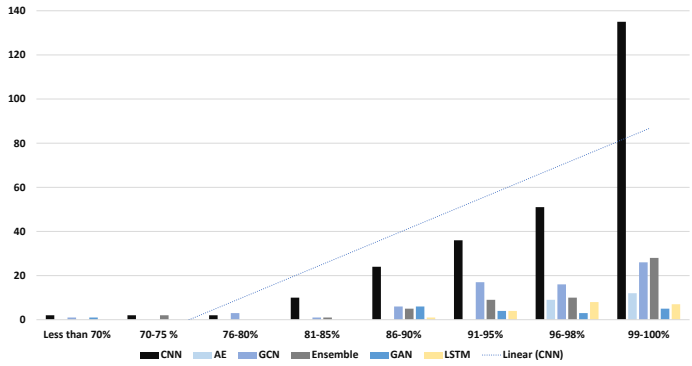


Figure 2.12: Frequency Distribution Chart showing the spread of Deep Learning Models

extended short-term memory models, and generative adversarial networks. While the number of articles based on GAN and LSTM was lower than those on GCN and Ensemble networks, autoencoders as a feature extraction technique outperformed GAN in synthetic data production.

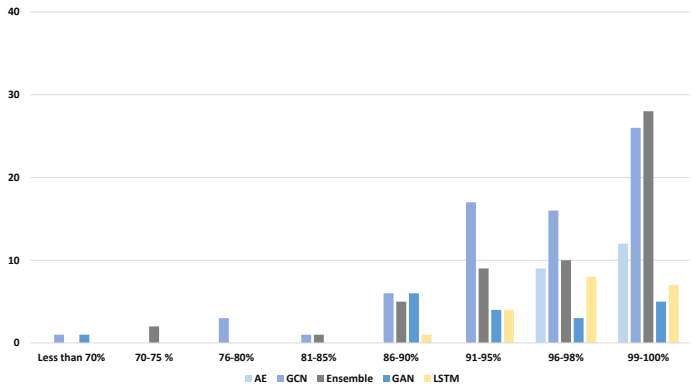


Figure 2.13: Frequency Distribution Chart showing the spread of Deep Learning Models (Exclusion: CNN)

Based on the studies discovered, the models are classified as supervised, semi-supervised or unsupervised learning methods. The next section delves into these learning styles and discusses relevant literature specific to these learning styles.

2.4 Summary of challenges and solutions proposed in literature

In our study, we found that the classification of HSI involves many challenges such as the presence of interfering pixels or tradeoff between high dimensionality and limited labelled samples in the case of supervised classification. Moreover, the role of unsupervised classification is also limited because of the uncertainty around the number of spectral classes

in datasets. The major issues that researchers working in HSI encounter, as well as the solutions provided for them, are detailed as follows:

1. **Limited Labelled Samples**: One of the biggest challenges in this field is the scarcity of labelled samples [175, 177, 210, 93, 211, 37, 30, 92, 212, 196] that leads to inadequate performance of the deep network. Because data labelling is time-consuming and labour-intensive, classification performance suffers due to the restricted number of labelled datasets. There are methods with which the problem of limited samples has been reduced, listed below:

    - **Data Augmentation**: Augmentation is a process by which new virtual samples are created to overcome the problem of limited samples [196]. There are two viable options for creating new samples include transformation [213, 214] and mixture-based strategies [215, 216].
    - **Transfer Learning**: The information gained in the source domain is transferred to the target domain through transfer learning. Transfer learning is used to tackle the problem when labelled data is insufficient. There are studies [217, 116, 167, 218, 219, 120, 220, 221, 222] that apply transfer learning across domains and operate on unsupervised or supervised samples. It uses pre-trained networks with initial values of network parameters already defined.
    - **Feature Learning**: Feature learning aims to extract useful features from an amount of unlabeled data. A lot of studies [177, 223, 224, 93, 225, 226, 158] incorporated an unsupervised or semi-supervised type of feature learning. In the unsupervised type, unlabeled data is trained and fed into a deep network. On the other hand, semi-supervised learning allows the features to be extracted, transferring this knowledge and merging with trained labelled data. Thereby, intermediate features are reconstructed using labelled data.
    - **Network Optimization**: This strategy improves the network's performance by changing the parameters. There are different ways proposed in the literature to optimize network performance. The use of the residual layer to make the most use of discriminative features has been proposed in several studies [136, 227, 228, 229, 230, 35, 231].

2. **Hughes Phenomenon or Curse of Dimensionality**: HSI presents a unique challenge due to its high dimensionality, with numerous spectral bands, coupled with the limited availability of labelled training data. This combination often leads to the Hughes Phenomenon or curse of dimensionality, where traditional classification algorithms struggle to effectively utilize the available data for accurate classification [229, 143, 232]. The high dimensionality of HSI data exacerbates the problem by

increasing computational complexity and making it difficult for algorithms to discern meaningful patterns without a sufficiently large amount of labelled training data. Consequently, the inadequate number of labelled samples further compounds the issue, resulting in suboptimal training and a subsequent drop in classification accuracy. These challenges are particularly pronounced in HSI applications, where the acquisition and annotation of labelled data can be costly and time-consuming. As a result, there is a need for novel methodologies and techniques to address the inherent limitations posed by the high dimensionality of HSI data and the scarcity of labelled training samples.

3. **Imbalanced Class Distribution**: In HSI, imbalanced class distribution poses a significant challenge [174, 122]. Due to the nature of HSI data, where certain classes may be rare or underrepresented, traditional classification algorithms struggle to accurately classify minority classes. This imbalance can lead to biased models that prioritize majority classes at the expense of minority ones, resulting in poor performance on crucial but less frequently occurring classes. Addressing this imbalance is essential for developing robust and reliable HSI classification systems. While techniques such as Synthetic Minority Over-sampling Technique (SMOTE) [174] and weighted ensemble methods have been proposed to mitigate this issue, further advancements are needed to ensure accurate classification across all classes, especially in scenarios where data scarcity exacerbates the imbalance problem.

4. **Generalization to other datasets or Inter-Domain Adaptability**: Achieving robust classification performance across different hyperspectral datasets presents a significant challenge in HSI classification. Annotating samples for training data is inherently labor-intensive and time-consuming, limiting the availability of labeled datasets. Moreover, pre-trained models may struggle to generalize to new datasets or encounter difficulties when faced with previously unseen classes [233, 30].

5. **Supervised training is time-consuming and labour intensive**: Regarding HSI, annotating labelled samples costs too much time and effort, which is considered one of the most challenging aspects of hyperspectral image analysis [93]. Considering single-level characteristics in a single layer leads to information loss. However, employing many networks to acquire multi-level features is computationally costly.

6. **Extracting more discriminant features or refined spatial-spectral features**: Maximizing the discriminative power of hyperspectral data poses a significant challenge in HSI classification tasks. The abundance of spectral and spatial information available in HSI datasets necessitates sophisticated techniques to extract and leverage relevant features effectively [234, 235, 34, 236].

7. **Challenges related to CNN in HSIC**: Below are some challenges related to the most prominent model in HSIC:

    - **CNN's limitation in modelling sample relations**: The most extensively utilized network for HSIC is CNN, which has limitations in modelling sample correlations. The inherent nature of hyperspectral imagery, characterized by high-dimensional data with intricate spectral-spatial relationships, often exceeds the capacity of traditional CNN architectures to effectively encode these correlations [237, 206, 211, 117].
    - **CNNs need longer training time**: One of the challenges associated with CNN in HSIC is the extended duration required for training. The depth and complexity of CNN architectures, coupled with the high-dimensional nature of hyperspectral data, often result in prolonged training times. This extended duration can hinder the scalability and efficiency of CNN-based classification systems, especially when dealing with large-scale or real-time applications where timely processing is crucial [202, 92].
    - **Most CNNs are supervised, and labelled HSIs are time-consuming and costly**: Most CNNs utilize supervised data and labelled HSI data is challenging to come by. Although unsupervised CNN-based approaches exist, most focus on data reconstruction rather than discriminability.

8. **Overfitting and Gradient Vanishing**: Addressing overfitting and gradient vanishing poses a significant challenge in the context of HSIC. With the deep architectures commonly used in deep learning models, there is a risk of overfitting, especially when dealing with limited training data [238, 239]. Additionally, the vanishing gradient problem can impede the training process, particularly in deep networks with many layers. These issues can hinder the ability of the model to generalize well to unseen data and affect the overall performance and reliability of the classification system [94, 240].

2.5 Summary

This chapter provides a thorough analysis of over 250 existing deep learning (DL) models tailored for hyperspectral image classification, by focusing on the aforementioned four main research objectives. It examines each DL model in detail, finding that the CNN stands out for its overall accuracy compared to other models such as LSTM, AE, GAN, and GCN. The study also highlights other effective models like attention networks, active learning, transformer networks, few-shot learning, and deep metric learning. The study highlights the

utilization of three major benchmark datasets—Indian Pines, Pavia University, and Salinas—in existing literature for performance evaluation. Notably, the evaluation indicates varied model performance across these datasets. Moreover, the existing models are also analyzed based on the their learning approach that is supervised, semi-supervised, and/or unsupervised DL. The analysis reveals a prevalent reliance on supervised learning in most models, underscoring the study's recommendation for exploring semi-supervised, unsupervised, or alternative learning strategies to effectively address the challenge of limited labeled samples. Overall, the findings shed light on the landscape of DL breakthroughs in HSI classification, highlighting key challenges and suggesting practical solutions. The study emphasizes the importance of addressing limitations posed by the scarcity of labelled non-traditional hyperspectral datasets and exploring various training modes to improve DL algorithm performance in HSI classification tasks.

CHAPTER 3

XCEP-DENSE: A NOVEL LIGHTWEIGHT EXTREME INCEPTION MODEL FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Convolutional Neural Networks have demonstrated their effectiveness in extracting discriminative features in the HSI domain. Despite their success, challenges persist due to the limited availability of labelled samples and the intrinsic high dimensionality of hyperspectral data. Overcoming these challenges is crucial for effective HSI data analysis and achieving high classification accuracy. While some existing models have achieved near-perfect classification accuracy, the depth of the model often leads to an excessive number of parameters, which leads to high training time. To tackle this, the Inception model, renowned for winning the ImageNet competition, has been exemplary. The Inception model employs inception modules with parallel convolutional filters, effectively capturing multi-scale features. Taking inspiration, a new model, namely, Xception, was introduced which leverages its success in handling diverse visual recognition tasks, aligning well with the spectral-spatial nature of hyperspectral data. The transition from Inception to Xception signifies a strategic evolution in the pursuit of efficient deep learning models. Xception, an extreme version of Inception, replaces inception modules with depthwise separable convolutions, enhancing parameter efficiency while maintaining classification accuracy. This transition aims to leverage extreme inception architectures' benefits, optimizing feature extraction for HSI's challenges of limited labelled samples and high dimensionality. To further improve the performance, this chapter introduces a new Xception network, namely Xcep-Dense, which is configured with dense network and optimization parameters to alleviate overfitting and gradient vanishing. The proposed network's performance is validated using two benchmark hyperspectral datasets, Indian Pines and Salinas.

3.1 Introduction

HSI stands at the intersection of spatial and spectral data, usually collected continuously from satellites or sensors across the electromagnetic spectrum ([241], [242]). This type of data finds applications across diverse fields, including resource management [243], agriculture [244], security and surveillance [14], astronomy, space surveillance [13], and coastal marine environment monitoring [245]. Despite its widespread use, a significant challenge in hyperspectral data applications is the accurate classification of ground features. For this, DL based models have been found promising because of their powerful feature extraction capability for nonlinear problems [7]. The deep networks such as Convolutional Neural Networks (CNNs) have shown promising results for image classification as these excel in filtering contextual spatial features of images through local connections [7].

The evolution of Convolutional Neural Networks (CNNs) began with Le-Net-style models, initiated by Yann LeCun in the early 1990s for handwritten digit recognition [246]. Recognizing the potential of CNNs, refinement of Le-Net paved the way for sophisticated architectures, with AlexNet emerging as a milestone in 2012, winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [247]. AlexNet's success spurred a trend towards constructing deeper networks, driven by the ILSVRC Challenge's goal to enhance image classification algorithms. However, the training of huge number of parameters and the limited availability of labelled samples pose a significant hurdle for HSI classification, leading to the problem of overfitting [248]; [249] and high training time. Additionally, CNN-based algorithms, while effective, demand substantial computational power and storage space ([42]; [250]), becoming computationally exhaustive with increased network operations. To address this, Google researchers introduced a new model, namely, Inception model, that a kind of revolutionized the field of classification by winning the ILSVRC Challenge [251, 252]. The main innovation of the Inception model lay in its inception modules and its parallel convolutional filters for efficient multi-scale feature extraction while minimizing model parameters and computational costs. The inception model marked a pivotal moment in CNN's evolution, emphasizing the importance of diverse architectural elements for tackling image classification challenges. Its ability to map cross-channel and spatial correlations with a single convolution kernel set a new standard. In summary, the journey from Le-Net to inception models signifies the continuous exploration and refinement of CNN architectures. Le-Net pioneered convolutional networks, AlexNet pushed performance boundaries, and the trend towards deeper networks culminated in the groundbreaking success of the inception model in 2014. Thus, given the challenge of maintaining classification accuracy with a decrease in computations, this chapter proposes a novel deep-learning-based lightweight classification model for HSI. The proposed model has the following contributions:

- The proposed model is based on an extreme version of inception, replacing standard two-dimensional convolutions with three-dimensional depthwise and pointwise-separable convolutions. With three-dimensional convolutions involved, the network can efficiently extract spectral-spatial information in hyperspectral cubes, causing a better capture of and a lesser number of parameters involved, making the network lightweight.

- The proposed methodology includes dimensionality reduction and 3D-SSCS (three-dimensional spectral-spatial cubing and slicing) technique that slices in the spatial and spectral dimensions simultaneously by efficiently reducing the number of calculations required and accessing the spectral-spatial information to improve classification accuracy.

- A dense network is customized and applied to the proposed Xception Network that further helps to discover discriminative features, encourages a strong gradient flow, enhances parameter efficiency and simplifies network training.

- In comparison to conventional convolutions, the proposed network based on depthwise convolutions is lightweight. The proposed optimized model is computationally efficient as it takes fewer model parameters resulting in faster convergence. The proposed network's performance is validated using two benchmarks HSI datasets, IP and SA.

3.2 Related Models

The design of CNN started with Le-Net [253] models that were simple piles of convolutions used for extracting features. This concept was refined into AlexNet [254] architecture with multiple convolutions repetitions between max pooling operations. The yearly ILSVRC Competition (ImageNet Large Scale Visual Recognition) and the arrival of VGG architecture [255] fuelled a trend towards making networks deeper. The Inception architecture was the first in the series to have the benefits of factoring convolutions, which won the ILSVRC 2014 challenge. Alotaibi and Alotaibi [47] combined Inception and ResNet's core ideas into a deep hybrid ResNet-Inception model for excellent classification performance. This architecture had a small number of layers but still achieved good results by converging to a minimum of 50 epochs. The first layer of Inception V1 and Inception V2 was later depthwise separable convolution [251]. Chollet [253] introduced the Xception network, which uses separable convolutions to reduce the size and computational cost of CNNs. Xiong, Yuan, and Wang [256] presented a model AI-NET attained excellent classification performance by proposing an attention inception module.

Other lightweight models provided excellent classification performance by maintaining a trade-off between accuracy and computation time that include SSRN [257], ResNet Inception [47], HDDA [258], FDSSC [259], DSSIR Net [260] and AUSSC [125]. Other models that took fewer parameters and yielded high classification accuracy included Cui et al. [83], FDSSC [259], p-ResNet [131], SSRN [136], SSUN [261], DBMA [262], DBDA [45] and SSUN [261]. The proposed model and its classification accuracy, shortcomings and limitations are summarized in Table 4.1. The accuracy of each model is provided concerning the dataset included in that specific study. Table 3.1 shows related models and achieved performance.

Table 3.1: Studies on Supervised Models

Ref	Proposed Model	Datasets	OA	AA	KC	Work Done with Advantages	Limitations
[263]	SAE-LR	**KSC**	98.76	97.90	98.62	SAE-LR using 4,00,000 epochs are used to generate discriminative features and generates high accuracy	Massive potential for accurate hyperspectral data classification. Long Training Time
		PU	98.52	97.82	98.07		
[136]	SSRN	**IP**	99.19	98.93	99.07	Learns spatial and spectral features separately, Mitigates the declining accuracy phenomenon, Needs 200 epochs only to train network, performs well with limited training samples	Overly long training time (6-10 Times more than CNN); Computationally expensive; Network training on limited data could be done to achieve the same results
		PU	99.79	99.66	99.72		
		KSC	99.61	99.33	99.56		
[259]	FDSSC	**IP**	99.75	99.67	99.72	Gives excellent accuracy in shorter training time that needs only 80 epochs, Narrower Model, Fast convergence	Epochs can be further reduced to design a lightweight model; networks could be supplied with lesser training data
		PU	99.97	99.97	99.96		
		KSC	99.96	99.96	99.95		
[125]	AUSSC	**IP**	94.55	94.44	93.77	Employs relatively small convolutional kernels to cut down on the parameters and alleviate overfitting in 400 epochs; Centre Loss function to improve accuracy	Uses more convolutional kernels, Centre loss functions increases the computational loss, Deep network resulting in longer training time, Needs more epochs
		KSC	98.26	97.48	98.0		
		SA	96.13	97.37	95.7		
[262]	DBMA	**IP**	98.19	96.31	97.94	Uses spatial and channel attention, fused to obtain more discriminative features. Works with limited labelled samples	Attention blocks help in faster convergence but end up increasing the parameters and, eventually, the computational cost
		PU	98.88	98.71	98.50		
		SA	98.04	98.85	97.82		
[45]	DBDA	PU	96.00	96.45	94.67	No pre-processing is required to reduce the dimensions, Less consumption time in comparison to models like FDSSC or DBMA	Training time can be reduced; Parameters can be reduced even more
		SA	97.51	98.00	97.23		
		IP	95.38	96.47	94.74		

Continued on next page

Table 3.1: Studies on Supervised Models (Continued)

Ref	Proposed Model	Datasets	OA	AA	KC	Work Done with Advantages	Limitations
[264]	AD-Hybrid SN	**IP** **PU** **SA**	97.02 98.32 99.59	92.34 97.23 99.24	96.6 97.8 99.5	Better explores spatial-spectral features; Works with limited training data; Accesses spectral and spatial data both	Unsatisfactory classification of specific ground objects; Network optimization could lead to better classification performance of proposed work
[220]	ResNet Inception	**IP** **PU** **PC** **SA**	90.57 95.31 99.02 95.53			A few layers still provide good results by converging to a minimum using 50 epochs.	Accuracy can be further improved; Specifically, in datasets like India Pines, classes with fewer samples are not considered in the overall data
[258]	HDDA	**IP** **PU** **SA**	96.80 98.28 98.85	95.83 97.07 99.25	96.34 97.72 98.72	Improved Feature extraction capability with two dense modules and 200 epochs	Limited Data available; Network can be tested with a lesser number of epochs; Parameters can be reduced
[260]	DSSIR Net	**IP** **SA** **PU**	97.18 99.35 99.31	96.82 99.60 99.20	96.78 99.27 99.05	Joint spatial-spectral extraction with an added attention module performed with 200 epochs	Moderate computational complexity; Parameters can be reduced while maintaining classification performance
[256]	AI-NET	**IP** **SA**	93.07 94.64		91.75 92.73	Attention inception and multi-resolution convolution filters to extract discriminative features	Classification accuracy could be increased; Network convergence could have been faster
[83]	Cui et. al	**IP** **PU** **SA**	96.51 96.48 97.18	93.30 94.25 98.44	96.02 95.34 96.87	Extracts features with a lightweight network, Uses depthwise convolutions to reduce the parameters and computational complexity	Classification accuracy could be increased with lesser training samples. Determining spatial window size is challenging
[261]	SSUN	**IP** **PU** **KSC**	98.40 99.46 97.71		98.14 99.26 97.45	Accustomed spectral and spatial feature extractors using LSTM and multiscale convolutions	High computational burden involved in training time. Learning the weight vector is a challenging part

3.3 Proposed Methodology

We introduce Xcep-Dense, a novel hybrid model network that blends the strengths of Xception and Dense networks to tackle the complexities of hyperspectral data analysis. As illustrated in Figure 3.1, the model's architecture unfolds in a multi-stage process. Firstly, PCA (Section 3.3.1 takes center stage, gracefully scaling down the hyperspectral cube to a more manageable dimension, ensuring computational efficiency without compromising valuable information. Secondly, the model divides the reduced data into smaller cubes (Section 3.3.2, ready for efficient processing within the network's intricate layers. These prepared cubes pass through the crafted hybrid network (Section 3.3.3, where Xception and Dense layers collaborate for feature extraction and learning. The forthcoming sections discuss the

preprocessing and classification of data.

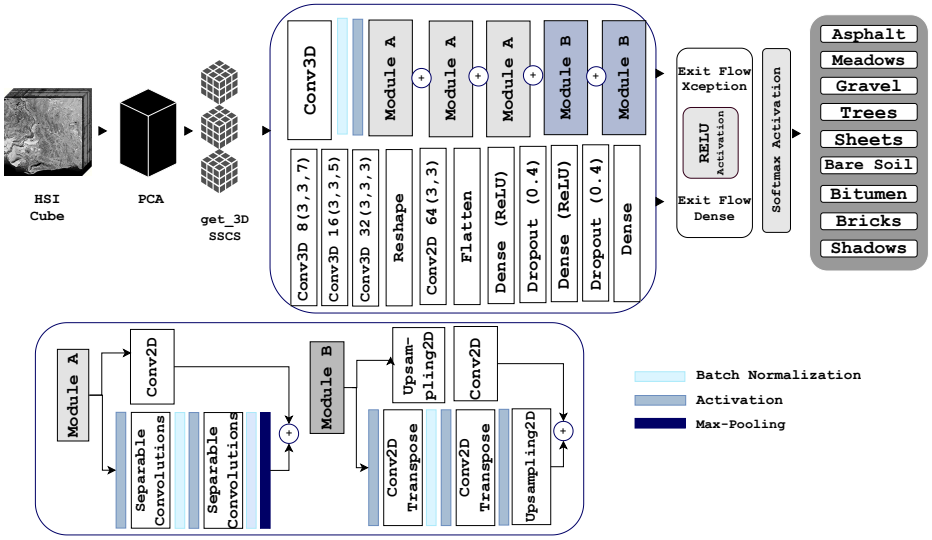
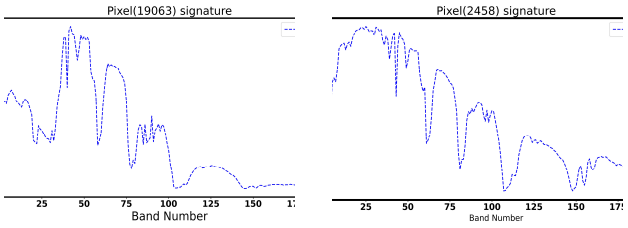


Figure 3.1: Block Diagram of Xcep-Dense



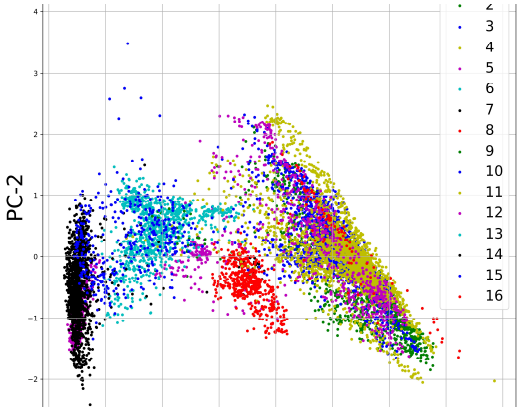
(a) Pixel 1578 (b) Pixel 66464

Figure 3.2: Spectral Signatures for IP Dataset

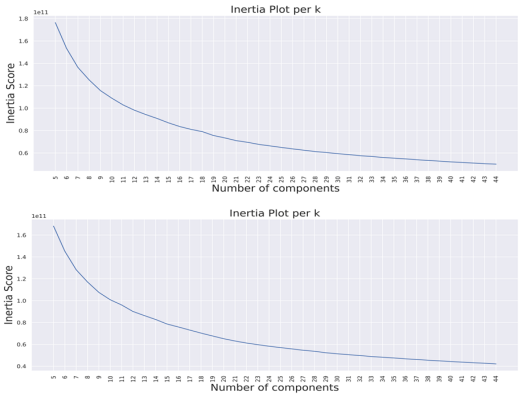
3.3.1 K-Means with PCA

Hyperspectral cubes contain a wealth of spatial and spectral information. Figure 3.2a shows the value of pixel 1578 in the IP dataset, and Figure 3.2b shows the value of pixel 66464 in the SA dataset. PCA was used to obtain the essential discriminative features and cut down on the dimensions. It generates orthogonal projections using eigenvalue decomposition, eventually minimizing the squared errors between the original and projected data. It is evident from the figure that neighbouring pixels have a slight variance, which requires a dimensionality reduction. Figure 3.3a shows different clusters in IP dataset for the two most variant components. PCA reduces the N available original components to M principal components having the highest eigenvalues [265].

Since hyperspectral data is highly skewed, manually selecting the number of features is ineffective. As a result, k-means clustering was attempted to measure the compactness of clusters in a data set while identifying discriminative features distinct from back-



(a) Clustered labels in the IP dataset



(b) (Top) K-Means before PCA (Bottom) K-Means after PCA

Figure 3.3: Spectral Signatures for IP Dataset

ground noise. This technique involves dividing the data into different clusters based on the compactness of the data points or inertia, which is referred to as the Euclidean distance. Every cluster has a defined centroid to which all other data points are measured about. Each data point is assigned to a cluster based on the shortest Euclidean distance. Figure 3.3b (top) shows the k-means clustering performed initially and Figure 3.3b (bottom) shows k-means clustering after applying PCA that lowered the inertia score depicted on the y-axis. Due to the highly skewed nature of hyperspectral data, the high inertia score $(176356304985.60797, 153231226798.99573, ..., 50065360614.01206)$ is plotted on the y-axis as an exponential value $1e10^{+11}$. The optimal value considering not to lose the discriminant bands was taken 25 for Indian Pines and 10 for Salinas. The initial set of data points can be denoted by $A = [a_0, a_1, a_2.....a_n]$ with o observations and f features where each data point belongs to a cluster with the nearest centroid. The final set of centroids can be shown by $B = [b_0, b_1, b_2.....b_n]$ with k clusters and f features, having minimum intra-cluster distance and maximum inter-cluster distance.

3.3.2 3-D SSCS

The 3-D Cube algorithm serves to break down multidimensional data into smaller, more manageable 3-D cubes. This serves as a pre-processing technique specifically tailored for HSI datasets, aims to segment the input data into smaller, uniform 3D blocks. This transformation, achieved through its algorithmic steps as depicted in Algorithm 1 and 3.4 visually, facilitates further analysis by reducing complexity and enabling fine-grained examination.

The algorithm operates in four steps. The first step is zero-padding that safeguards against boundary issues by padding the input data with zeros, ensuring context retention for each extracted cube. In the second step, The hyperspectral images under consideration include an IP image, measuring 145 X 145 pixels with 224 spectral bands, and an SA

image, sized at 512 X 217 pixels with an identical band count as the IP image. The process of obtaining 3D cubes involves taking the input images, 145 X 145 X 224 and 512 X 217 X 224, and splitting them into a set of smaller cubes with dimensions of 25 X 25 X 3. It extracts cubes of a predefined size, specified by the SplitSize parameter, at every possible location. In the third step, each extracted cube and its corresponding label, sourced from the input label array y, are stored in dedicated arrays sD and sL, creating a well-organized representation. The algorithm offers the flexibility to filter out cubes with specific label criteria using the removeZero parameter, resulting in a more concise and relevant dataset. By meticulously segmenting the data, the algorithm generates a structured output consisting of the split data array (sD) and the split label array (sL), ready for subsequent analysis.

Algorithm 1 Create3DCubes

Require: X: Input data, y: Labels, SplitSize: Size of cubes, removeZero: Boolean for removing zero-labeled cubes

Ensure: sD: Split data, sL: Split labels

$m \leftarrow (SplitSize - 1)/2$ {Margin}
$zeroPaddedX \leftarrow zeroPadding(X, m)$ {Zero-pad input data}
$sD \leftarrow \mathbf{0}^{X[0] \times X[1] \times SplitSize \times SplitSize \times X[2]}$ {Initialize SplitData}
$sL \leftarrow \mathbf{0}^{X[0] \times X[1]}$ {Initialize SplitLabels}
$sI \leftarrow 0$ {Initialize splitIndex}
for $r \leftarrow m$ **to** $zeroPaddedX.shape[0] - m - 1$ **do**
    for $c \leftarrow m$ **to** $zeroPaddedX.shape[1] - m - 1$ **do**
        $patch \leftarrow zeroPaddedX[r - m : r + m + 1, c - m : c + m + 1]$
        $sD[sI, :, :, :] \leftarrow patch$
        $sL[sI] \leftarrow y[r - m, c - m]$
        $sI \leftarrow sI + 1$
    end for
end for
if $removeZero$ **then**
    $sD \leftarrow sD[sL > 0, :, :, :]$
    $sL \leftarrow sL[sL > 0]$
    $sL \leftarrow sL - 1$
end if
return sD, sL

3.3.3 The Xception Network

The proposed model includes an extreme inception model based on depth-wise separable convolutions. ”It is based on the idea that mapping spatial correlations and spectral

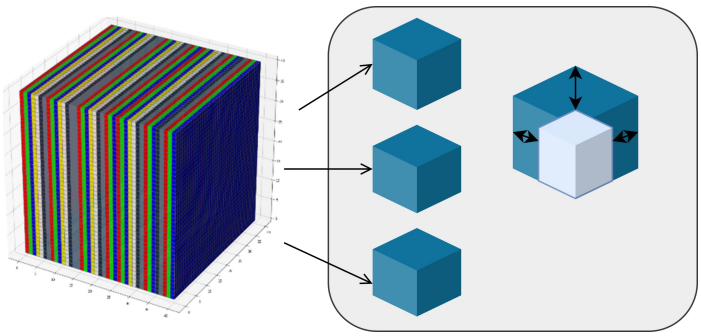


Figure 3.4: 3-D cubing sliced into smaller cubes and Padding with Zeroes

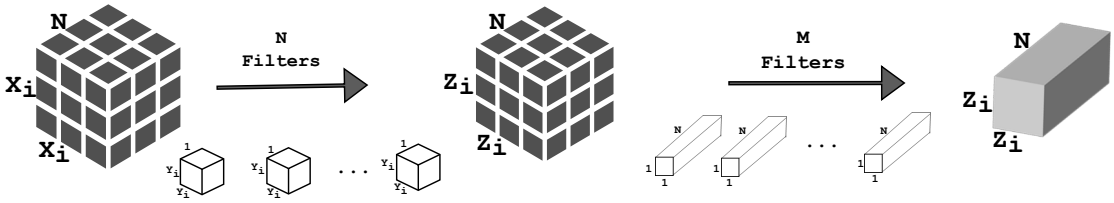


Figure 3.5: Xception Network with Separable or Depthwise Convolutions

associations can be completely decoupled" [252]. The goal of the extreme inception network modification is to remove the vast amounts of trainable parameters. Furthermore, the original Xception network uses two-dimensional convolutions, which are replaced by three-dimensional convolutions in the proposed network to fit the nature of hyperspectral data better. The Xception network is a queue of depthwise separable convolutions. Our proposed Xception network outperforms the original Xception network and achieves comparable performance with fewer parameters. Figure 3.5 depicts spatial convolutions executed independently on each input channel, preceded by pointwise convolution. Depthwise convolutions employ 1×1 convolution to map the relationships between spectral channels and spatial data. These convolutions can deal with spatial and depth dimensions with fewer computations, lowering computational overhead. Figure 3.6 depicts the flow of the Xception network, which is divided into three parts: entry, middle, and exit.

The proposed model includes 2D and 3D convolutional layers that convert input data into feature maps, including three Conv3D layers, four Conv2D layers, two separable Conv2D layers, and two Conv2D transposes. The feature maps are routed through the activation layers to create and extract features [266]. For classification, Xcep-Dense employs the most popular activation functions for hyperspectral imaging analysis, ReLU (Rectified Linear Unit), and Softmax [266]. The pooling layer provides invariance to slight data distortions and helps to prevent overfitting in networks by reducing the quantity of data and model parameters. Figure 3.6 shows the placement of the upsampling layer, which doubles input dimensions by drawing on multiple surrounding points. It shows the elaborated con-

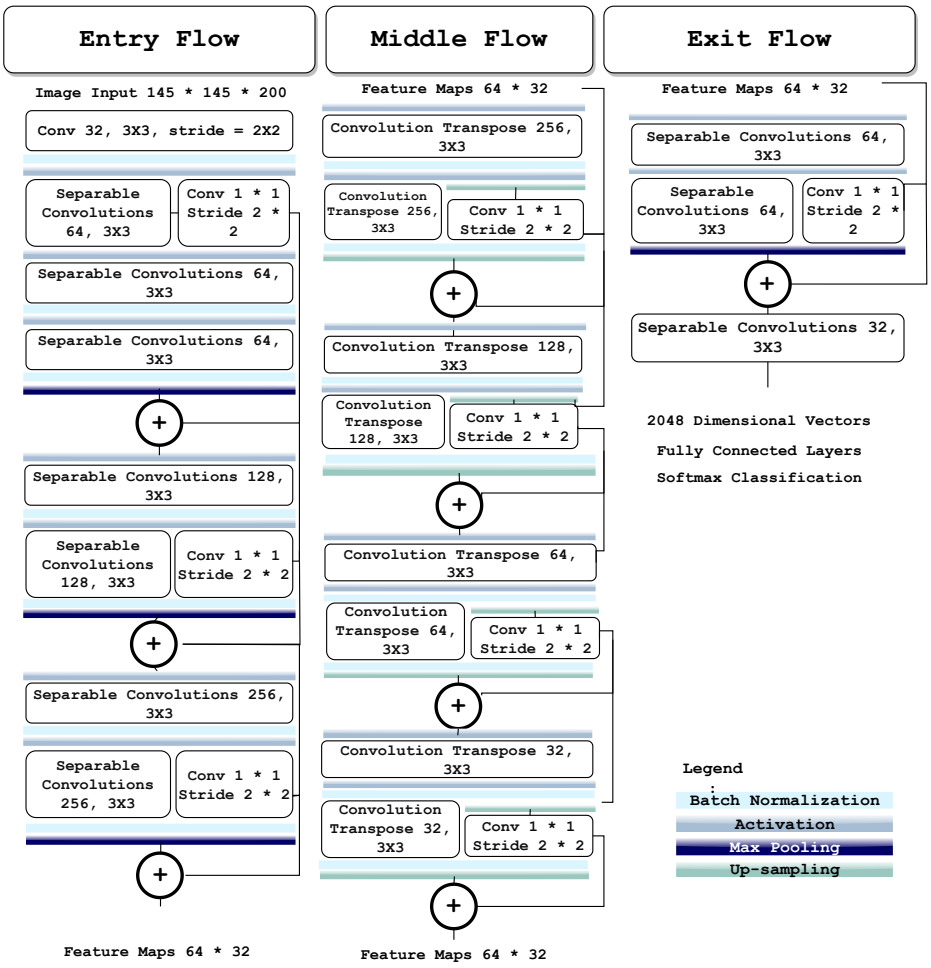


Figure 3.6: Overall Flow of Xcep-Dense

figuration of the proposed Xception3D model with layers, types, and parameters in each layer.

3.3.4 Dense Network

The final component includes three dense layers, two activated by ReLU and the third by Softmax. ReLU activation yields an element-wise maximum of 0, and the input tensor and Softmax is employed for the classification network, interpreting the result as a probability distribution. Figure 3.7 depicts four dense layers and their interconnections. This network is optimized by adding two dropout layers strongly coupled with the preceding. To perform matrix multiplication, neurons in the dense layer are entirely linked with neurons in the preceding layer, implying that row vectors in the preceding layer are equivalent to column vectors in the dense layer. The dense layer reduces vector dimensions from 256 to 128. In this case, A_i denotes the output of the i-th layer shown in Equation 3.1, and X_i denotes the nonlinear transformation operation shown in Equation 3.2.

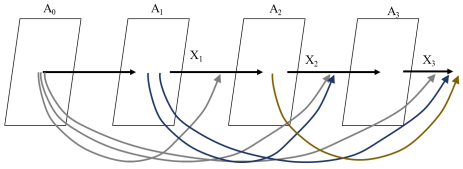


Figure 3.7: Dense Layers

$$A_i = X_i(A_{i-1}) \quad (2) \tag{3.1}$$

$$A_i = X_i(A_0, A_1, A_2, \ldots\ldots\ldots\ldots\ldots A_{i-1}) \quad (3) \tag{3.2}$$

To improve generalization error and alleviate overfitting, the proposed network uses two dropout layers with a parameter of 0.4, which offers an effective way of randomly dropping out nodes during the network training. Further, to alleviate overfitting and speed up network convergence, Batch normalisation standardises the inputs to a layer for each micro-batch, which stabilises the learning process and reduces the number of epochs required to train the network. Input layer A is defined by $A = [a_1, a_2, a_3, \ldots\ldots\ldots a_n]$ where a_n are the input samples, and n is the batch size. Equations 3.3, 3.4, and 3.5 represent mean, variance, and normalized input. In Equation 3.5, $\hat{a}$ is a single component's new value, E[x] is its mean, Var(x) is its variance, and ϵ is a constant.

$$\mu_B = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{3.3}$$

$$Var(x) = \sigma_B^2 = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu_B)^2 \tag{3.4}$$

$$\hat{a} = \frac{a_i - E(x)}{\sqrt{Var(x) + \epsilon}} \tag{3.5}$$

3.3.5 Optimizing Xcep-Dense

While Stochastic Gradient Descent (SGD) effectively minimizes the objective function and error rate, its computational demands can be overwhelming when dealing with extensive datasets. In a quest for further optimization, Kingma and Ba [267] introduced the Adam optimizer, a powerful enhancement that leverages historical gradient information to accelerate convergence and improve accuracy. Adam achieves this by ingeniously storing exponentially decaying averages of past gradients, incorporating both the first moment (mean) x_t and the second moment (variance) y_t of the gradients. Equations 3.6 and 3.7 offer a mathematical glimpse into this process, revealing the crucial roles of decay rates β_1 and β_2

in regulating gradient updates.

$$x_t = \beta_1 x_{t-1} + (1 - \beta_1) d_t \tag{3.6}$$

$$y_t = \beta_2 y_{t-1} + (1 - \beta_2) d_t^2 \tag{3.7}$$

To counter potential bias toward zero when β_1 and β_2 are small, Adam introduces bias-corrected first and second moments, as captured in Equations 3.8 and 3.9.

$$\hat{x}_t = \frac{x_t}{1 - \beta_1^t} \tag{3.8}$$

$$\hat{y}_t = \frac{y_t}{1 - \beta_2^t} \tag{3.9}$$

These refined moments, in turn, guide the weight update rule depicted in Equation 3.10, ensuring a more informed and adaptive optimization process.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{y}_t + \epsilon}} \hat{x}_t \tag{3.10}$$

Learning Rate

Learning rate acts like the network's cruise control, dictating how fast it learns and finds the best answer. We start fast with a high learning rate, then slow down as we get closer to the goal to avoid "wobbling" back and forth. But if we go too slow, it takes forever to get there. The idea behind the learning rate with exponential decay was to keep the learning rate high at first and then gradually decrease it as iterations progressed so that the gradient converges to an optimal value quickly to get a suboptimal solution. Keeping the learning rate parameter too high caused the gradient to swing back and forth from the global optimum. Keeping the training parameter too small slowed convergence and increased network training time, increasing computational complexity. The proposed model initially employed a fine-tuned weight decay (L2 regularisation) rate from 1e-5, which did not make the network perform well. This experiment was finally fine-tuned and set to 1e – 6 for the proposed model. Equation 3.11 represents the learning rate parameter where D is the decay rate, g_s is a global step and d_s is decay step, α_d is the exponential decay learning rate, and α represents the learning rate at the initial stage.

$$\propto_d = \alpha * D^{\frac{g_s}{d_s}} \quad (10) \tag{3.11}$$

To avoid another common problem called "overfitting," where the network focuses too much on specific details and forgets the bigger picture, we add a "penalty" term to the

learning process [123]. This gently nudges the network towards broader solutions that work for everyone, not just the specific examples it saw during training. The loss function is represented in Equation 3.12 where L_0 is the original loss function with n as the training sample size and w model weights given by $\frac{\lambda}{2n}\sum_w w^2$ as L2 norm penalty and λ is the hyperparameter regularising ratio.

$$L = L_0 + \frac{\lambda}{2n}\sum_{w} w^2 \quad (11) \tag{3.12}$$

By carefully setting the learning rate and adding this penalty, we help the network find the sweet spot - learning quickly and accurately without getting stuck on unimportant details. This lets it uncover the hidden patterns in the data like a skilled detective!

3.4 Results and Discussion

In this section, we start by discussing the experimental setup, Section 3.4.2 compares the classification accuracy of the proposed model using the statistics OA, AA, and the KC. In section 3.4.3 we compare the model's performance by comparing the parameters of the proposed model. This is followed by section 3.4.4, where we look at how increasing the number of training samples affects classification accuracy and performance evaluation of the xcep-dense network without involving the fully connected layers. Section 3.4.5 discusses the performance of the proposed network when dense layers were removed. Finally, section 3.4.6 includes the performance of Xcep-Dense on other benchmark datasets.

3.4.1 Experimental Setup

The experiments were performed in Python using Google Collaboratory. The number of extracted principal components was 25. Batch size was 256, the number of epochs was 20, the activation function was ReLU, and Softmax for dense network, the learning rate parameter was 0.001, and the exponential decay was 1e-06. Adam behaves like a resilient ball rolling down the loss function's convex curve. The training samples were 10% for IP and 5% for SA.

3.4.2 Experiment 1 : Performance of Xcep-Dense on IP and SA

To validate the effectiveness of our proposed model, we conducted a comprehensive comparison with state-of-the-art hyperspectral data classification algorithms found in the literature. Results for two benchmark datasets, IP and SA, are presented in Tables 3.2 and 3.3 respectively, showcasing outstanding overall classification accuracies of 98.52% and 99.69%. These accuracies outperform existing models, including the attention inception

network AI-NET [256], the hybrid deep ResNet and Inception network [47], a lightweight hyperspectral classification model based on MobileNetV3 [83], spectral-spatial residual network SSRN [136], end-to-end spatial and spectral network SSUN [261], and others. Notably, our proposed Xcep-Dense model achieves a classification accuracy of 98.52% for IP with only half the training samples used by the highest performing models, SSRN [257] and FDSSC [259], both achieving accuracies above 99%. Similarly, for the SA dataset, Xcep-Dense achieves the highest accuracy of 99.69% with only 5% of the samples used for training. These results underscore the improved performance of the Xcep-Dense model, even under limited training data conditions.

Figures 3.8(a) and 3.8(c) depict the training accuracy per epoch for IP and SA, highlighting the model's exceptional performance within just 20 epochs. The proposed 3D slicing algorithm, which slices hypercubes into smaller 3D cubes, reduces the number of parameters, making the network lightweight and facilitating faster convergence. Batch normalization (BN) further aids in gradient convergence, speeding up the training process and conserving time and resources. Compared to previous works requiring hundreds of epochs, our model achieves superior performance in just 20 epochs for SA and 100 epochs for IP. This is attributed to the concept of separable convolutions integrated into the network, resulting in optimal accuracy with reduced computational overhead. Figures 3.8(a-d) provide a detailed breakdown of the experimental results, demonstrating the effectiveness of the proposed model in mitigating overfitting.

Table 3.2: Classification results for IP dataset with 10% training data.

Classes	**SVM**	**AI-NET**	**Hybrid Deep Res Net**	**LSSAN**	**SSRN**	**SSUN**	**Chen et. al**	**AUSSC**	**FDSSC**	**3D-SACNN**	**AD-Hybrid SN**	**HDDA**	**DSSIR-Net**	**Proposed**
Ref		[256]	[47]	[83]	[257]	[261]	[268]	[125]	[259]	[269]	[264]	[258]	[260]	
	0.10	**0.30**	**0.10**	**0.05**	**0.20**	**0.17**	**0.20**	**0.10**	**0.20**	**0.15**	**0.05**	**0.05**	**0.10**	**0.10**
1	84.55	-	-	90.48	97.82	**100.0**	99.47	98.33	70.00	89.66	49.02	81.48	98.88	**100.0**
2	87.5	-	91.88	95.82	**99.17**	97.26	98.06	94.60	90.38	96.30	94.79	94.73	96.65	97.82
3	77.51	-	85.30	93.26	**99.53**	99.36	96.68	90.06	86.48	98.45	98.21	97.59	96.64	98.39
4	76.8	-	-	95.44	97.79	**99.85**	97.83	93.73	92.34	95.10	96.46	94.76	94.81	87.79
5	56.41	-	97.11	91.17	99.24	**99.56**	99.35	98.59	96.65	98.52	96.94	99.55	98.62	93.56
6	82.45	-	-	99.32	99.51	99.75	92.96	97.84	95.11	96.19	98.20	**100.00**	99.13	99.84
7	92.23	-	-	93.08	98.70	99.58	99.65	86.57	40.00	99.07	**100.00**	**100.00**	94.18	96.0
8	80.0	-	**100**	99.49	99.85	**100**	97.19	97.22	92.51	**100.0**	99.90	**100.00**	99.92	**100.0**
9	96.67	-	-	57.78	98.50	**100**	96.74	91.61	**100.00**	**100.0**	65.28	**100.00**	83.71	88.88
10	50.0	-	84.98	96.07	98.74	98.15	98.14	92.40	84.87	96.49	95.57	94.53	97.07	**99.65**
11	84.21	-	86.27	97.67	99.30	97.15	96.97	93.97	95.32	98.15	99.03	97.60	97.65	**99.95**
12	86.68	-	89.56	94.88	98.43	**99.52**	96.24	94.52	92.45	94.15	90.57	95.17	98.69	98.68
13	86.51	-	-	99.24	**100.0**	99.81	**100.0**	97.67	99.70	96.58	98.32	98.99	**100.0**	98.91
14	97.36	-	99.68	98.82	99.31	99.39	99.39	96.65	96.39	**99.71**	98.85	99.17	98.78	98.33
15	37.12	-	-	95.52	99.20	99.05	**100.0**	94.63	90.36	97.51	98.24	93.31	95.76	99.42
16	13.63	-	-	95.06	97.82	**100.0**	99.62	92.70	98.01	94.54	98.04	86.36	98.70	95.23
OA	80.34	93.07	90.57	96.51	**99.19**	98.40	98.02	94.55	**99.75**	97.47	97.02	96.80	97.18	**98.52**
AA	74.35	-	91.84	93.3	98.93	99.22	98.01	94.44	**99.67**	96.90	92.34	95.83	96.82	97.03
KC	72.04	91.75	-	96.02	99.07	98.14	-	93.77	**99.72**	97.12	96.60	96.34	96.78	98.31

Table 3.3: Classification results for SA dataset with 5% training data.

Classes	**AI-NET**	**Hybrid Deep Res Net**	**LSSAN**	**AUSSC**	**3D-SACNN**	**2D-SACNN**	**AD-Hybrid SN**	**R-Hybrid SN**	**HDDA**	**DSSIR Net**	**Proposed**
Ref	[256]	[47]	[83]	[125]	[269]	[269]	[264]	[270]	[258]	[258]	[260]
	0.05	**0.05**	**0.01**	**0.02**	**0.15**	**0.15**	**0.01**	**0.01**	**0.01**	**0.05**	**0.05**
1	-	99.60	**100.00**	**100.00**	99.88	**100.00**	99.81	**100.00**	**100.00**	**100.00**	99.84
2	-	99.95	99.66	99.82	99.90	99.79	**100.00**	99.97	**100.00**	**100.00**	**100.0**
3	-	99.90	**100.00**	95.04	99.23	99.84	99.98	99.49	**100.00**	99.86	**100.0**
4	-	99.43	99.43	97.99	**99.89**	98.98	99.17	98.72	98.04	99.66	99.62
5	-	99.55	98.52	98.69	99.76	99.52	99.50	98.43	98.83	**99.91**	99.44
6	-	99.79	99.96	99.98	**100.00**	**100.00**	**100.00**	99.90	99.80	**100.00**	**100.0**
7	-	99.89	99.83	99.56	99.57	99.22	99.97	99.96	**100.00**	**100.00**	**100.0**
8	-	92.41	93.45	92.83	98.30	96.67	**99.70**	98.23	97.80	99.07	99.36
9	-	**100.00**	99.99	99.41	99.89	99.96	**100.00**	99.99	99.64	**100.00**	**100.0**
10	-	97.31	95.16	98.25	99.56	99.71	98.96	97.90	99.44	**99.79**	99.48
11	-	98.50	99.16	93.90	98.26	**99.77**	99.22	96.46	99.72	97.78	99.21
12	-	99.79	99.92	98.40	99.76	**100.00**	99.92	99.09	99.84	**100.00**	**100.0**
13	-	99.13	99.45	98.59	99.74	**100.00**	95.59	82.82	**100.00**	99.80	**100.0**
14	-	99.06	99.33	96.13	98.25	99.46	97.49	97.25	99.43	98.93	**100.0**
15	-	79.53	93.01	89.32	98.16	95.24	**99.57**	95.12	96.88	98.83	99.42
16	-	99.22	97.90	99.94	**100.00**	99.72	99.00	99.71	98.51	**100.00**	99.94
OA	94.64	95.33	97.18	96.13	99.19	98.50	99.59	98.25	98.85	99.35	**99.69**
AA	-	-	98.44	97.37	99.38	99.24	99.24	97.69	99.25	99.60	**99.77**
KC	92.73	-	96.87	95.70	99.10	98.33	99.50	98.0	98.72	99.27	**99.66**

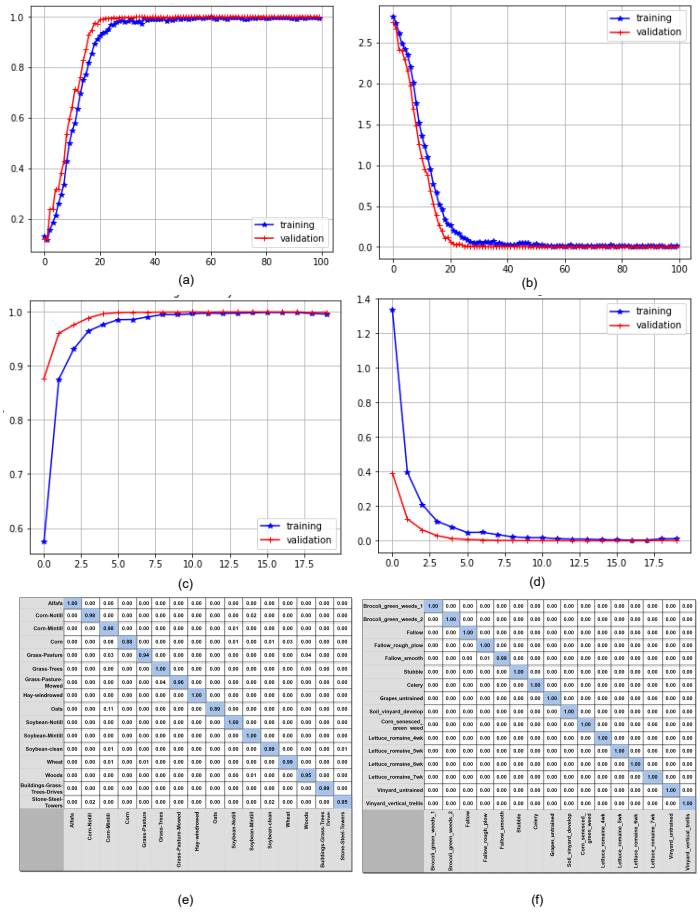


Figure 3.8: (a) Training Accuracy for IP Dataset and (b) Training Loss for IP Dataset; (c) Training Accuracy for SA Dataset and (d) Training loss for SA Dataset; (e) Confusion Matrix for IP (f) Confusion Matrix for SA

3.4.3 Experiment 2 : Comparison of Parameters

The Xcep-Dense network is designed with a significantly lower parameter count compared to models utilizing depthwise convolution. This reduction is achieved through the incorporation of two-dimensional, three-dimensional, transpose, spatial separable, and depthwise separable convolutions, effectively minimizing calculations and parameters. Depthwise separable convolutions combine depthwise and pointwise convolutions, processing each channel individually and employing pointwise convolutions with a sliding window of 1×1. As suggested by Chollet [252], Assuming A_i is the size of the input convolution kernel, A_o is the size of the output feature map, and Y and X are the numbers of spectral channels in the input and output feature map, respectively. The number of parameters in conventional convolutions can be written as $A_i \times A_i \times X \times Y$, and the number of calculations can be written as $A_i \times A_i \times X \times Y \times A_0 \times A_0$. However, the number of parameters in the case of depthwise convolutions is $A_i \times A_i \times X$ and the number of calculations is $A_i \times A_i \times X \times A_0 \times A_0$. The number of parameters in the case of pointwise convolutions is $X \times Y$, and the calculation amount can be written as $X \times Y \times A_0 \times A_0$. Hence the number of parameters in the case

Table 3.4: Comparison of Parameters

Model	Reference	Model Parameters (in millions)
Cui et al	[83]	0.2 M
SSRN	[257]	0.36 M
FDSSC	[259]	1.23 M
pResNet	[131]	1.13 M
SSUN	[261]	1.22 M
DBMA	[271]	0.61 M
DBDA	[45]	0.38 M
SSUN	[261]	1.22 M
Xcep-Dense	Proposed Work	**0.1 M**

of depthwise versus conventional convolutions can be expressed by Equation 15, and the calculation ratio of depthwise versus conventional convolutions is expressed by Equation 16

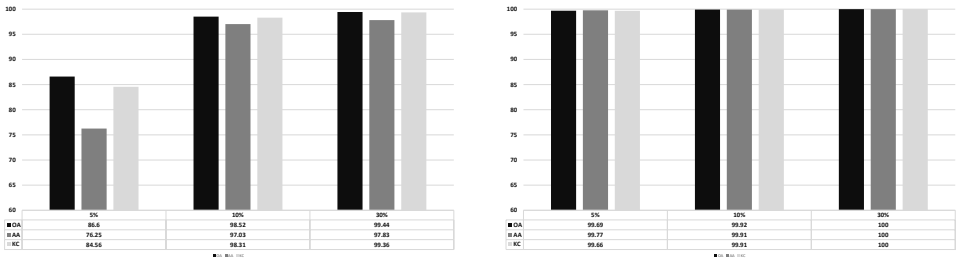
$$P_{\frac{DC}{CC}} = \frac{(A_i \times A_i \times X) + (X \times Y)}{A_i \times A_i \times X \times Y} = \frac{1}{Y} + \frac{1}{A_i \times A_i} \tag{15}$$

$$CA_{\frac{DC}{CC}} = \frac{(A_i \times A_i \times X \times A_0 \times A_0) + (X \times Y \times A_0 \times A_0)}{A_i \times A_i \times X \times Y \times A_0 \times A_0} = \frac{1}{Y} + \frac{1}{A_i \times A_i} \tag{16}$$

The depthwise convolutions in the proposed xception network quilted with dense layers drastically reduce the model parameters and the number of calculations going in the background, making the model lightweight. Table 3.4 shows that proposed Xcep-Dense involving depthwise convolutions outperforms existing lightweight models including inception or residual blocks. The proposed model uses lesser model parameters (109840 = 0.1 million) in comparison to existing deep convolutional networks Cui et al [83], SSRN [257], FDSSC [259], pResNet [131], SSUN [261], DBMA [271], DBDA [45], SSUN [261] making the proposed network lightweight. The replacement of 3 × 3 or 5 × 5 convolutions by 1 × 1 convolutions reduces the overall number of calculations making the training process faster.

3.4.4 Experiment 3 : Effect of Training Data

The proposed model's performance was assessed with varying training data percentages (5%, 10%, and 30%). Figures 3.9a and 3.9b illustrate the results for IP and SA datasets, respectively. Optimal results were observed with 10% of the IP dataset for training. Beyond 10% samples, a marginal improvement in overall accuracy and computing time was noted. Similarly, the peak performance for the SA dataset occurred with 5% training samples. Consequently, we selected 10% and 5% as the optimal training samples for IP and SA datasets, respectively.



(a) Data: IP (b) Data: SA

Figure 3.9: Effect of Training Samples

3.4.5 Experiment 4 : Effect of Dense Layers

The performance of the proposed network witnessed a significant boost with the incorporation of dense layers. When the model was applied to the IP dataset with only 5% training samples, the OA reached 89.91%, AA was 82.20%, and KC achieved 88.45% for the Xception network. The introduction of dense layers played a crucial role in reducing the parameters of the high-dimensional hyperspectral imagery to 2048, resulting in a remarkable improvement of 5%, 8%, and 5% in OA, AA, and KC, respectively. The utilization of Rectified Linear Unit (ReLU) activation facilitated faster convergence for the Xception network, mitigating the risk of vanishing gradients. The incorporation of dropout regularization, coupled with dense layers, further strengthened dependencies between adjacent layers, contributing to an overall increase in accuracy. Figure 3.10 displays the training loss and accuracy for the Xception network without the involvement of dense layers, illustrating the clear impact of dense layers in mitigating overfitting and enhancing network performance.

3.4.6 Experiment 5: Performance of Xcep-Dense on other benchmark datasets

While the table showcases promising performance for Xcep-Dense on three datasets PU, KSC and SA-A, a closer look reveals some nuances. OA is impressive, reaching 99.97% on SA-A. However, this exceptional result might be dataset-specific. KSC lands at a significantly lower OA (86.39%) despite having a larger training data size (0.10) compared to PU and SA-A (both 0.05). This inconsistency highlights the potential for the model to be sensitive to the characteristics of the data it's analyzing. Furthermore, AA paints a similar picture. While SA-A boasts a stellar 99.95% AA, PU and KSC fall behind at 98.43% and 81.18% respectively. This variation suggests that Xcep-Dense might struggle with certain classes within the datasets, even if overall accuracy appears high. Overall, the table indicates that Xcep-Dense has the potential for excellent performance. However, drawing

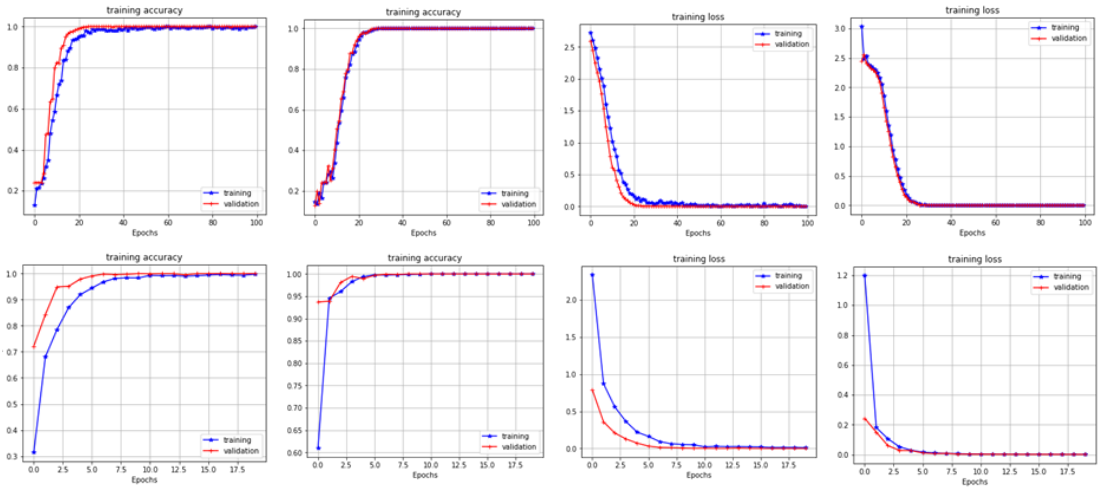


Figure 3.10: Training Accuracy and Training Loss of Xception network for IP and SA with and without Dense layers involved Top layer showing **IP Dataset (Left to Right)**: Training Accuracy With Dense, Training Accuracy No Dense, Training Loss with Dense, Training Loss No Dense; Bottom layer showing **SA Dataset (Left to Right)**: Training Accuracy With Dense, Training Accuracy No Dense, Training Loss with Dense, Training Loss No Dense

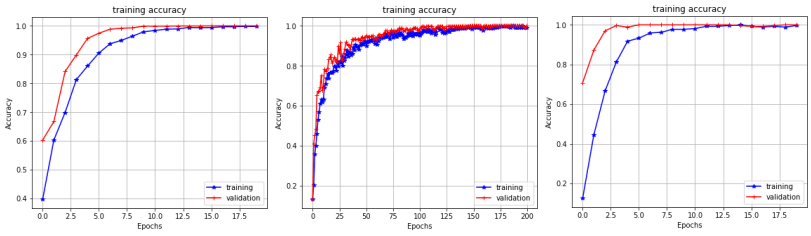


Figure 3.11: Training Accuracy Xcep-Dense for other three benchmark datasets (Left to Right) : Pavia U, KSC, Salinas

definitive conclusions requires further context. Understanding the specific characteristics of each dataset and the purpose of the classification task would allow for a more critical evaluation of the model's strengths and weaknesses.

3.5 Summary

The proposed lightweight Xcep-Dense network addresses the challenge of extensive parameters in existing deep learning models, offering comparable classification accuracy with significantly fewer parameters. Leveraging depth-wise and pointwise separable convolu-

Dataset	Training Data	OA	AA	KC
Pavia University (PU)	0.05	99.27	98.43	99.04
Kennedy Space Centre (KSC)	0.10	86.39	81.18	84.77
Salinas-A	0.05	99.96	99.97	99.95

Table 3.5: Performance of Xcep-Dense on Pavia U, KSC, and Salinas-A

tions, the network achieves efficiency in parameter reduction. The integration of dense modules enhances network depth and enables a more robust extraction of spatial-spectral properties. Key optimization techniques, including batch normalization (BN) and dropout layers, contribute to faster convergence. Additionally, the ReLU activation function plays a crucial role in preventing overfitting and reducing the number of training parameters. This method stands out for its superior classification accuracy, outperforming state-of-the-art models even with limited training data. The proposed network's efficiency lies in achieving excellent classification performance with fewer parameters, leading to faster convergence and mitigated overfitting. While this model achieves great classification accuracy, it doesn't work for cross-domain data. This model is tuned to work for data with similar properties and bands. For future work, expanding the network's capabilities to handle cross-domain data and enhancing its adaptability to different domains would be beneficial. Despite this limitation, the proposed method stands out for its superior classification accuracy, outperforming state-of-the-art models even with limited training data.

CHAPTER 4

DEEP SIAMESE NETWORK WITH HANDCRAFTED FEATURE EXTRACTION FOR HYPERSPECTRAL IMAGE CLASSIFICATION

The prominence of deep learning models for the classification of hyperspectral images is directly proportional to their ability to exploit spatial context and spectral bands jointly. However, the success of these models is reliant upon having a substantial number of labelled training samples. This poses a significant challenge in the context of hyperspectral images, where obtaining annotated samples is a time-consuming and labour-intensive process, resulting in the limited availability of labelled data. While traditional machine learning algorithms exist, they come with prolonged training times, and employing very deep pre-trained networks such as GoogleNet and VGGNet proves ineffective for hyperspectral image classification. The idea of one-shot classification has been quite motivating in recent years to deal with the problems of limited labelled samples, and imbalanced distribution of samples leading to poor classification results and overfitting issues. To overcome these challenges and leverage on few-shot classification, this chapter proposes a new deep Siamese network for HSIC method that can work with limited samples or imbalanced samples. The proposed Siamese network has a handcrafted feature generation network that extracts discriminative features from the image.

4.1 Introduction

Hyperspectral images, rich in spatial and spectral information [235], [238], find application in diverse fields such as defence, military, medical diagnosis, geological mapping, object recognition, environmental monitoring, anomaly detection, and forensic analysis [272] [273] [274] [275]. A pivotal challenge in hyperspectral imaging is accurate ground object

classification [276] and various learning approaches, including supervised, unsupervised, and semi-supervised learning, can be employed for the classification process. However, achieving robust performance in supervised classification encounters significant obstacles, primarily stemming from the limited availability of labelled samples [277] [278]. Additionally, the capability of deep learning models to effectively handle imbalanced datasets in hyperspectral image (HSI) classification poses another challenge, given the uneven distribution of samples across categories [279]. The annotation of samples adds to the complexity, being a time-consuming process, thereby contributing to suboptimal classification performance [175], [177], given the heavy reliance of deep learning models on sufficient training data.

Supervised deep learning methods have demonstrated superior performance compared to unsupervised approaches, which categorize data based on similarities and patterns [138]. Many supervised methods, such as Linear Discriminant Analysis (LDA), aim to maximize separability between classes but may not suit the classification of nonlinear HSI data [280]. Feature extraction methods like PCA [281], Independent Component Analysis (ICA) [282], Local Linear Embedding (LLE) [280], and spatial-spectral modules like Extended Morphological Profile (EMP) [23] or Morphological Attribute Profile (MAP) [283] have been employed in the past. Traditional machine learning algorithms, including support vector machines, are effective for small training samples but require longer training times for large datasets. In recent years, deep learning, encompassing models like Convolution Neural Networks [49], [284], Generative Adversarial Networks [285], Autoencoders [251], [255], Transfer Learning [213], [286], [10], and Recurrent Neural Networks [16], has gained prominence for achieving state-of-the-art results in HSI classification. However, these models face challenges, such as the need for substantial training data and the risk of overfitting [49], [284], [285], [213], [286]. Despite the effectiveness of deep learning models, their performance heavily relies on the availability of a significant amount of training data, making them susceptible to challenges posed by limited labelled samples and imbalanced datasets in HSI classification [279]. The inadequacy of training data for deep learning models can result in the vanishing gradient problem and may lead to overfitting, especially when dealing with large datasets [49], [284], [285], [213], [286]. To address these challenges, the concept of one-shot learning has been explored, leveraging the Siamese network to train models with fewer data points and mitigate issues related to overfitting and limited training data [10], [287]. This chapter proposes a Siamese CNN with a handcrafted feature extraction module to effectively handle problems like imbalanced class distribution, limited labelled samples, and overfitting in HSI classification.

In summary, this chapter makes a significant contribution by addressing challenges in hyperspectral image classification, offering a novel approach to one-shot learning through Siamese CNN with a specialized feature extraction module.

- The proposed work doesn't need too many instances of any class to train the network, rather uses adequate training data to build a strong network using Siamese as the base model to determine the similarity of input pairs.
- Both datasets are pre-processed after the removal of the noisy bands to further improve the classification accuracy. A novel feature extraction module is handcrafted, fine-tuned and merged with the Siamese network helps in increasing the classification performance by extracting finer discriminative features from the input image.
- The proposed network alleviates the problem of overfitting and mitigates gradient vanishing by utilizing the Adam optimizer, binary cross-entropy function, and dropout layers. To prove the worth of the model, the proposed method is applied to two benchmark hyperspectral data sets, IP and PU and results prove that the designed model can outperform other models even with small-scale training data.

4.2 Related Models

Because of the prominence of deep learning models over conventional machine learning models for image classification tasks, CNNs were first used by [288] and recently they seem to outperform all other models specifically in the area of image classification [254], [289]. The IP and PU dataset was classified using CNN providing a greater accuracy with small training samples in comparison to traditional models like SVM. To make the performance even better, this study suggests the use of a Siamese network as it can deal with data having a small number of samples per category. This study also suggests the use of a dropout layer to optimize the network. [290] worked on the problem of dimensionality of HSI by proposing spatial-spectral regularized sparse hypergraph embedding (SSRSHE) and possesses better classification performance by reducing the influence of noisy data. However the proposed model uses information in both spectral and spatial domains, therefore more time-consuming than the existing spectral models. [50] proposed a framework based on Siamese network and autoencoders making full use of the limited labelled sample information to provide a classifier for HSI. This model was proven great in generating discriminative features even on the edges of cubes however accuracy of the model declines with training samples as low as one percent. In 2020, [49] came up with the idea of dual path Siamese network that provides better results in case of limited samples than models like SVM or CNN but still has a scope of improving accuracy. Recently, [291] proposed a method based on few shot classification and attention mechanisms that solve the problem of annotating samples being a time-consuming and costly task. It explores the relationships between labelled and unlabeled samples using the Siamese network to further analyze the intra-class and inter-class distances. However, the model did not produce great classifica-

tion results. [279] and [31] use Siamese networks with the augmented data and calculate a similarity score. The former study works well with small and imbalanced samples with a limitation that when the samples in each category reach a satisfactory mark, the model may not work well. Later study [31] makes use of dynamic augmentation that changes the value of loss as per the augmentation with a risk of misclassification. [228] developed a model for limited samples, SaSiResNet based on superpixel segmentation using the Siamese network that even works for unlabeled samples with a disadvantage to deriving an optimal number of superpixels and making the model computationally efficient.

Table 4.1: Studies on Supervised Models

Ref	Proposed Model	Datasets	OA	AA	KC	Work Done with Advantages	Limitations
[292]	Pathak et. al	**IP** **PU** **SA**	91.09 96.74 94.81	83.53 96.18 97.13	89.78 95.64 94.21	Addition of newer classes in the existing dataset does not lead to the retraining of network	Algorithm needs optimization and fine-tuning
[293]	Hu et. al	**IP** **PU** **SA**	90.16 92.56 92.60			Great accuracy with small training samples, Better than SVM.	Can better perform with the use of Siamese network optimized with dropout layer
[51]	CAP1	**PU** **SA**	94.90 99.94	95.42 99.95	93.24 99.92	Combined CNN and Capsule networks	Need to enhance CapsNets, that takes longer time than CNN
[290]	SSRHE	**IP** **PU**	89.78 92.59	90.88 90.55	88.4 90.2	Worked on the problem of dimensionality by proposing SSRSHE and possesses better classification performance	Time taking than the existing spectral models
[50]	w-S-SAN	**IP** **PU**	93.57 97.61	92.06 96.49	92.77 96.83	Proposed a framework based on Siamese network + autoencoders making full use of the limited labelled sample information.	Accuracy of the model declines with training samples as low as one percent.
[228]	SaSiResNet-2	**IP** **PU**	76.33 82.32	80.98 85.30	73.33 77.31	Developed a model for limited samples, SaSiResNet based on superpixel segmentation using Siamese network that even works for unlabeled samples	Disadvantage to derive optimal number of superpixels and make the model computationally efficient.
[291]	AwGCN	**IP** PU	70.47 62.94	72.31 63.82	42.70 59.39	Proposed a method based on few show classification and attention mechanisms, Solves the problem of annotating samples, Explores the relationships between labelled and unlabeled samples	The model did not produce great classification results.
[294]	Zhu et. al	**IP** **PU** **HU**	99.63 99.97 95.36	99.79 99.95 96.00	99.58 99.96 94.99	Proposed SSDGL with attention mechanism	
[295]	TLFN	**PU** **SA**	95.63 97.00	94.03 98.61	94.21 96.64	Increased classification performance	Limitation of long computational time and validation required on other hyperspectral datasets

Continued on next page

Table 4.1: Studies on Supervised Models (Continued)

Ref	Proposed Model	Datasets	OA	AA	KC	Work Done with Advantages	Limitations
[49]	Dual-SCNN	**PU**	85.81	85.44	81.36	Proposed Dual Path Siamese network that provides better results in case of limited samples than models like CapsNet, SVM or CNN	Still has a scope of improving accuracy.
		KSC	90.80	89.63	89.75		
[279]	SSDDA	**PU**	93.22	88.85	91.02	Use Siamese networks with the augmented data and calculate a similarity score. Works well with small and imbalanced samples.	When the samples in each category reach a satisfactory mark, the model may not work well.
		IP	95.16	92.13	83.33		
[296]	AP3D	**PU**	91.37	90.50	88.45	Proposed Attention Aware pseudo-3D CNN (AP3D) that pays attention to important features in the image at all levels of convolutions	Complex model, Reduction in the model size required, Also, classification accuracy needs to be improved
		IP	89.14	94.36	87.61		
		SA	93.56	96.77	92.83		
[297]	SVM + SCNN	**PC**	99.68	99.26	99.55	Proposed a fine-tuned, SVM-based algorithm that outperformed other machine learning-based algorithms	Inclusion of a feature extraction module could have greatly improved the effectiveness of the proposed method
		PU	99.08	99.08	98.79		
		IP	99.04	99.14	99.87		
[31]	SA-SCNN	**KSC**	94.64	94.24	94.03	Makes use of dynamic augmentation that changes the value of loss as per the augmentation	Risk of misclassification.
		PU	88.11	85.58	84.49		
[298]	CAG	**IP**	77.01	78.27	91.14	Performs HSI classification using attention mechanism and graph networks to generate a relationship between features	Uses NVIDIA GTX1070 GPU for performing the experiments, therefore time consumption is lower
		PU	90.30	88.81	87.01		
		SA	91.14	94.14	90.15		
[150]	DGCN-GC	**SA**	87.33	88.23	86.49	Works on the problem of limited samples using a hybrid network combined point graph and distribution graph (DGCN), fully extracts features to study correlations	Still a scope of improving accuracy
		IP	77.46	77.79	77.63		
		PU	79.87	79.84	77.09		

4.3 Proposed Methodology

In this section, we delve into the details of the proposed model, which starts with the description of proposed siamese network in subsection 4.3.1, pre-processing of data in subsection 4.3.2 and handcrafted feature extraction network in subsection 4.3.3.

4.3.1 Siamese Network Architecture

The Siamese Network, renowned for its capacity to assess similarity, proves especially valuable in image classification scenarios involving numerous categories or datasets with imbalanced class distributions [58, 299]. In such cases, distance-based methods, leveraging the same neural network to compute a similarity metric between the pattern to be classified and a database of stored patterns, are commonly employed [300]. The Siamese network is selected for its capability to mitigate the importance of labels by categorizing them as either similar or dissimilar, rendering it particularly suitable for HSI due to the scarcity of labelled

or supervised samples [50].

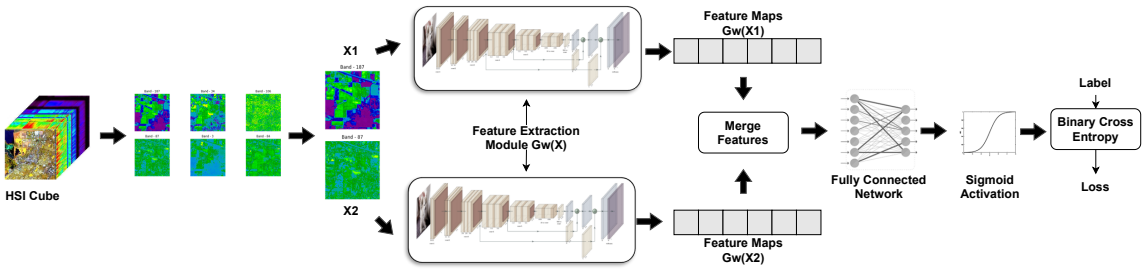


Figure 4.1: Working of the Proposed Model

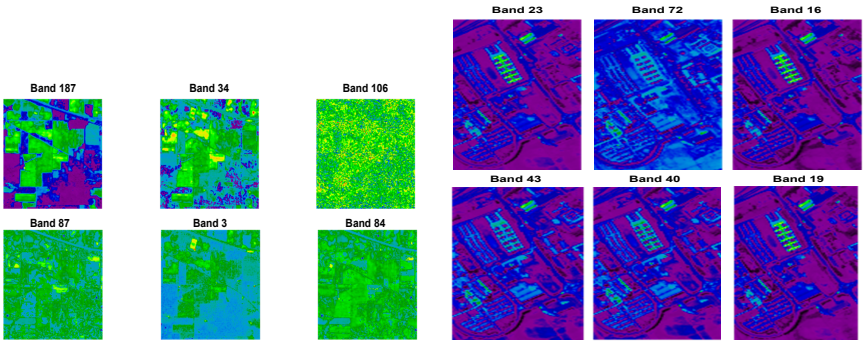
The proposed model, illustrated in Figure 4.1, integrates a feature generation module applied to each image. This module produces reduced feature vectors with discriminative features and subsequently computes the similarity score. The core of the Siamese network comprises CNN layers, including convolution layers, pooling layers, dense layers, and dropout layers, each serving a specific purpose. Convolutional layers capture low, mid, and high-level features, treating the input as an HSI cube containing multiple bands. The feature extraction module processes bands to generate feature maps, with each map utilizing a distinct convolutional kernel for diverse feature analysis. The pooling layer further reduces feature dimensions by calculating mean or variance within subtle rotations. The proposed model uses a max pooling layer, determining the maximum value among neighboring pixels in a feature map. Finally, vectors pass through fully connected layers to summarize information, followed by sigmoid activation.

This network operates by creating usable batches for training, taking two images (A and B) as input and outputting a similarity score. The fundamental assumption is that if the input images belong to the same group, the similarity score is 1, indicating complete resemblance, and 0 otherwise. The random batches algorithm randomly selects two sample input images, with the first half from the same category and the second half from a different category. This approach ensures a mix of similar and dissimilar image pairs during training, enhancing the network's ability to learn robust features.

In 2016, [301] proposed a pixel pair labelling scheme that maps a pixel pair set to a multiclass label set. Let A be an input HSI with n labelled samples, represented as $((x_i, y_i)_{i=1}^{n})$ with labels $y_i \forall 1, 2...C$, where C is the number of classes. The random pixel pair vector $[x_i, x_j]$ is selected from the training set, and the labeling strategy is defined by Equation 4.1. This equation compares two input pixels, returning the category "1" if they belong to the same label l; otherwise, if the samples belong to different classes, the label remains 0.

$$(x_i, x_j) = \begin{cases} 1 & \text{if } x_i = x_j = l \\ 0 & \text{if } x_i \neq x_j \end{cases} \tag{4.1}$$

[23] defines equation 4.2 to test any pixel vector t, network computes the similarity



(a) Dataset : IP (b) Dataset : PU

Figure 4.2: Band Images

score s_i of pixel pair $(t,\ s_i)$ for the available pixels x_i and computes the label $\hat{l}$ by taking the maximum s_i. Here $\hat{l}$ is the predicted label, and $\arg\max_i$ is the argument at which the maximum value is achieved applied at similarity scores s_i.

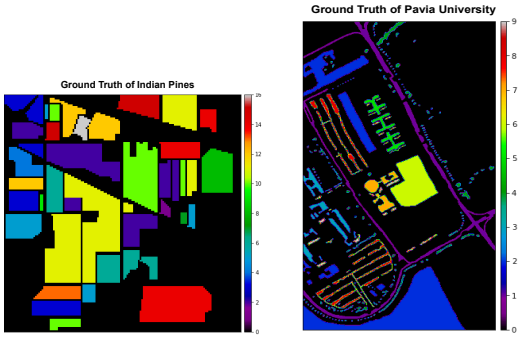
$$\hat{l} = \arg\max_i s_i \tag{4.2}$$

The forthcoming sections dig into the pre-processing of data followed by the custom-built feature extraction network.

4.3.2 Preprocessing

The validation of our model was conducted using two widely recognized HSI datasets obtained from [302] dataset, including IP and PU. Figures 4.2a and 4.2b present the spectral bands and ground truth distribution for the IP dataset, offering insights into the dataset's spectral characteristics. Similarly, Figures 4.3a and 4.3b provide visualizations of the spectral bands and ground truth distribution for the PU dataset. These visual representations contribute to a better understanding of the datasets and their characteristics, serving as essential components in the validation process of our proposed model.

Due to the high dimensional resolution of hyperspectral images, processing them entails increased computational complexity [303]. To address this, preprocessing plays a crucial role in selecting the appropriate bands and reducing the data's dimensionality and complexity. As depicted in Figure 4.4, the original spectral-spatial dimensions of the IP dataset were 145 * 145 * 224. Following the removal of noisy bands by [304], the preprocessed data was reduced to 145 * 145 * 220, with 145 * 145 representing the spatial dimension and 220 denoting the number of spectral bands in the IP dataset. Each band contained 145 * 145 = 21,025 pixels, resulting in a total of 220 bands. This data was then split, allocating 10% for training samples and 90% for testing samples. Following this division, the total number of training samples amounted to 2,102, with testing conducted on 18,923 samples.



(a) Dataset: IP (b) Dataset: PU

Figure 4.3: Ground Truth Images

For the PU dataset, the original hyperspectral cube possessed spectral-spatial dimensions of 610 * 340 * 103. After extracting 103 bands, the resulting image was scaled to 207,400 * 103. For training purposes, 20,740 samples were utilized, and testing was carried out on 186,660 samples. A similar extraction process was applied to the ground truth images, which had a spatial dimension of 145 * 145 pixels. As the ground truth serves as the label or category for each data point, the final data shape was 21,025 * 221, including bands and the target label for the IP dataset. Meanwhile, for the PU dataset, it was 610 * 340 * 104 = 207,400 * 104 pixels. This preprocessing ensures that the data is appropriately formatted for effective training and testing of the proposed model.

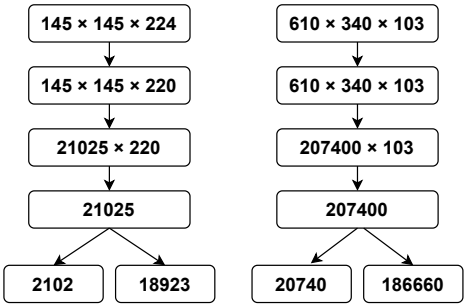


Figure 4.4: Preprocessing for IP and PU dataset

4.3.3 Components of Similarity Model

The similarity model comprises two input layers simultaneously receiving two separate images. The feature extraction approach retrieves features from both data points, and concatenates them using the concatenate layer, followed by a fully connected network including dense and dropout layers and a sigmoid activation that aids in image classification with a probabilistic similarity score.

As illustrated in Figure 4.5, images A and B undergo the feature generation block, producing encoded feature vectors f(a) and f(b) through CNN. The similarity score between

feature vectors is then calculated using D(a, b), representing the Euclidean distance on f(a) and f(b) as depicted in equation 4.3.

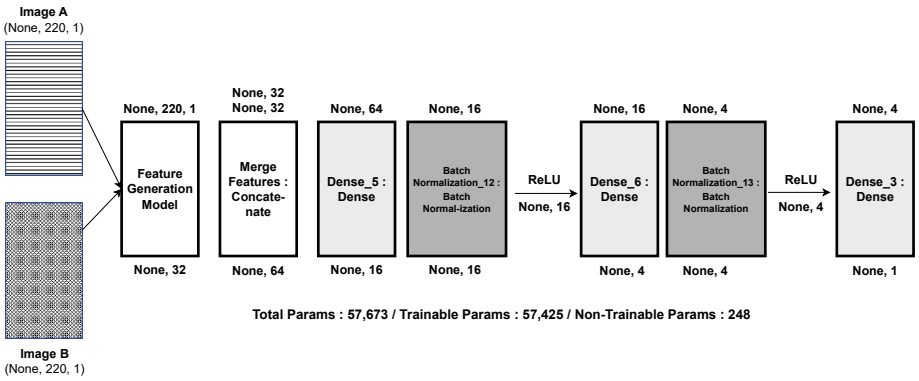


Figure 4.5: Layer-wise breakdown of proposed model

$$D(a, b) = \|f(a) - f(b)\|_2 \tag{4.3}$$

Algorithm 2 GenRandomBatch(inGroups, BatchHalfsize)

Initialize empty lists: OutImgA, OutImgB, OutScore
Create a list allGroups containing indices for all input data groups
for each matchGroup in [True, False] **do**
  Randomly choose batchHalfsize indices from allGroups and assign them to groupIdx
  for each cIdx in groupIdx **do**
    Randomly choose a sample from the corresponding group in inGroups and add it to OutImgA
  end for
  if matchGroup **then**
    Set bGroupIdx to be the same as groupIdx
    Append 1 to OutScore batchHalfsize times
  else
    Create an empty list nonGroupIdx
    for each cIdx in groupIdx **do**
      Randomly choose a different group index from allGroups and add it to nonGroupIdx
    end for
    Set bGroupIdx to be nonGroupIdx
    Append 0 to OutScore batchHalfsize times
  end if
  for each cIdx in bGroupIdx **do**
    Randomly choose a sample from the corresponding group in inGroups and add it to OutImgB
  end for
end for
Stack OutImgA, OutImgB, and OutScore along first dimension and return result

The Mathematics behind Generating Random Batches

The process of generating random batches in a Siamese network involves selecting pairs of images from the dataset, labelling them as positive or negative, and assigning similarity scores accordingly. Positive pairs consist of two images from the same category, labeled as similar with a designated similarity score typically set to 1. Conversely, negative pairs are formed by pairing an image with another from a different category, labelled as dissimilar with a similarity score of 0. This process is iterated to create batches of training data, where each batch contains a mix of positive and negative pairs along with their corresponding similarity labels. The provided algorithm outlines this process succinctly: it initializes empty lists for output images, paired images, and similarity scores, selects random indices from the input data groups, and iterates through each group to create positive and negative pairs accordingly. Finally, the generated pairs are stacked along with their labels to form the training batch, ensuring the Siamese network learns to discriminate between images with similar and dissimilar characteristics during training. Algorithm 3 briefs the process of generating random batches.

Inculcated Feature Extraction Module

The Feature Extraction Module is specifically crafted to draw out discriminative features from the input image. This tailored module consists of four blocks, each comprising a convolutional layer followed by a batch normalization layer, as illustrated in Figure 4.6. The process begins with the input image passing through the convolutional layer, applying convolutions that effectively reduce the depth of the channels. Following every two such convolution and batch normalization blocks is a max-pooling layer. Max-pooling serves as a technique to condense the features within a segment of a feature map generated by a convolution layer. Consequently, the max-pooling layer produces a feature map containing the most discriminative and variant features from the preceding feature map. The subsequent flattened layer transforms this into a one-dimensional vector.

To counter overfitting, the network undergoes fine-tuning with batch normalization and dropout layers as inspired by [136]. Batch normalization enhances linearity, and the subsequent max-pooling layer reduces computational complexity. After max-pooling, data undergoes flattening into a single-dimensional entity before entering the dense layer. A dropout layer with a 0.5 rate is introduced to counteract overfitting. This combination of layers and techniques strikes a balance between model complexity and regularization, ensuring effective feature extraction while preventing overfitting. As depicted in Figure 4.6, the input to the model starts at dimensions 220 * 1 and gets reduced to 32 * 1 after passing through the model. The customization of the feature extraction module involves placing the right parameters for optimal classification accuracy with minimal runtime.

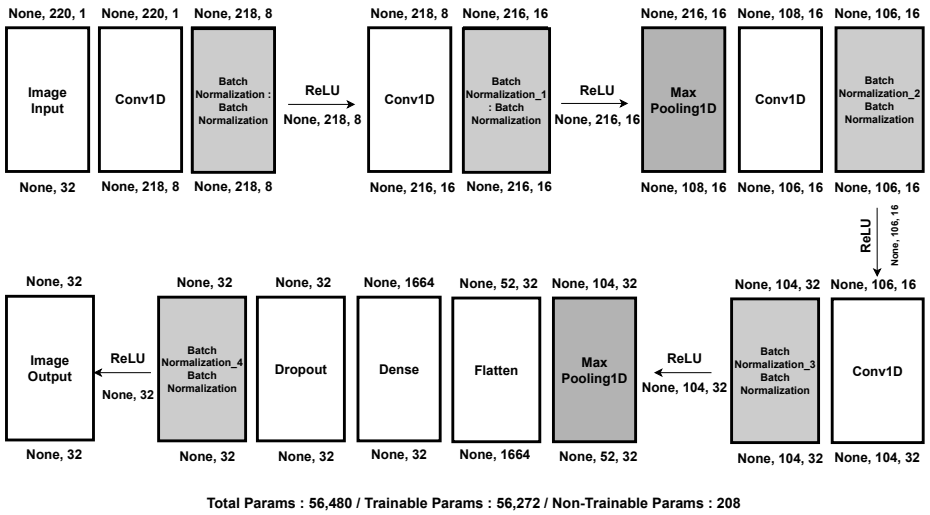


Figure 4.6: Inculcated Feature Extraction Model

The band extraction process is detailed in Figure 4.7. The IP dataset, presented as a cube with dimensions 145 * 145 * 220 (Width * Height * Spectral Band), undergoes band selection, as shown in Figure 4.7 (top middle). The corresponding ground truth image, with distinct labels for each category represented in different spectra, is displayed in Figure 4.7 (top right). The band extraction process, illustrated in the bottom left of Figure 4.7, assigns a class to each pixel based on the ground truth, signifying data extraction corresponding to various data points or pixels through spectra.

Binary Cross Entropy for Similarity Score Calculation

In the context of a Siamese network, which is often used for tasks like image similarity or dissimilarity, the binary cross-entropy, represented by Equation 4.4 loss plays a crucial role in quantifying how well the network is performing in distinguishing between similar and dissimilar pairs of images.

$$-\frac{1}{N}\sum_{i}^{n}\sum_{j}^{m} y_{ij}\log\left(p_{ij}\right) \tag{4.4}$$

In this equation 4.4, N is the total number of image pairs in the dataset, n is the index for each image pair in the sum, $m = 2$ because we are dealing with a binary classification problem: determining whether the pair is similar ($y_{ij} = 1$) or dissimilar ($y_{ij} = 0$), y_{ij} is the ground truth label for the i-th image pair. It is binary, indicating whether the pair is similar or dissimilar, p_{ij} is the predicted probability that the i-th image pair is similar.

For each image pair i, the binary cross-entropy loss is calculated based on the predicted probability (p_{ij}) and the ground truth label (y_{ij}). The inner sum ($\sum_{j}^{m}$) is over the two classes (similar or dissimilar), summing up the contribution from each class. The outer sum

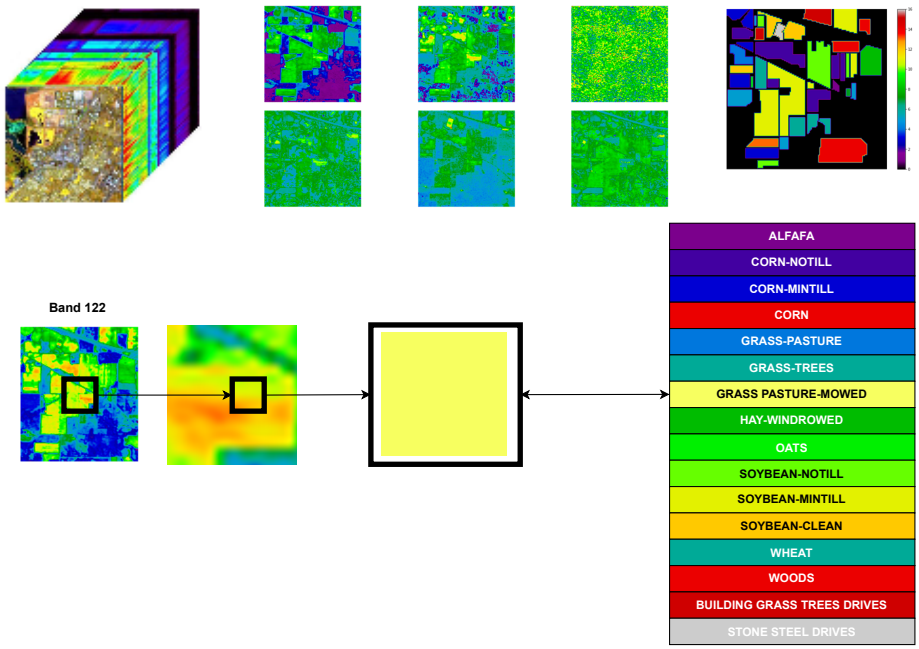


Figure 4.7: *Left to Right, Top to Bottom* (a) IP HSI Cube (b) Randomly selected spectral bands (c) Ground Truth depicting 16 classes (d) Band Extraction and Similarity matching process

$(\sum_{i}^{n})$ is over all image pairs, summing up the losses for each pair. The negative sign and the fraction $-\frac{1}{N}$ turn the task of minimizing the loss into an optimization problem. The goal of proposed Siamese is to minimize this binary cross-entropy loss during training. The network adjusts its parameters to improve the agreement between predicted probabilities and true labels, making it proficient in distinguishing between similar and dissimilar image pairs. The binary cross-entropy loss is a crucial component of the training process, guiding the network to learn effective representations for the given similarity or dissimilarity task.

4.4 Results and Discussion

This section focuses on the application of the proposed Siamese CNN for hyperspectral dataset classification, employing two benchmark HSI datasets: PU and IP.

4.4.1 Experimental Settings

The Siamese network, configured with a batch size of 32, generates random batches by halving the equal size. Network tuning utilizes Adam optimizer with a learning rate of 0.0001 and binary cross-entropy as the hyperparameter for mean absolute error calculation. During training, random batches processed through the Feature Generation model incorporate convolutions with a kernel size of 3 and ReLU activation, alongside a dropout parameter of 0.5 for overfitting prevention. TensorFlow handles the model implementation,

while Scikit Learn is utilized for metrics evaluation.

4.4.2 Train and Test Split

For the IP dataset with spatial and spectral dimensions of 145 * 145 * 220, training encompasses 10% of the samples, reserving the remaining 90% for testing. The Similarity Model undergoes training for 110 epochs, utilizing a batch size of 64 and 500 steps per epoch.

To ensure a comprehensive and fair comparison, three distinct categories were defined:

- This group encompasses widely-used machine learning techniques such as Support Vector Machine (SVM), Decision Trees (DT), and Logistic Regression (LR). While these models yield reasonable results, there exists potential for enhancing categorization accuracy.

- The second category includes deep learning methods, known for their remarkable performance in hyperspectral image categorization. Existing models like 3DAES [58], AwGCN [291], 3DAES naive [58], CAP1-PU [51], Hu et al. [293], DGCN [150], Dual-SCNN-AT-Mix-up [49], Ap3D [296], SSRHE [290] are considered for the PU dataset, while CNN-PPF [301], SaSiResnet-2 [228], Hu et al. [293], CAG [298], w-S-SAN [50], DGCN [150], CNN [50], Ap3D [296], SSRHE [290] are evaluated for the IP dataset.

- The third category comprises the results obtained from the proposed network. This novel approach aims to contribute to the field of hyperspectral image classification, and its performance is thoroughly compared with both traditional machine learning models and existing deep learning methods.

4.4.3 Experiment 1: PU

Initially, the experiment was conducted on the PU dataset. The comparative analysis of the proposed network with the studies' suggested literature is visually depicted in Figure 4.8, utilizing OA, AA, KC as the evaluation metrics. The classification accuracy for each class is detailed in Table 4.2. The results notably illustrate the exceptional classification performance of the proposed network, showcasing the highest OA and superior accuracies for most categories. In contrast, when employing machine learning models like SVM, LR, and DT on the same dataset, the OA dropped to 55%, 83%, and 74%, respectively, even with a substantial 80% training data. Remarkably, our proposed model attains impressive accuracy with just 10% of the data. Notably, for classes such as "Painted Metal Sheets," "Bitumen," and "Shadows," the model achieves accuracy levels of 95% or higher.

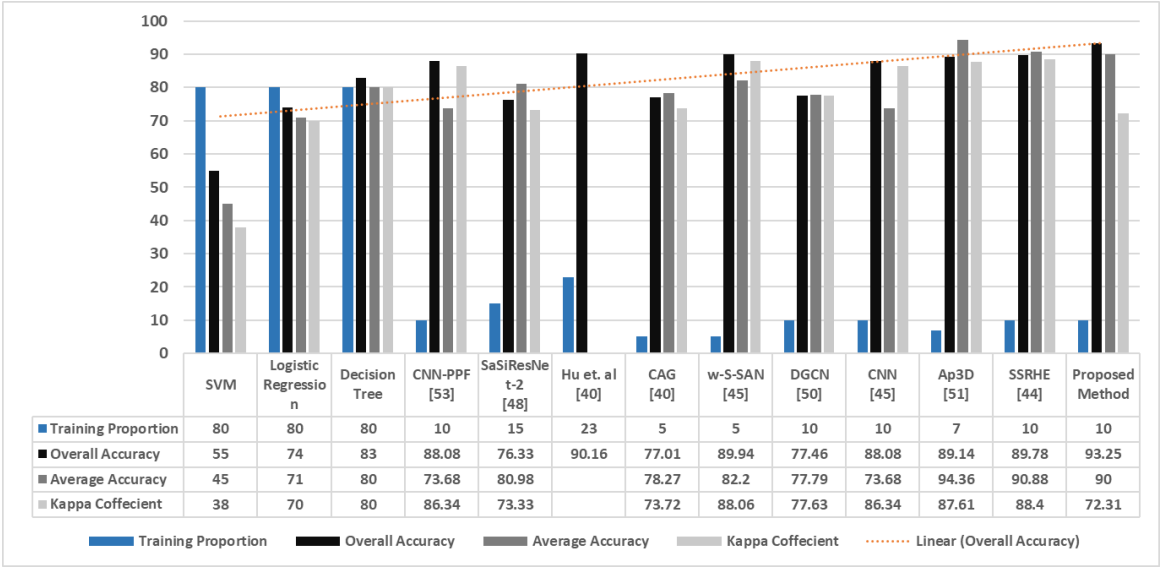


Figure 4.8: Comparison Chart for PU Dataset

Table 4.2: Class-wise Accuracy Comparison for PU

	Category	**SVM**[305]	**CAPS1-PU**[51]	**Hu et. al**[293]	**Dual-SCNN-AT-Mix-up**[49]	**SS-RHE** [290]	**DGC-NDC** [150]	**Ap3D**[296]	**Proposed**
1	Asphalt	87.47	89.43	87.34	80.76	91.19	78.99	91.00	**92.27**
2	Meadows	93.86	96.86	94.63	89.96	98.12	80.91	97.13	**99.49**
3	Gravel	84.23	92.29	86.47	69.51	78.6	79.43	78.23	**98.62**
4	Trees	97.80	97.97	96.29	96.24	89.26	79.23	98.11	**99.49**
5	Painted Metal Sheets	99.86	100.0	99.65	99.49	**99.77**	80.97	98.96	97.08
6	Bare Soil	95.19	96.09	93.23	77.60	85.22	81.03	65.58	**99.33**
7	Bitumen	94.82	**98.31**	93.19	77.23	90.18	80.60	91.88	94.59
8	Self-Blocking Bricks	92.58	87.96	86.42	80.60	79.33	79.30	95.22	**98.94**
9	Shadows	99.93	99.85	**100.0**	97.59	100	78.33	98.42	95.48
	Overall Accuracy	92.97	94.90	92.56	85.81	92.59	79.87	91.37	**95.17**
	Average Accuracy	93.97	95.42		85.44	90.55	79.84	90.50	92.00
	Kappa Score		93.24		81.36	90.2	77.09	88.45	80.00

4.4.4 Experiment 2: IP

The second phase of the experiment focused on the IP dataset. The class-wise accuracy breakdown is presented in Table 4.3, and a comprehensive visual representation of the evaluation metrics is showcased in Figure 4.9. Demonstrating remarkable efficiency, the proposed model attains an impressive 93.25% OA with a mere ten percent of the dataset allocated for training. Upon closer inspection of the results, it is evident that certain classes have limited training data, such as ”Soybean-clean,” ”Oats,” ”Stone-Steel-Towers,” and ”Grass-pasture-mowed,” yet they still exhibit a commendable level of classification accuracy. Our model secures the second-best performance among the compared models, with the highest accuracy model not considering classes with limited samples.

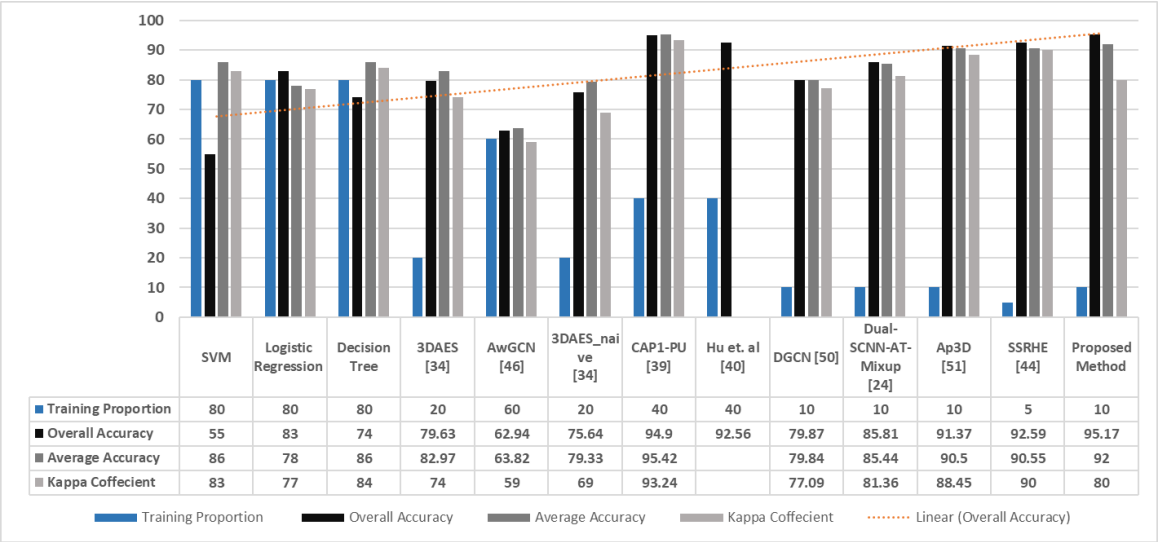


Figure 4.9: Comparison Chart for IP Dataset

Table 4.3: Class-wise Accuracy Comparison for IP

	Category	**EMP-SVM**[305]	**CNN-PPF**[51]	**SaSiRes-net**[49]	**SS-RHE** [290]	**DGC-NDC** [150]	**Ap3D**[296]	**Pro-posed**
1	Alfalfa	81.82	-	**95.88**	94.44	81.10	80.20	75.05
2	Corn-notill	78.85	92.99	66.61	88.17	76.20	90.94	**99.15**
3	Corn-mintill	90.75	96.66	81.92	80.46	73.82	**100**	99.46
4	Corn	73.96	-	80.51	84.04	75.50	93.29	79.15
5	Grass-pasture	95.10	98.58	89.18	96.55	73.83	97.27	88.01
6	Grass-Trees	96.99	100	89.71	97.02	78.46	99.77	**99.10**
7	Grass-pasture-mowed	87.50	-	100.00	100	80.80	88.45	81.48
8	Hay-windrowed	99.07	100	95.65	98.60	81.75	83.29	95.81
9	Oats	6.67	-	36.00	80.00	65.56	87.06	**1.0**
10	Soybean-notill	86.81	96.24	77.68	83.89	77.16	100	**98.16**
11	Soybean-mintill	96.80	87.80	59.23	89.50	78.44	95.10	**97.65**
12	Soybean-clean	85.75	98.98	75.96	83.71	81.07	96.97	**97.95**
13	Wheat	97.96	**-**	**98.59**	100	79.34	100	72.39
14	Woods	97.66	99.81	90.08	95.52	76.81	97.44	**1.0**
15	Buildings-Grass-Trees-Drives	92.03	-	81.14	83.57	79.00	**100**	95.37
16	Stone-Steel-Towers	68.85	-	77.47	97.59	85.80	100	89.71
	Overall Accuracy	91.09	**94.34**	76.33	89.78	77.46	89.14	***93.25***
	Average Accuracy	83.53	**96.78**	80.98	90.88	77.79	94.36	***90.00***
	Kappa Score	89.78	94.73	73.33	88.4	77.63	87.61	72.31

Table 4.4: Training and Testing Time taken

PU (610 * 340 * 103)			**IP (145 * 145 * 220)**		
Total Parameters	Training Time	Testing Time	Total Parameters	Training Time	Testing Time
2.576 * 104	300	11	5.648 * 104	280	7

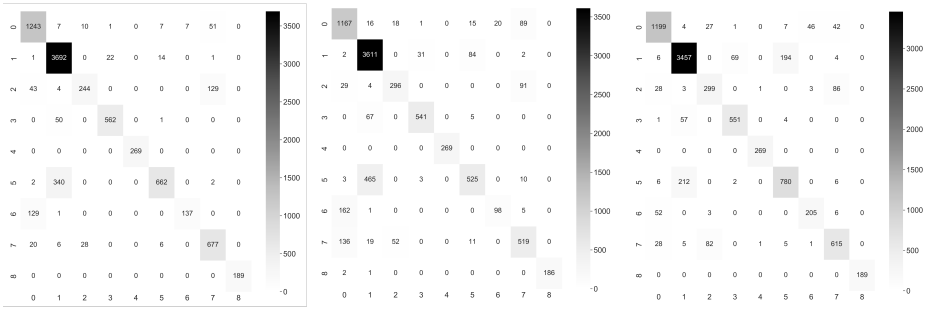


Figure 4.10: (Left to Right) Confusion Matrix, Dataset: PU for (a) SVM (b) LR (c) DT

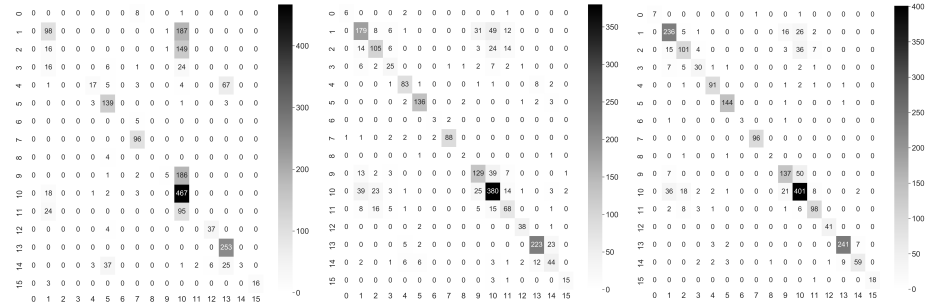


Figure 4.11: (Left to Right) Confusion Matrix, Dataset: IP for (a) SVM (b) LR (c) DT

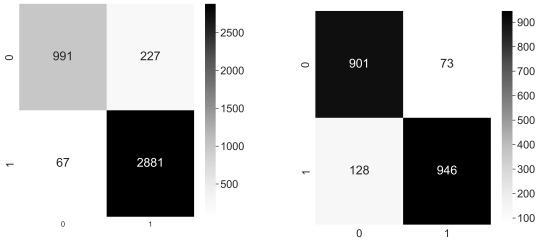


Figure 4.12: Confusion Matrix for PU and IP

4.4.5 Confusion Matrices and Convergence

The comparison between correctly and incorrectly predicted classes is visually represented in Figures 4.10 and 4.11 for PU and IP for the category of machine learning models, while Figures 4.12 and 4.13 illustrate the outcomes for our proposed Siamese network model. The confusion matrices of traditional machine learning models align with the accuracy chart, indicating a considerable number of misclassifications. Figure 4.12 specifically exhibits the confusion matrix tailored for a multilabel problem in our proposed Siamese network model. This 2*2 matrix captures pairwise multilabel confusion, emphasizing the model's behaviour under binary cross-entropy, where any dissimilarity score less than 1 is considered a misclassification.

Table 4.4 and Figure 4.13 provide insights into the training and testing times using Scikit Learn and TensorFlow. The training duration for PU and IP amounted to 300 and

280 seconds, respectively, with testing requiring 11 and 7 seconds. The proposed Siamese model demonstrates efficient convergence, indicating the potential for further optimization in training and testing times through the adoption of frameworks like Caffe.

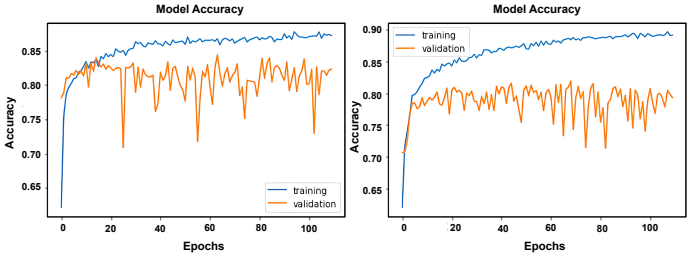


Figure 4.13: *(Left to Right)* Model Accuracy for PU and IP

4.4.6 Discussion

The proposed deep Siamese network with handcrafted feature extraction demonstrates strong performance on the benchmark hyperspectral image datasets, achieving state-of-the-art classification accuracies even with limited training data. Several key strengths of the model contribute to its success:

- **Few-Shot Learning Capability:** The Siamese architecture enables effective few-shot learning by comparing input image pairs and determining their similarity, reducing reliance on large labelled datasets that are challenging to obtain for hyperspectral images.
- **Handcrafted Feature Extraction:** The custom-designed feature extraction module extracts discriminative features tailored to hyperspectral data, enhancing the model's ability to capture relevant spectral and spatial information.
- **Robustness to Imbalanced Data:** The results show the model's resilience to imbalanced class distributions, a common issue in hyperspectral datasets, indicating its potential for real-world applications.
- **Computational Efficiency:** Despite its strong performance, the model architecture is relatively lightweight, with faster convergence times compared to some existing models, making it more computationally efficient.

However, the proposed model also has some limitations and potential failure cases that should be addressed:

- **Sensitivity to Hyperparameters:** The performance of the model may depend heavily on the choice of hyperparameters, such as the architecture of the feature extraction module and the optimization settings, requiring careful tuning for each dataset.

- **Dependence on Spectral Preprocessing:** The model's performance may be sensitive to the spectral preprocessing steps, such as band selection and noise removal, which can vary across different datasets and applications.

To further improve the proposed model and address these limitations, several future research directions can be explored:

- **Interpretability Enhancement:** Techniques such as attention mechanisms, saliency maps, or explainable AI methods could be incorporated to improve the interpretability of the model's decisions, especially for critical applications like environmental monitoring or target detection.
- **Automated Hyperparameter Tuning:** Implementing automated hyperparameter optimization strategies, such as Bayesian optimization or evolutionary algorithms, could alleviate the burden of manual tuning and potentially improve model performance.
- **Domain Adaptation and Transfer Learning:** Investigating domain adaptation and transfer learning techniques could enable the model to leverage knowledge from related datasets or domains, potentially reducing the need for large amounts of labeled data in the target domain.
- **Ensemble and Multi-Modal Approaches:** Combining the proposed model with other deep learning architectures or incorporating multi-modal data (e.g., LiDAR, radar) could lead to more robust and accurate hyperspectral image analysis systems.
- **Distributed and Parallel Training:** Exploring distributed and parallel training strategies could further improve the computational efficiency of the model, enabling scalability to larger datasets and higher spatial or spectral resolutions.

Overall, the proposed deep Siamese network with handcrafted feature extraction represents a promising approach for HSIC, particularly in scenarios with limited labelled data. By addressing its limitations and leveraging future research directions, the model's performance and applicability could be further enhanced, contributing to the advancement of hyperspectral remote sensing and related fields.

4.5 Summary

The proposed few shot classification model, based on a Siamese network with enhanced feature extraction, outperforms current models in the literature, achieving state-of-the-art classification accuracy. The fine-tuned feature extraction block utilizes hyperparameters to extract discriminative features, contributing to improved classification accuracy even in

scenarios where the network is trained with minimal data. This fine-tuning, specifically tailored for HSI features, results in superior performance compared to predefined feature extraction models. The experimental results on two benchmark hyperspectral datasets highlight the effectiveness of the proposed network, showcasing an overall accuracy of 95.17% for the PU dataset and 93.25% for the IP dataset, even when trained with a limited amount of data. Notably, the network exhibits robustness to imbalanced data. Furthermore, compared to existing models in the literature, the feature extraction module, despite being less dense, delivers superior results and demonstrates faster convergence.

CHAPTER 5

A 3D CONVOLUTIONAL AUTOENCODER EMBEDDED SIAMESE ATTENTION NETWORK FOR CLASSIFICATION OF HYPERSPECTRAL IMAGES

Supervised deep learning networks have demonstrated exceptional performance in HSIC, capitalizing on their capacity for end-to-end optimization and leveraging their strong potential for nonlinear modelling. However, labelling HSIs, on the other hand, necessitates extensive domain knowledge and is a time-consuming and labour-intensive exercise. To address this issue, this chapter introduces a novel semi-supervised network constructed with an autoencoder, siamese action, and attention layers that achieves excellent classification accuracy with labelled limited samples. The proposed convolutional autoencoder is trained using the mass amount of unlabelled data to learn the refinement representation referred to as 3D-CAE. The added siamese network improves the feature separability between different categories and attention layers improve classification by focusing on discriminative information and neglecting the unimportant bands.

5.1 Introduction

While supervised models excel in classifying HSI, acquiring labelled samples continues to pose a challenge in this domain [306], as the labelling remains a manual, time-consuming, and labour-intensive task, which necessitates the expertise of individuals to identify and label each pixel in the image. The semi-supervised models require less manual data labelling, which saves time and money [39, 57], However, their performance has not been at par with the supervised models. To address this, various deep learning methods such as few-shot learning (FSL), zero-shot learning have been explored for HSIC [307]. This exploration makes the model semi-supervised and also makes notable advancements in dealing with

the limited data challenge and also to improve the classification accuracy. Models like these enable the utilization of a few labelled samples to learn generalized representations and improve classification accuracy on unseen data points [308, 309]. Building upon these advancements, this chapter aims to further improve the performance of semi-supervised models by integrating the strengths of few-shot learning and Siamese networks not only on same-domain but also on cross-domain datasets. More specifically, the objective is to tackle the challenges posed by limited data and enhance the classification accuracy, while also taking into account concerns such as overfitting and gradient vanishing. The contribution of the proposed work can be summarized in the following points

- The proposed network distinctively amalgamates Three-Dimensional (3D) convolutional autoencoder, Siamese network, and attention layers—setting it apart for superior performance in cross-domain and same-domain HSI classification, substantiating its novelty through experimental contributions. The crux of our innovation is the seamless integration and synergy of key components—3D convolutional autoencoder, Siamese network, and attention layers—setting our approach apart for enhanced performance in both cross-domain and same-domain HSI classification.

- Our model's semi-supervised nature is a pivotal strength, effectively overcoming the challenge of limited labelled data in HSI. This aspect is particularly valuable for real-world applications where obtaining labelled data is often resource-intensive. The 3D-CAE significantly contributes to unsupervised feature learning, generating features that may lack class separability but play a crucial role in the overall effectiveness of the model.

- The Siamese network backed up with binary cross entropy adds value by enhancing feature separability, and leveraging three-dimensional convolutions to reduce intra-class distances and increase interclass distances.

- To efficiently learn with the available HSI three-dimensional hyperspectral data, the embedded attention layers in the network, which include both spectral and spatial, reduce the duplicate information by emphasizing only significant channels and pixels and suppressing the less informative ones.

- The effectiveness of the proposed semi-supervised spectral-spatial network is verified through experiments conducted on benchmark HSI datasets. The results demonstrate that the network achieves exceptional overall accuracy, not only on same-domain datasets but also on cross-domain datasets.

5.2 Related Models

As discussed above, there exist various supervised deep learning models for HSI classification with very high classification accuracy. However, these models require substantial labelled data, which is difficult to acquire and training such models with limited samples risks overfitting. The network memorizes training data instead of learning patterns, hindering gradient propagation. In other words, the network's parameters might not update properly, leading to slow convergence or even stagnation in the learning process. This can hinder the model's ability to capture complex patterns in hyperspectral images [52]. Consequently, it may struggle to accurately classify new, unseen hyperspectral images [266, 51]. To address this, numerous supervised and semi-supervised deep learning models for HSI classification with limited or no samples have been proposed and their performance has been compared in the following table :

In 2016, one of the most noteworthy works was introduced by [11], who introduced a supervised three-dimensional convolutional neural network (3D-CNN) model specifically designed to tackle the challenges associated with limited data availability and class imbalance in HSI classification. The work also employs dropout and L2 regularization to mitigate the overfitting and vanishing gradients problem usually encountered in the erstwhile deep-learning models for HSI classification. In a similar vein, [5] discusses a spectral-spatial unified network (SSUN) having a unified objective function with the incorporation of multiscale long-short-term memory (LSTM). The experimental findings presented in the paper underscored its effectiveness in achieving accurate HSI classification, albeit with a high computational cost due to the model complexity. To mitigate the issue of vanishing gradients, researchers proposed a spectral-spatial residual network (SSRN) [310], which leverages residual blocks to enable more effective gradient backpropagation. This approach exhibits excellent performance when tested on datasets within the same domain. However, the SSRN fails to maintain its performance when extended to cross-domain datasets. [61] address this issue by presenting a deep cross-domain few-shot learning model (DCFSL). They demonstrate that leveraging few-shot learning (FSL) techniques greatly aids in transferring knowledge from one domain to another. More specifically, DCFSL learns on one domain of datasets and extends this learning to unseen domains by following the meta-learning approach. Thus, DCFSL significantly enhances classification capabilities in scenarios where labelled samples are scarce. Afterwards, many FSL-based deep learning models [51, 6, 61, 61, 47] for HSI classification have been introduced to further enhance the performance with limited labelled samples.

Among the aforementioned FSL-based HSI classifiers, the Siamese network introduced by [6] has gained more attention because of its ability to reduce the involved network parameters and reduce the overfitting problem. Therefore, various variations of the Siamese

Table 5.1: Studies on Semi-Supervised Models

Ref.	Model	Learning	Dataset	Training Ratio	OA	AA	KC
[11]	3D-CNN	Supervised (Su)	IP	0.17	97.56	99.23	97.02
			PU	0.1	99.54	99.66	99.41
			KSC	0.1	96.31	94.68	95.9
[5]	SSUN	Su	PU	0.1	99.46	99.62	99.26
			IP	0.17	98.4	99.27	98.14
			KSC	0.1	97.71	97.17	97.45
[310]	SSRN	Su	IP	0.3	99.19	98.93	99.07
			PU	0.2	99.79	99.66	99.72
			KSC	0.3	99.61	99.33	99.56
[61]	DCFSL	Semi-Supervised (S-Su)	PU	205	83.65	83.77	78.7
			IP	205	66.81	77.89	62.64
			SA	205	89.34	94.04	88.17
[61]	S3-Net	S-Su	SA	320	95.95	98.23	95.49
			IP	320	91.54	95.58	90.38
			PU	180	96.27	96.91	95.11
[47]	DSR-GCN	S-Su	HU	75	81.11	82.40	79.57
			SA	180	97.35	96.64	97.04
			IP	180	82.29	88.64	79.94
			PU	45	89.49	91.01	86.39
[311]	S3BoF	Su	SA	0.05	99.81	99.82	99.79
			IP	0.05	96.82	95.28	96.38
			PU	0.05	99.73	99.49	99.64
[266]	Dual-SCNN	Su	KSC	0.003	89.95	89.03	88.7
			PU	0.001	84.9	85.28	80.54
[54]	3DAES	S-Su	SA	160	90.39	94.41	89.00
			PU	90	79.63	82.97	74.00
[48]	3DVSCNN	S-Su	PU	90	76.43	75.47	69.00
			SA	160	89.17	94.07	88.00
[312]	CapsNet	Su	PU	0.01	94.9	95.42	93.24
			SA	0.06	99.94	99.95	99.92
[51]	MDL4OW	S-Su	PU	90	76.55	81.42	70.00
			SA	160	82.44	90.57	81.00
[313]	3DCAE	Su	IP	0.1	92.35	92.04	-
			PU	0.05	95.39	95.36	-
			SA	0.05	95.81	97.45	-

network such as [61, 47, 311, 266, 54], each with its handcrafted characteristics, have been introduced, for HSI classification. One of them is proposed by [311] to introduce the Siamese-Based Bag-of-Features (S3BoF) network for improving feature extraction and reducing computational overhead by optimizing negative sample pairs. While the network achieves high overall accuracy, it exhibits slightly poorer performance on categories with a lower number of samples. To address this issue, [61] replace three-dimensional layers with one-dimensional and two-dimensional layers resulting in a reduction in network complexity. However, Xue et al.'s spectral-spatial-siamese-based network (S3Net) suffers from overfitting due to a high number of parameters and limited labelled samples involved. Another notable study conducted by [47] improves the discriminative ability of Siamese using a differentiated scale restricted graph convolutional network (DSR-GCN), which in turn helps in increasing the classification accuracy. In 2020, [266] made a big leap in the direction of achieving high classification accuracy by introducing a dual path Siamese-based CNN (Dual-SCNN) combined with adversarial-based training. By employing this approach, the authors intend to augment the diversity and quantity of training samples, thereby improving the network's capability to generalize and perform well on new and unseen examples. In a similar vein as of [266], [54] propose a three-dimensional semi-supervised Siamese network integrated with an autoencoder (3DAES). By merging the autoencoder with the Siamese network and training it primarily on unlabeled samples, the model effectively addresses the problem of limited labelled samples, as the autoencoder, which was initially proposed by [314] for unsupervised learning, consists of an encoder and a decoder that aim to reconstruct input data accurately, helps in augmenting the data. More specifically, the encoder transforms the input data α into a compressed hidden representation ω, while the decoder decodes the hidden representation into output $\alpha^{'}$. Both the encoder and decoder have symmetric connections with similar patterns. In another work, [313] combines an autoencoder trained on unsupervised samples, with CNN. Thus, the authors aim to leverage the unsupervised learning capability of autoencoders along with the powerful feature extraction capabilities of CNNs, to get a boost in classification accuracy. Further, Mei et al.'s model aligns well with the nature of HSI data and demonstrates promising results. To further enhance the performance, [312] discusses a spectral-spatial capsule network (CapsNet) by employing CNN, having the ability to extract features using weight-sharing convolutional kernels. Afterwards, various extensions to autoencoders such as layer-wise training [315], parameter reduction using PCA [316] and combinations of autoencoders with other models to enhance their capabilities in HSI applications, have been introduced. To briefly summarize the outcomes of the aforementioned models with their classification accuracy on different datasets, along with the current trends in the literature of HSI classification, is presented in Table 6.1.

The structure of this chapter is as follows. Section 5.3 presents the proposed methodol-

ogy detailing description of the 3D-CAE-SiamAtt. The experimental results and discussion are presented in Section 6.4. Finally, Section 6.5 summarizes the work.

5.3 Proposed Methodology

In this section, a novel convolutional autoencoder-based siamese attention network is designed specifically for HSI classification. While antecedent works have individually delved into components like the Autoencoder (3D-CAE), Siamese network, and attention layers, our innovation lies in their unprecedented integration, specifically tailored for the unparalleled enhancement of cross-domain and same-domain HSI classification. The strategic integration of architectural elements, coupled with a customized tuned model incorporating advanced regularization and optimization techniques, leads to unparalleled classification performance.

Our proposed hybrid model, featuring a distinctive combination of encoder and decoder layers with customized parameters, is crafted to optimize HSI classification performance, presenting an experimental and novel contribution. The 3D-CAE, a pivotal component, plays a crucial role in learning data representations using unsupervised data, contributing significantly to the overall effectiveness of the model. The semi-supervised nature of our model is a key strength, addressing the challenge of limited labelled data in hyperspectral image classification. This aspect is vital for real-world applications where obtaining labelled data can be resource-intensive. The Siamese network employs three-dimensional convolutions and enhances feature separability by reducing the intraclass distance and increasing interclass distance, leveraging the high-dimensional and information-rich nature of hyperspectral data. Embedded attention layers, incorporating both spectral and spatial aspects, efficiently learn from three-dimensional hyperspectral data. These layers reduce duplicate information and emphasize only significant channels and pixels using a binary cross-entropy loss function, enhancing the network's performance. Our proposed semi-supervised spectral-spatial network's effectiveness is validated through comprehensive experimentation, demonstrating exceptional performance in HSI classification tasks.

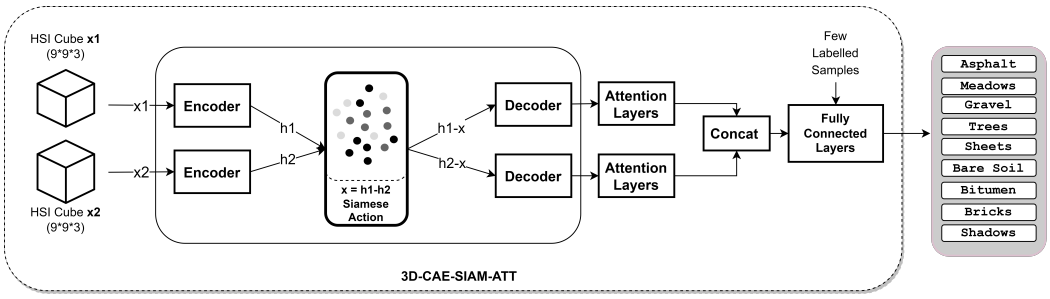


Figure 5.1: Brief Diagram of The Proposed 3DCAE-SIAM-ATT

Figure 5.1 illustrates the overall structure of the proposed model that consists of a convolutional autoencoder embedded with Siamese action and spectral-spatial attention layers. The handcrafted autoencoder is trained with unlabelled samples to learn the best possible representation as discussed in 5.3.1. Next, two branches of the trained autoencoder are created and incorporated to form a siamese network as detailed in 5.3.2. Each of the branches takes a different HSI cube and gives its own reconstructed output which are concatenated together. The concatenated representation is then passed through attention layers to highlight relevant features. Next, the output of the attention layers is used for multi-class classification as discussed in detail in sub-section 5.3.3. At last to summarize the working of the proposed network, Algorithm 4 is provided in subsection 5.3.4.

5.3.1 Three Dimensional Convolutional Autoencoder (3D-CAE)

The proposed architecture for HSI classification consists of a three-dimensional convolutional autoencoder (3D-CAE) along with a Siamese network for similarity learning. Figure 5.2 illustrates the structure of the proposed autoencoder, and Table 5.2 provides an overview of its associated parameters. The overall structure of the model has an encoder and decoder, comprising a convolutional autoencoder. The attention layers are added after the decoder to capture relevant information and enhance the representation learned by the encoder. The attention layers help the autoencoder focus on important features or regions in the input data, improving its ability to reconstruct the input accurately. The encoder takes a hyperspectral cube, say $H_C \in R^{(x \times y \times z)}$ where x and y represent the height and width of the cube, respectively, and z represents the number of spectral bands contained in the cube. The encoder component g_e of the autoencoder extracts discriminative features and produces an encoded feature representation $e_\phi^{H_C} \in \mathbb{R}^d$. The decoder component of the autoencoder then takes $e_\phi^{H_C}$ and reconstructs the hyperspectral image $H_C^{'} \in \mathbb{R}^{(x \times y \times z)}$. During training with unlabeled samples, the autoencoder aims to reduce the loss between the inputted cube and the reconstructed cube using binary cross entropy and improves convergence using the Adam optimizer.

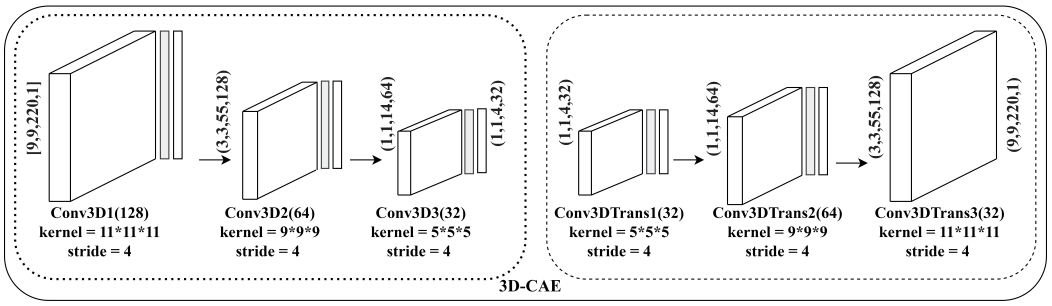


Figure 5.2: 3D-CAE

Table 5.2: Parameters of 3D-CAE

Layers	Type	Kernel Number	Kernel Size	Stride	Padding
CONV1	Conv3D + BN + L2 (0.1) + Activation(ReLU)	N	11×11×11	4×4×4	same
CONV2	Conv3D + BN + L2 (0.1) + Activation(ReLU)	N	9×9×9	4×4×4	same
CONV3	Conv3D + BN + L2 (0.1) + Activation(ReLU)	N	5×5×5	4×4×4	same
DECONV1	Conv3DTranspose + L2 (0.1) + BN + Activation(ReLU)	N	5×5×5	4×4×4	same
DECONV2	Conv3DTranspose + L2 (0.1) + BN + Activation(ReLU)	N	9×9×9	4×4×4	same
DECONV3	Conv3DTranspose	N	11×11×11	4×4×4	same
FC	Fully Connected + Sigmoid	2	-	-	-

The proposed architecture for HSI classification consists of a three-dimensional convolutional autoencoder (3D-CAE) along with a Siamese network for similarity learning. Figure 5.2 illustrates the structure of the proposed autoencoder, and Table 5.2 provides an overview of its associated parameters. The overall structure of the model has an encoder and decoder, comprising a convolutional autoencoder. Attention layers are added after the decoder to capture relevant information and enhance the representation learned by the encoder. These attention layers help the autoencoder focus on important features or regions in the input data, improving its ability to reconstruct the input accurately. The encoder takes a hyperspectral cube, denoted as $H_C \in \mathbb{R}^{(x \times y \times z)}$, where x and y represent the height and width of the cube, respectively, and z represents the number of spectral bands contained in the cube. The encoder component g_e of the autoencoder extracts discriminative features and produces an encoded feature representation $e_\phi^{H_C} \in \mathbb{R}^d$. The decoder component of the autoencoder then takes $e_\phi^{H_C}$ and reconstructs the hyperspectral image $H_C^{'} \in \mathbb{R}^{(x \times y \times z)}$. During training with unlabeled samples, the autoencoder aims to reduce the loss between the inputted cube and the reconstructed cube using binary cross entropy and improves convergence using the Adam optimizer.

The reconstructed outputs of the autoencoder are represented by Equation 5.1, given as follows:

$$H_C^{'} = g_d(g_e(H_C)) \tag{5.1}$$

where g_e refers to the encoding function, which takes an input and maps it to a latent representation or feature vector. It is tasked with extracting distinctive features from the input data. g_d refers to the decoding function, which takes the latent representation produced by the encoding function and reconstructs the input data. It performs the mapping of the latent representation back to the original input space, with the objective of generating a reconstructed version of the input data. The convolutional layer having kernel Ke in equation 5.2

$$Ke \in \alpha^{x_{ke} \times y_{ke} \times z_{ke} \times ke_{in} \times ke_{out}} \tag{5.2}$$

where x_{ke}, y_{ke} and z_{ke} represent the height, width, and depth of the kernel, respec-

tively, having ke_{in} as the input and ke_{out} as the output bands. The value of Ke impacts the network performance greatly. If the value of Ke is too high, the fine-grained subordinate features remain ignored. However, keeping the value of Ke too small increases the computational time due to the slow processing of the HSI cube. After the initial step of dimensionality reduction using principal component analysis and a window size of 9, the pre-processed data is divided into smaller cubes, and padding is applied as needed. The resulting input tensor is then passed through a handcrafted autoencoder. This autoencoder is composed of three-dimensional convolutional layers with filter sizes 128, 64, and 32 respectively, each followed by batch normalization and ReLU activation. The inclusion of the batch normalization layer helps normalize the activations, facilitate gradient propagation, and stabilize the training process, while ReLU introduces non-linearity to the model. The Conv3DTranspose layers in the decoder aid in learning the inverse mapping, allowing the network to generate meaningful and high-fidelity representations of the input hyperspectral data. The choice of kernel sizes, strides, and padding in the Conv3DTranspose layers is optimized to mitigate potential reconstruction issues. To enhance robustness and avoid overfitting, there are regularization techniques, including L2 regularization with a coefficient of 0.1, and batch normalization after each Conv3DTranspose layer. These measures are aimed at promoting stable training and reducing the risk of reconstruction-related artefacts.

The results of the reconstructed cubes are measured using Equation 5.3, where L signifies loss. The similarity between the reconstructed and input cubes becomes higher as the value of L decreases. The autoencoder was trained using the Adaptive Moment Estimation optimizer, with binary cross entropy serving as the loss function.

$$L = ||H_C^{'} - H_C||^2 \tag{5.3}$$

5.3.2 Siamese-Action (SIAM)

The Siamese network is created by combining two instances of the autoencoder, which share identical weights. The Siamese network takes pairs of inputs and learns to predict their similarity. The proposed inculcated Siamese network, depicted in Figure 5.3 takes two parallel inputs in sample pairs. The encoders take a separate HSI cube x_1 and x_2 sliced from input HSI and generates its own embedding $e_\phi^{x_1}$ and $e_\phi^{x_2}$ by mapping it into a lower dimensional metric space. Next, the Euclidean distance $D(x_1, x_2)$ in the spectral space and spatial dimension is calculated using Equation 5.4.

$$D(x_1, x_2) = ||(e_\phi^{x_1} - e_\phi^{x_2})||_2 \tag{5.4}$$

Thereafter, the embedding are modified as $e_\phi^{x_1} - D(x_1, x_2)$ and $e_\phi^{x_2} - D(x_1, x_2)$ and

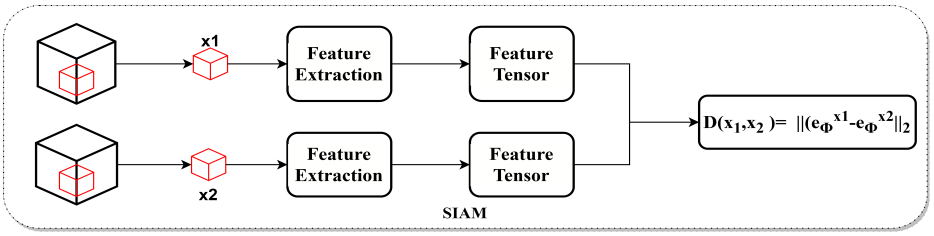


Figure 5.3: Siamese-Action

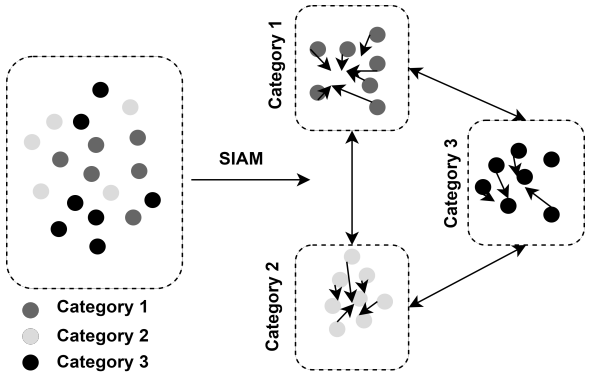


Figure 5.4: Siamese increasing the inter-class and decreasing intra-class distance

passed to embedded decoders for reconstruction. The more identical the two samples are, the more presumably they correspond to the same class. The loss function for the proposed Siamese is binary cross entropy summarized in the following Equation 5.5. Figure 5.4 shows the Siamese network increasing the inter-class distance and decreasing the intra-class distance between various categories.

$$L_{bce} = -gt_n \log(\sigma(\gamma_n)) - (1 - gt_n) \log(\sigma(1 - \gamma_n)) \tag{5.5}$$

In this context, γ_n denotes the predicted probability of pixel n belonging to the other class, while $1 - \gamma_n$ represents the probability of the pixel belonging to the same class. The activation function σ is utilized as the sigmoid function, which compresses the predicted values between 0 and 1, interpreting them as probabilities. The logarithmic terms penalize deviations from the true labels. If the predicted probability aligns with the true label, the corresponding term approaches zero, contributing minimally to the overall loss. However, significant deviations lead to higher losses. The term $-gt_n \cdot \log(\sigma(\gamma_n))$ penalizes entries where the predicted probability $\sigma(\gamma_n)$ for samples belonging to the same class is close to zero. The network enhances separability between different classes by assigning low probabilities to dissimilar samples. The term $-(1 - gt_n) \log(\sigma(1 - \gamma_n))$ penalizes entries where the predicted probability for pixels belonging to the other class is close to zero. This encourages the network to assign higher probabilities to similar samples, bringing samples from the same class together. By optimizing the binary cross-entropy loss function, the Siamese network effectively guides the model to achieve enhanced separability, ensuring

that the learned representations in the feature space align with the spatial and spectral characteristics of the HSI data. Notably, it influences the convergence speed, emphasizing faster convergence due to its ability to guide the network with clearer gradients. Additionally, the quality of learned representations is significantly improved, enabling the network to discern and highlight relevant features in hyperspectral data.

Algorithm 3 GenRandomBatch(inGroups, BatchHalfsize)

```
Initialize empty lists: OutImgA, OutImgB, OutScore
Create a list allGroups containing indices for all input data groups
for each matchGroup in [True, False] do
    Randomly choose batchHalfsize indices from allGroups and assign them to groupIdx
    for each cIdx in groupIdx do
        Randomly choose a sample from the corresponding group in inGroups and add it to OutImgA
    end for
    if matchGroup then
        Set bGroupIdx to be the same as groupIdx
        Append 1 to OutScore batchHalfsize times
    else
        Create an empty list nonGroupIdx
        for each cIdx in groupIdx do
            Randomly choose a different group index from allGroups and add it to nonGroupIdx
        end for
        Set bGroupIdx to be nonGroupIdx
        Append 0 to OutScore batchHalfsize times
    end if
    for each cIdx in bGroupIdx do
        Randomly choose a sample from the corresponding group in inGroups and add it to OutImgB
    end for
end for
Stack OutImgA, OutImgB, and OutScore along first dimension and return result
```

5.3.3 Attention Layers (ATT)

The attention layers were added to suppress the effect of noisy pixels [317]. The involved self-attention mechanism in the attention module extracts the significant pixels by exploiting the association between the hyperspectral pixels contained in the HSI cubes. This can be likened to the human eye's capability to focus on the essential elements in an input image. For the HSI cube $H_C \in \alpha^{x \times y \times z}$ having $x \times y$ as the spatial size and z as the spectral channels, the operation of the attention module is symbolized in Equation 5.6 and Equation 5.7 where $M_{Spe} \in \alpha^{1 \times 1 \times C}$ denoting one-dimensional spectral map and $M_{Spa} \in \alpha^{x \times y \times 1}$ denotes two-dimensional spatial attention map [318].

$$H_C^{'} = M_{spe}(H_C) \otimes H_C \tag{5.6}$$

$$H_C^{''} = M_{spa}(H_C^{'}) \otimes H_C^{'} \tag{5.7}$$

After passing through the attention layers, labelled samples are added to the fully connected layer to rectify the unsupervised samples and train the network to perform the classification process. Figure 5.5 shows the spectral and spatial attention layers involved. Lastly, the addition of a flattening layer is essential for converting the output of the preceding layer, which is multidimensional, into a one-dimensional vector. This conversion is necessary when transitioning from convolutional layers to fully connected layers. The final layer of the model consists of a single output neuron which activates with the sigmoid function. For optimization during training, the popular Adam optimizer is utilized, which dynamically adjusts the learning rate to facilitate convergence.

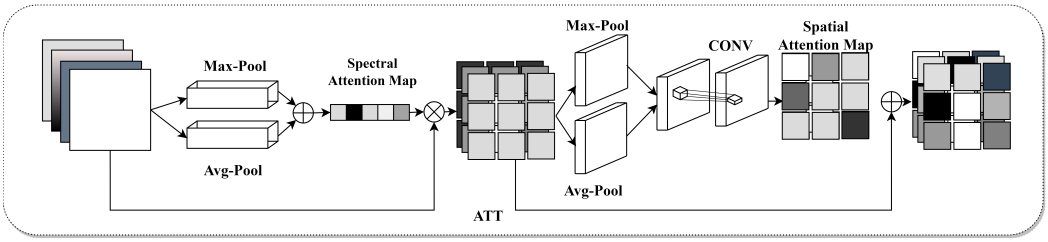


Figure 5.5: Attention Network

The proposed methodology introduces a novel convolutional autoencoder-based siamese attention network tailored for HSIC. It combines encoder and decoder layers to optimize classification performance, addressing limited labelled data challenges. The 3D-CAE component learns representations from unsupervised data, while the Siamese network enhances feature separability. Embedded attention layers efficiently learn from hyperspectral data, reducing duplicate information. Comprehensive experimentation validates the model's effectiveness in HSIC tasks. Algorithm 4 summarizes the model's working, involving autoencoder training, similarity model construction, and attention mechanism implementation.

5.3.4 3DCAE-SIAM-ATT Algorithm

To summarize the working of proposed 3DCAE-SIAM-ATT model, an algorithm 4 which follows a multi-step process is provided below.

The algorithm establishes an autoencoder that comprises an encoder and a decoder. The encoder utilizes CNN architectures to extract significant features from the input images. These extracted features are subsequently employed by the decoder to reconstruct the original images. The autoencoder is trained using the available training data, enhancing its performance through the Adam optimizer and minimizing the binary cross-entropy loss. The training process involves specifying parameters such as the number of epochs, batch size, and learning rate. Once the autoencoder is trained, a similarity model is constructed

Algorithm 4 3DCAE-SIAM-ATT

Training Set $T = \{T_1, T_2, \ldots, T_n\}$; learning rate α, epochs e
Initialize weights and bias $W^i, b^i \; \forall i = 1$ to n
Build **3D-CAE()** inculcated with **SIAM()**
Build **ATT()**
for each iteration **do**
  for each $i \in [1, n]$ **do**
    Forward propagation
  end for
  for each $i \in [1, n]$ **do**
    Fine-tune W^i, b^i
  end for
end for

3D-CAE()
  Generate encoded tensor e_t and reconstructed tensor d_t using layers given in 5.2
  Minimize reconstruction error given in Equation **??**
SIAM()
  Increase $InterCla_D$ and Reduce $IntraCla_D$
  Extract Patches from Training Set x_{i1}, x_{i2} using Algo 3
  Feed x_{i1}, x_{i2} to extract features y_{i1}, y_{i2}
  Calculate Distance **D** using Equation 6.3
  Train the network using L_{BCE} mentioned in 6.4 with **Adam** optimizer
ATT(i,n,a)
  For input tensor i, spectral channels n, attention units a
  a_w = dense(i,a)
  r_w = reshape(a_w,[1,1,1,n])
  a_i = ElementWiseMultiply(i, r_w)
  o_t = sum(a_i)
  Return o_t

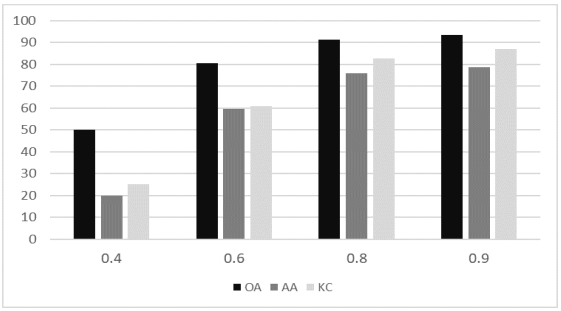
return $W^i, b^i \; \forall i = 1$ to n

on top of it. This model includes a dense layer with sigmoid activation, which predicts the similarity between pairs of images based on their encoded features. The similarity model is then trained using a generator that produces batches of image pairs and their corresponding similarity labels. It is to be noted that an attention mechanism is implemented, which highlights salient regions in the reconstructed images, to enhance the reconstruction process.

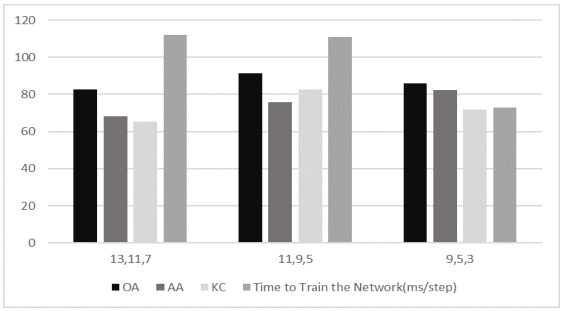
5.4 Experimental Results and Discussion

The experimental results and discussion section of this chapter is divided into two subsections. Section 6.4.2 presents the experimental results, including a comparative analysis with the SOTA models showcasing the performance of the proposed model while delving into ablation studies in section 5.4.1 which are conducted during the experimentation

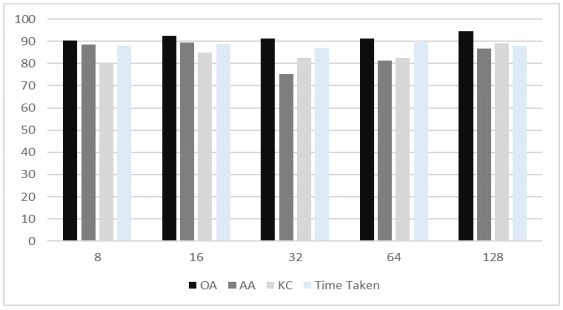
process.



(a) Training Data



(b) Kernel Size



(c) Batch Size

Figure 5.6: Impact of Different Parameters

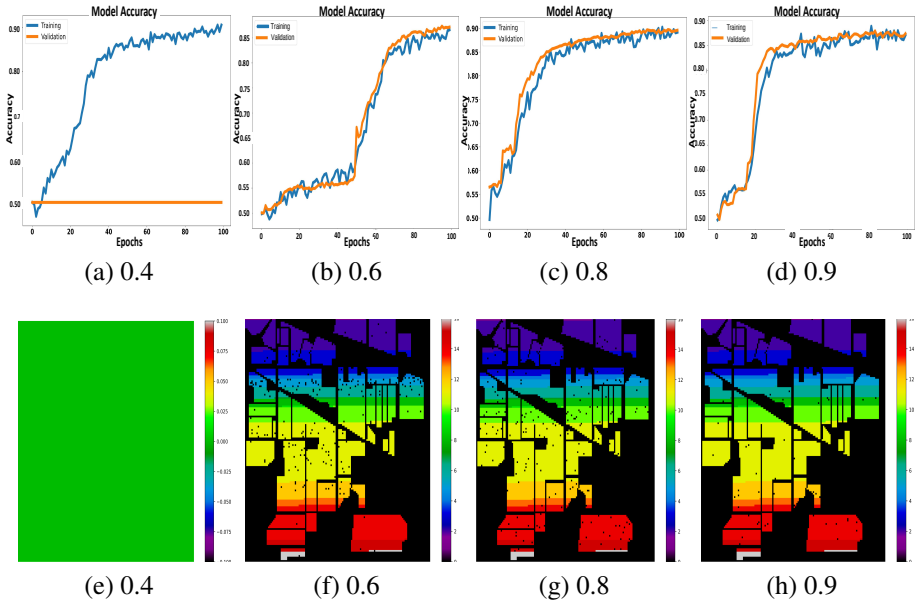
5.4.1 Ablation Studies

To study the impact of change in different parameters and hyper-parameters and other components of the proposed model, ablation studies have been conducted, by considering the IP dataset. This chapter helps in finding the best setting of the model along with the architecture for optimal performance and is organized into six sub-sections detailed as follows. Section 5.4.2 examines the influence of the training ratio on classification performance, Section 5.4.3 explores the impact of different kernel sizes, Section 5.4.4 investigates the consequences of removing attention layers, Lastly Section 5.4.5 delves into the implications of utilizing different optimizers.

5.4.2 Experiment 1: Percentage of Training Samples

The findings in Figure 5.6a demonstrate the influence of varying amounts of training sample sizes on the performance of the proposed model, as measured by OA, AA, and KC. It can be observed that when the network is trained with 40% unsupervised samples, the OA is relatively low. However, as the sample size increases to 60%, there is a noticeable performance improvement. A significant gain in performance is observed when the training samples reach 80%. However, the overall accuracy is only marginal when the training

samples are increased to 90%. These findings suggest that the proposed model achieves the best performance when trained with 80% unsupervised samples.



(a) 0.4 (b) 0.6 (c) 0.8 (d) 0.9

(e) 0.4 (f) 0.6 (g) 0.8 (h) 0.9

Figure 5.7: Classification Accuracy and Classification Maps for varying values of n

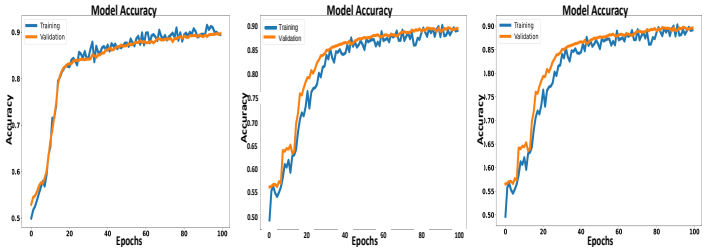
The training performance and classification maps provided in Figure 5.7 serve to reinforce these findings by demonstrating a decrease in misclassifications with an increase in training samples. The model exhibits excellent convergence, signifying its ability to learn and enhance performance. Furthermore, there is minimal or no overfitting, and the model demonstrates strong generalization capabilities, accurately performing on both the training and validation datasets.

5.4.3 Experiment 2: Impact of Kernel Size

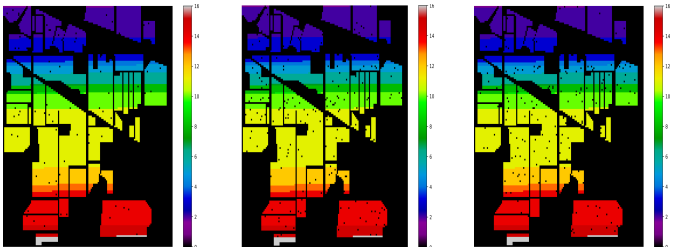
The performance of the network has also been assessed by varying the kernel sizes used in the autoencoder, as presented in Figure 5.6b. The results show that the best overall accuracy was obtained with kernel sizes 11, 9, and 5. Furthermore, Figure 5.8 highlights that the proposed model achieves the most favorable convergence, as depicted in Figure 5.8b, and exhibits the lowest misclassifications, as shown in Figure 5.8e, when the autoencoder employs kernel sizes 11, 9, and 5.

5.4.4 Experiment 3: Removal of Attention Layers

To assess the impact of the integrated attention layers on the model's performance, a comprehensive analysis was conducted. The classification performance was systematically evaluated by removing these attention layers, and the outcomes are illustrated in Figure 5.9. The



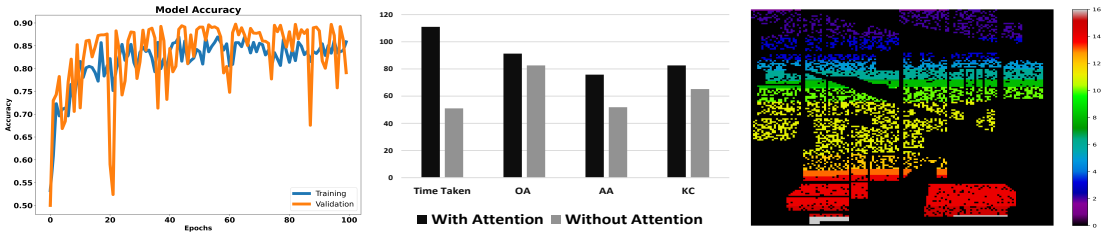
(a) 9,5,3 (b) 11,9,5 (c) 13,11,7



(d) 9,5,3 (e) 11,9,5 (f) 13,11,7

Figure 5.8: Classification Accuracy and Classification Maps for varying sizes of Kernel

results demonstrate a notable decrease in OA, AA and KC for the model without attention. Specifically, the OA, AA and KC metrics dropped from 91%, 75.83% and 82.6% dropped to 82.6%, 51.85%, and 65.21%, respectively. Interestingly, despite the decrease in accuracy, the training process exhibited a significant speed improvement, with the network achieving a training time of only 51 ms/step after the attention module's removal, compared to 111 ms/step with the attention layers. The classification accuracy while training and classification map is included in Figure 5.9.



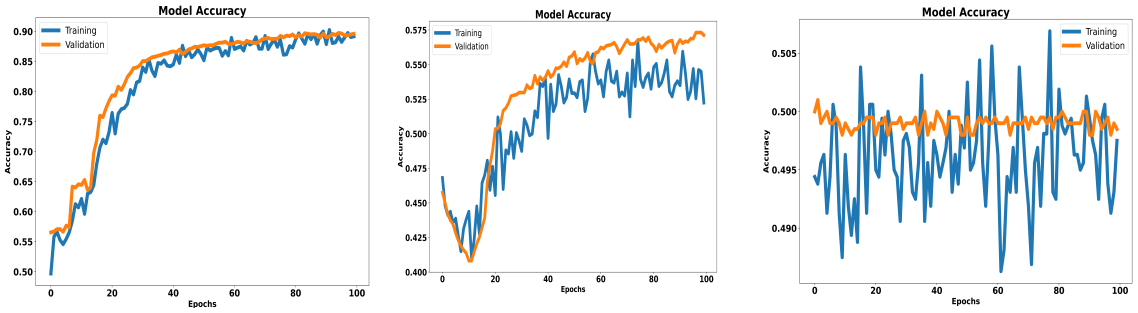
(a) Classification Accuracy (b) Time Taken and OA (c) Classification Map

Figure 5.9: Performance Evaluation after Removal of Attention Layers

This observed reduction in training time without attention layers can be attributed to the streamlined computational process. Attention layers introduce additional computational overhead due to their sequential processing and recalibration operations. When removed, the model's training process becomes more efficient and faster, allowing for quicker convergence during the training phase. However, this efficiency comes at the expense of a reduction in classification accuracy, as the attention layers play a crucial role in capturing and emphasizing relevant features for improved classification performance.

5.4.5 Experiment 4: Effect of using different Optimizers

To explore the performance of the proposed model with different optimizers, the three most commonly used optimizers were chosen for experimentation. The results of this experimentation are presented in Figure 5.10, which indicates that the accuracy of the proposed model is the lowest when optimized with RMSprop, better when optimized with AdaGrad, and the best when optimized with the Adam optimizer.



(a) Adam (b) Adagrad (c) RMSprop

Figure 5.10: Effect of different optimizers on accuracy

5.4.6 Experiment 5: Comparison with other models

To comparatively analyze the performance of the 3D-CAE-SiamAtt model, several SOTA models such as Dual- SCNN- AT-Mixup [49], Dual-SCNN-Mixup [49], CapsNet [51], 3D-CNN [319], SSUN [261], SSRN [136], DCFSL [320] and MDL4OW [58], have been considered and performance has been evaluated on all the aforementioned datasets. These models were chosen for comparative analysis due to their status as state-of-the-art (SOTA) approaches in HSIC. Dual-SCNN-AT-Mixup, Dual-SCNN-Mixup, CapsNet, 3D-CNN, SSUN, SSRN, DCFSL, and MDL4OW represent a diverse range of methodologies and architectures, each known for its competitive performance across various datasets. Their selection allows for a comprehensive comparison, considering the strengths and weaknesses of different approaches. Furthermore, these models have been extensively evaluated and benchmarked in prior related research, providing well-documented performance metrics for comparison. Their relevance to the research context lies in their ability to address specific challenges in HSIC, such as spatial-spectral fusion or attention mechanisms. By comparing them against the proposed 3D-CAE-SiamAtt model, insights can be gained into its effectiveness and novelty relative to established approaches.

Firstly the comparison is done in terms of OA, AA, and KC by detailing the experimental findings in Table 5.3, Table 5.4 and Table 5.5, for the IP, PU and SA datasets, respectively, where the best performances are highlighted in bold. The tables demonstrate

that the proposed 3D-CAE-SiamAtt model achieves an overall accuracy of 91.30% on the IP dataset, 87.24% on the PU dataset, and 93.60% on the SA dataset.

Table 5.3: Results from IP to IP dataset

	2D-CNN [321]	3D-CNN [321]	Hybrid SN [321]	3D-CNN [319]in[322]	SSUN [261]in[322]	SSRN [136]in[322]	DCFSL [320]in[322]	3DAES [58]in[322]	3D-CAE-SiamAtt
C1	77.67	71.42	91.78	98.55	**100**	89.36	**100**	97.83	**100**
C2	86.51	78.32	84.36	65.49	65.86	**87.66**	74.09	62.76	32.77
C3	84	70.23	82.41	79.63	81.12	**84.47**	81.23	69.83	49.51
C4	68.98	72.9	94.39	93.86	**96.73**	79.39	96.31	87.2	51.89
C5	**95.09**	87.18	94.14	89.33	90.43	95.05	91.84	87.7	68.73
C6	**97.95**	92.94	93.3	93.87	90.15	96.94	97	92.94	90
C7	71.15	86.78	88.58	99.55	98.57	63.68	**100**	**100**	**100**
C8	98.46	95.53	98.11	99.42	98.62	**99.6**	98.34	98.77	96.44
C9	90	61.11	71.69	98.75	**100**	43.56	**100**	**100**	**100**
C10	87.13	75.17	**89.2**	78.1	73.16	76.59	80.15	72.23	59.05
C11	87.77	77.9	**94.1**	63.34	71.43	89.97	72.31	62.15	41.09
C12	75.67	71.34	**86.16**	71.76	80.8	73.46	78.05	72.06	46.88
C13	97.42	96.43	96.49	98.2	99.08	90.6	99.68	**100**	99.51
C14	**98.32**	92.6	90.19	83.07	88.29	96.97	93.8	91.64	87.27
C15	85.79	80.63	85.44	90.62	**96.31**	83.59	94.29	82.44	90.15
C16	97.84	97.22	81.98	98.82	99.86	82.68	99.59	**100**	**100**
OA	89.09	81.3	89.57	77.13	80.05	87.09	83.1	75.52	**91.3**
AA	87.48	81.73	88.9	87.65	89.4	83.35	**91.04**	86.1	75.83
KC	**87.58**	78.62	88.12	74.25	77.44	85.35	80.87	72.32	82.6

Table 5.4: Results from PU to PU dataset

	Dual-SCNN-AT-Mixup [49]	Dual-SCNN-Mixup [49]	CapsNet [51]in[49]	3D-CNN [319]in[322]	SSUN [261]in[322]	SSRN [136]in[322]	DCFSL [320]	MDLOW [309]in[58]	3DAES [58]	3D-CAE-SiamAtt
C1	80.76	81.33	80.18	83.28	84.7	**91.79**	82.20	72.77	78.49	43.73
C2	**89.96**	89.9	82.26	85.96	81.68	84.51	87.74	76.24	76.8	73.01
C3	69.51	63.56	64.11	81.89	73.12	**85.23**	67.46	70.39	64.96	35.58
C4	96.24	96.93	89.8	90.57	**97.72**	94.1	93.16	93.21	92.91	89.84
C5	99.49	98.96	97.73	99.92	99.18	**99.99**	99.49	99.41	95.27	95.98
C6	77.6	79.27	81.85	84.93	83.37	**97.67**	77.32	71.44	82.19	85.06
C7	77.23	79.35	78.83	94.9	89.76	**97.72**	81.18	88.03	84.19	75.93
C8	80.6	74.57	81.27	64.4	**92.56**	91.77	66.73	63.8	79.13	47.82
C9	97.59	99.58	93.64	60.06	99.61	99.76	**98.66**	97.47	92.76	92.5
OA	85.81	85.41	81.9	83.84	85.2	**89.75**	83.65	81.42	82.97	**87.24**
AA	85.44	84.56	83.28	82.88	89.08	**93.62**	83.77	76.55	79.63	71.05
KC	81.36	81.01	76.65	79.02	80.92	**86.84**	78.70	70	74	74.49

In other words, the 3D-CAE-SiamAtt achieves the highest overall accuracy for IP, second highest for PU and third highest for SA dataset. Other models that achieved higher accuracy included SSRN for PU, and SSRN, 3DAES and 3DLSN for SA. The performance of SSRN is better due to the supervised nature of this chapter which trains the network using labeled data. Although semi-supervised models have the advantage of leveraging a larger dataset for training, the absence of explicit labels for the majority of the data in-

troduces additional uncertainty and can make the learning process more challenging. As a result, the classification accuracy of semi-supervised models may not match the performance of supervised models. On the other hand, 3DAES the semi-supervised model built using Siamese achieved better accuracy for SA, however, it was marginally lower for the other two datasets. It is to be noted that the proposed model achieves better or comparable results because of the involved three-dimensional convolution layers and attention layers which fit the nature of HSI considering both spatial and spectral information. Further. the involved attention layers discriminate the significant bands and ignore the noisy bands. Furthermore, it is to be noted that the proposed network took 111 ms/step for IP, PU and SA while 103 ms/step for PC with a total of 100 epochs having 50 steps each.

Table 5.5: Results from SA to SA dataset

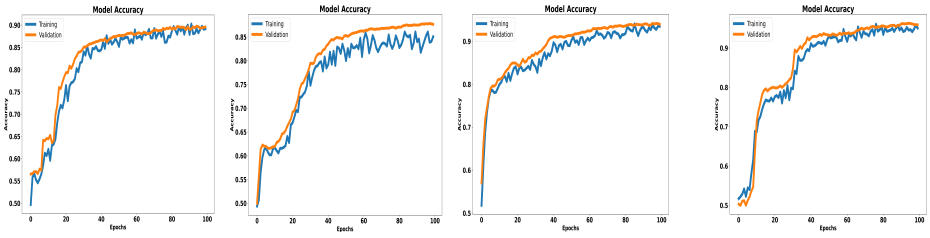
	SSFTT [323]	ASPCNet [323]	3DCNN [319]in[322]	SSUN [261]in[322]	SSRN [136]in[322]	3DLSN [324]	3DCAE [308]in[58]	MDL4OW [309]in[58]	3DAES [58]	3D-CAE-SiamAtt
C1	99.49	95.84	99.41	97.05	**100**	99.87	99.28	86.12	99.97	98.8
C2	99.84	**99.93**	99.46	95.95	99.64	99.46	59.04	91.97	99.18	97.74
C3	99.83	98.68	99.61	98.61	**99.89**	97.93	66.54	80.34	95.15	64.67
C4	95.84	91.88	99.17	**99.88**	97.81	99.75	98.65	99.28	96.47	98.56
C5	96.38	97.46	98	**98.54**	98.17	95.23	81.94	95.06	91.18	96.26
C6	98.28	98.64	99.83	99.72	**99.94**	99.75	98.52	99	99.17	93.81
C7	99.46	99.59	99.2	97.13	99.81	99.27	97.31	98.16	**99.9**	97.76
C8	75.6	86.07	78.79	80.54	88.15	75.05	68.11	47.38	75.66	**95.9**
C9	**99.97**	97.15	99.4	99.56	99.56	99.25	95.06	97.43	98.96	98.58
C10	89.01	93.62	92.89	95.6	**96.34**	92.92	92.43	87.36	89.74	84.28
C11	**99.91**	96.23	98.85	98.38	97.3	98.68	72.26	96.16	99.54	92.97
C12	99.26	96.28	99.75	99.77	99.04	99.86	72.16	99.76	99.03	**100**
C13	97.98	89.31	99.64	99.38	98.72	99.98	99.78	99.75	99.76	**100**
C14	96.13	94.17	**99.36**	98.6	95.09	98.89	89.93	98.1	97.24	96.35
C15	82.85	77.33	87.82	84.39	79.1	80.87	56.98	77.66	81.7	**96.21**
C16	98.27	**100**	96.85	97.08	98.93	94.65	44.35	95.64	87.94	96.01
OA	91.26	91.5	93.01	92.61	93.83	**95.71**	75.58	90.57	**94.41**	**93.6**
AA	95.51	94.51	**96.75**	96.26	96.72	91.05	72.64	82.44	90.39	94.24
KC	90.3	90.55	92.24	91.79	**93.13**	90	70	81	89	87.2

Secondly, the performance of 3D-CAE-SiamAtt has been comparatively analyzed by considering the cross-domain datasets, as the proposed model runs for not just same-domain but cross-domain datasets as well, provided that the number of bands is similar. Table 5.6 lists the results achieved for cross-domain datasets, where four scenarios (a) training on IP, testing on SA; (b) training on SA, testing on IP; (c) training on PU, testing on PC; (d) training on PC, and testing on PU, have been considered. The provided experimental results validate the proposed model's potential to train and test on cross-domains. More specifically, the proposed model achieves its best OA when training is performed on IP and testing on SA. However, it can also be observed that the cross-domain results are on a slightly lower side than those achieved on the same-domain datasets. This could be because cross-domain datasets present unique challenges due to variations in distribution,

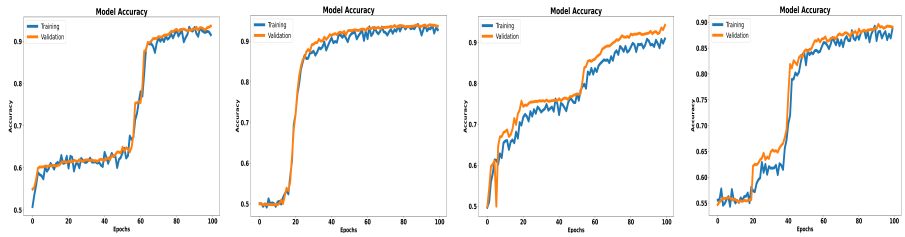
Table 5.6: Cross-Domain Datasets

	$T_r >$ **IP**; $T_e >$ **SA**	$T_r >$ **SA**; $T_e >$ **IP**	$T_r >$ **PU**; $T_e >$ **PC**	$T_r >$ **PC**; $T_e >$ **PU**
	3D-CAE-SiamAtt	**3D-CAE-SiamAtt**	**3D-CAE-SiamAtt**	**3D-CAE-SiamAtt**
C1	100	65.21	97.9	51.54
C2	99.19	26.75	98.6	67.48
C3	99.79	29.51	88.99	69.31
C4	100	24.89	97.5	77.67
C5	89.39	33.54	79.14	56.87
C6	96.33	47.8	97.9	75.4
C7	97.59	75	79.67	57.81
C8	94.21	85.56	97.8	88.94
C9	100	90	98.81	50.47
C10	50	30.76	-	-
C11	100	34.54	-	-
C12	100	21.92	-	-
C13	92.13	47.8	-	-
C14	65.7	51.69	-	-
C15	96.34	42.48	-	-
C16	34.25	73.11	-	-
OA	87.2	73.91	78.91	75.32
AA	88.43	48.78	92.92	66.17
KC	74.41	47.82	57.82	50.64

domain-specific features, and limited labelled data availability compared to same-domain datasets. However, these challenges provide valuable opportunities for advancements in the field of cross-domain classification. By addressing domain shift through techniques such as domain adaptation and leveraging feature augmentation, future models can improve the performance of models on cross-domain datasets. Moreover, developing specialized architectures and exploring innovative approaches can further enhance the classification results. These endeavours will contribute to a deeper understanding of cross-domain classification and drive progress in solving complex real-world problems spanning different domains.



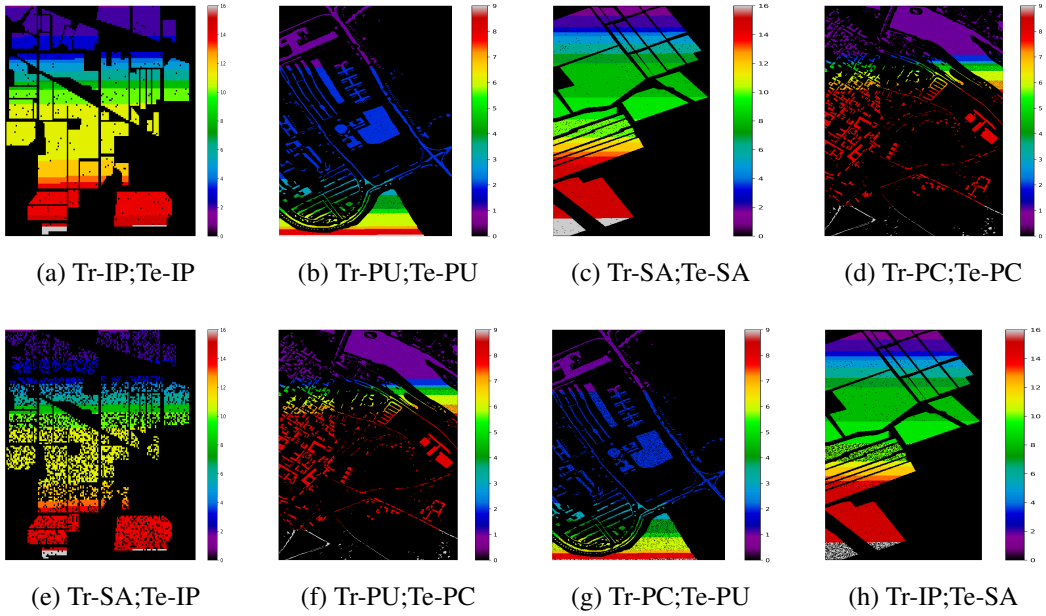
(a) Tr-IP;Te-IP (b) Tr-PU;Te-PU (c) Tr-SA;Te-SA (d) Tr-PC;Te-PC



(e) Tr-SA;Te-IP (f) Tr-PU;Te-PC (g) Tr-PC;Te-PU (h) Tr-IP;Te-SA

Figure 5.11: Model Accuracy Graphs achieved

To further qualitatively validate the performance of the model, the training accuracy graphs for the same domain and cross-domain datasets are demonstrated in Figure 5.11 which showcases the classification maps when experimented on the same or cross-domain datasets. The presented classification maps show the similarity of obtained maps to the ground truth images which also means that the ability of the model to work cross-domain and produce results is excellent. Further, the 3D-CAE-SiamAtt produces excellent classification maps with minimum noise when $T_r >$ IP; $T_e >$ IP, $T_r >$ PU; $T_e >$ PU and $T_r >$ SA; $T_e >$ SA. Moreover, the accuracy graphs to depict the performance of the proposed network on cross-domain data are presented in Figure 5.11 *(Bottom Row)*. On one hand, graphs trained for the same domain data produced smooth learning curves with no overfitting, there was slight overfitting observed while training PC and testing PU. All other datasets have smooth graphs validating the performance of the proposed model.



(a) Tr-IP;Te-IP (b) Tr-PU;Te-PU (c) Tr-SA;Te-SA (d) Tr-PC;Te-PC

(e) Tr-SA;Te-IP (f) Tr-PU;Te-PC (g) Tr-PC;Te-PU (h) Tr-IP;Te-SA

Figure 5.12: Classification Maps for Same-Domain and Cross-Domain Datasets

5.5 Summary

This chapter presented a novel semi-supervised deep learning network for HSI classification, which incorporates an autoencoder along with Siamese and attention layers. The convolutional autoencoder is trained in an unsupervised manner to refine the representation, solving the problem of limited samples. The handcrafted Siamese network enhances the feature separability between samples, by increasing the distance in samples belonging to different classes and minimizing the distance in samples belonging to the same class. The attention layers aided in extracting the informative spectral-spatial features by suppressing the noisy or less important ones. Four benchmark HSI datasets were used to validate the tenacity of the proposed 3D-CAE-SIAM-ATT. The proposed model garnered a classification accuracy of 91.3% and 93.6% for IP and SA, respectively, with a highest of 87.27% when tested on cross-domain data. The experiments also validate the domain adaptation capability of 3D-CAE-SiamAtt. Thus, the potential applications of the proposed HSI classification model can extend beyond its current domains of remote sensing, environmental monitoring, agriculture, and mineral exploration.

CHAPTER 6

UNIFYING AUTOENCODER-ENHANCED GANS WITH CNNS AND ZERO-SHOT LEARNING FOR HSIC

HSIC grapples with the twin challenges of high dimensionality and limited labelled data. These limitations hinder the development of generalizable classification models that can perform well across diverse datasets. To overcome these limitations, this chapter proposes a novel framework that synergizes autoencoders, generative adversarial networks and zero-shot learning. Leveraging the strengths of semi-supervised learning, this approach offers significant improvements in feature extraction, data augmentation, and classification accuracy by harnessing the power of generative adversarial networks built upon the crux of autoencoders. It further pushes the boundaries beyond traditional methods by enabling zero-shot learning, allowing the model to classify unseen data from classes not present in the training set. Additionally, the framework incorporates text embeddings to enrich feature representation for improved performance. This multimodal classification approach empowers the way for robust training and testing on cross-sensor datasets, even handling data with diverse spectra. It demonstrates remarkable accuracy across various domains, achieving a peak performance of 92.35% for cross-domain data and 91.83% for same-domain data, marking a significant leap forward in the generalizability of semi-supervised classification models.

6.1 Introduction

Deep CNNs have demonstrated their prowess in diverse computer vision domains, but their direct application to hyperspectral data is impeded by the unique attributes of spectral information [325]. The incorporation of numerous spectral bands and the paucity of labelled samples necessitate a specialized approach [326]. As a response to these challenges, researchers have explored innovative strategies that merge the capabilities of unsupervised

and supervised learning paradigms [327, 56]. This symbiotic fusion has yielded novel insights into feature extraction, data augmentation, and classification, ushering in a new era of enhanced hyperspectral image analysis. Moreover, in the realm of HSI classification, handcrafted feature-based methods have traditionally taken precedence over deep learning approaches [56]. These methods involve partitioning into distinct feature extraction and classifier modules, facilitating tailored customization to specific needs [328, 329]. This customized approach significantly contributes to enhancing classification accuracy.

In the context of HSIC, this chapter seeks to harness the synergistic potential of GANs and autoencoders, combined with the robust capabilities of CNNs. This integrated approach is designed to address the challenges posed by limited labelled samples. By uniting these frameworks, the model aims not only to enhance feature extraction and amplify the impact of data augmentation but also to incorporate the context provided by text embeddings. This comprehensive strategy strives to significantly enhance classification performance. Through the incorporation of cutting-edge deep learning techniques, the utilization of CNNs for improved feature extraction, and the exploitation of the synergy between unsupervised and supervised learning paradigms, this research endeavours to advance hyperspectral image analysis, ultimately achieving heightened accuracy and efficacy. The novelty of the proposed model lies in its synergistic integration and fine-tuning of various components, working towards the challenge of limited labelled samples. It facilitates cross-domain and cross-sensor classification without relying on specific spectral characteristics of training or testing data, thereby allowing testing on unseen samples. By combining GAN constructed using autoencoders and applying zero-shot learning, the model achieves enhanced feature extraction and classification accuracy, addressing critical challenges in hyperspectral data analysis.

The contributions of the proposed model are:

- **Enhanced Feature Extraction and Improved Latent Space and Distribution Modelling:** Our approach combines GAN and autoencoders to revolutionize hyperspectral image analysis. By coupling autoencoder-based feature extraction with GAN-generated synthetic data, we enrich the feature space and enhance data generalization, especially in scenarios with limited labelled samples. Additionally, leveraging GANs fine-tunes the autoencoder's latent space, aligning it with the true data distribution and boosting the significance of extracted features. This synergy elevates our model's capability for feature extraction and classification accuracy.

- **Cross-Domain Learning with Text Embeddings and Zero-Shot Learning:** Extending beyond conventional methods, the model incorporates text embeddings derived from Term Frequency - Inverse Document Frequency (TF-IDF) representations of associated text data. By fusing these embeddings with fine tunes features, the

model gains a deeper understanding of data, facilitating cross-domain learning and enhancing classification accuracy. Moreover, the inclusion of text embeddings enables zero-shot learning, allowing the model to classify samples from classes not seen during training, making it adaptable to evolving class definitions.

- **Addressing Data Scarcity through Unsupervised Learning:** Our model not only excels in supervised learning but also addresses data scarcity through unsupervised learning. The GANs ability to generate synthetic data enhances the performance of the classifier by providing diverse samples for training. This makes our model robust in scenarios where collecting labelled data is challenging.

6.2 Related Models

In this section, we briefly review some of the related works that address the problem of limited samples with the help of synthetic data. One such work in this category was discussed by Yu et al. [330]. In the discussed work, Yu et al. tackle the scarcity of labelled data by proposing an unsupervised model combining GANs and AE. In their architecture, the autoencoder captures latent representations of input data, which are then refined by the GAN to produce synthetic data, enhancing the encoder's precision. This symbiotic relationship between GANs and autoencoders enrich latent data representations and improves the discriminator's ability to represent the underlying data distribution [330]. While Yu et al.'s approach primarily focuses on feature extraction, it lays the groundwork for potential expansion into classification tasks. In contrast, Cao et al. [331] presented an unsupervised architecture based on contrastive learning, outperforming existing supervised feature extraction methods. This recognition of the effectiveness of unsupervised strategies in feature extraction contributes to the ongoing evolution of hyperspectral image classification, indicating a promising direction for future research.

Transitioning from this innovative advancement in HSIC, a parallel trajectory emerges from the domain of natural language processing—text embeddings, like Term Frequency-Inverse Document Frequency (TF-IDF) vectors, have garnered widespread attention in cross-scene HSI. Gao et al. [56] suggest this crossover infusion not only to enrich the representation of hyperspectral data but also to hold the promise of bolstering generalization and classification accuracy [56]. In the same vein, an intriguing approach by Pan et al. [233] is an incursion of text embeddings into hyperspectral image analysis, where they introduce semantic dimensions to the feature space. This noteworthy work seamlessly intersects this narrative with their innovative three-phase scheme. In this endeavour, the architecture embraces a multi-faceted approach, encompassing feature embedding, feature mapping, and label reasoning. The bedrock of their methodology revolves around harnessing the potential

Table 6.1: Summary of Related Works

Ref	Model	DA	OA	Learn	SA	Key Findings	Limitations
[56]	CNN + LSTM	IP-PU IP-KSC PU-KSC	90.42 92.40 83.76	S-Su	50	Proposes cross-domain model using word embeddings and CNN.	Complexity in distinguishing similar categories semantically.
[332]	3D-CAE-Siam-ATT	IP-SA SA-IP PU-PC PC-PU	87.02 73.91 78.91 75.32	S-Su	N/A	Few-shot and Cross-Domain Learning	Cross domain having constraint of same spectra
[53]	CNN	IP SA PU	64.19 85.24 67.85	Su	3-15	Handcrafted CNN architecture, Deals with limited samples, Reduces overfitting.	Requires large-labelled training data, Computational complexity.
[333]	Residual Deep PCA	IP PU SA	81.22 86.20 90.71	Un	N/A	Residual structure retains information between layers, Adds nonlinearity.	Requires tuning parameters, Linear feature extraction limits representation.
[39]	GAN	IP SA PU	64.19	S-Su	200	GAN enables semi-supervised learning, The discriminator is effective as a classifier.	GAN training can be unstable, Choosing architecture requires tuning.
[334]	GCN	RPC-RPU RPU-RPC	84.06 89.96	S-Su	N/A	Feature alignment for cross-domain data	Accuracy could be increased
[330]	GAN	YaleB ORL COIL MNIST Umist	N/A	Un	N/A	GANs improve subspace clustering, Dual GAN architecture is effective.	Requires tuning of GAN architectures and parameters.
[233]	ZSL	IP-SA SA-IP PU-PC PC-PU	N/A	S-Su	3-5	Introduces zero-shot learning for HSI classification across datasets.	Performance limited by dataset differences and lack of attributes.
[331]	AE	IP SA PU	97.08 99.60 99.46	Un	N/A	Combines autoencoder features and contrastive learning.	Requires sufficient unlabeled training data.

DA – Dataset; OA – Overall Accuracy; SA- Samples; CNN - Convolutional Neural network, GAN - Generative Adversarial Network; PCA - Principal Component Analysis; ZSL - Zero-Shot Learning; LSTM - Long Short Term Memory, 3D-CAE-Siam-ATT : Three Dimensional Convolutional Autoencoder Siamese Attention Network; GCN - Graph Convolutional Network, Su - Supervised, S-Su - Semi-Supervised; Un-Unsupervised; IP-Indian Pines; PU- Pavia University; PC-Pavia Center; SA-Salinas

of hyperspectral datasets acquired from the same sensor but employed in distinct contexts. The classifier's learning journey commences with one dataset and culminates with the evaluation of another. Inspired by the latest strides in zero-shot learning, Pan et al. [233] and Gao et al. [56] intricately weave label semantic representation into their framework. This pivotal addition establishes critical associations between seen categories in the training set and unseen categories in the testing set, effectively expanding the model's horizons to address evolving and expanding class definitions [53]. Table 6.1 shows the current trends in literature.

Amid these advancements, the proposed model's novelty lies an intriguing intersection: the convergence of GAN, AE, Deep CNNs, text embeddings, and zero-shot learning within a unified model. This innovative framework is based on associating the power of GANs and AE for elevated feature representation through unsupervised learning, while concurrently leveraging text embeddings for semantic augmentation and zero-shot learning, resulting in an adaptable tapestry suitable for diverse hyperspectral datasets. With the promise of heightened classification performance, our approach stands as a pioneering endeavour poised to redefine the frontiers of hyperspectral image analysis. With the combination of these aspects, our proposed model significantly advances hyperspectral image analysis. It empowers the field with enhanced feature representations, improved data augmentation, cross-domain learning, zero-shot classification, and the ability to thrive under data scarcity.

As this narrative unfolds, subsequent sections delve deeper into the intricacies of our proposed model. The architecture's blueprint, the nuances of its training process, and the illumination cast by experimental results all undergo scrutiny in Section 6.3. Furthermore, the impact of our proposed approach on traditional domain boundaries will be evident, as presented in Section 6.4. Finally, the chapter is summarized in Section 6.5.

6.3 Proposed Methodology

In this chapter, we introduce an innovative approach for HSI classification, specifically tailored to address the challenges arising from limited labelled samples in HSI datasets. Our approach involves the synergistic integration of adversarial networks, empowered by autoencoders and convolutional neural networks. The proposed network is based on two fundamental aspects: firstly, harnessing the capabilities of GANs and AEs to revolutionize feature representations derived from hyperspectral data through unsupervised learning, and secondly, incorporating CNNs to capitalize on labelled samples available. In the following subsections, we provide an overview of the proposed model in Subsection 6.3.1, delve into the design of the autoencoder-embedded discriminator in Subsection 6.3.2, and discuss the design of the generator in Subsection 6.3.3. The subsequent subsection 6.3.5 explores

supervised training using CNNs.

6.3.1 GAN-AE Embedded Convolutional ZSL based HSI Classifier (CONVZ-AEG-TF-HIC)

In this cutting-edge research endeavour, we propose an approach for HSI classification that addresses the limitation of limited labelled samples in HSI datasets. The proposed method involves the seamless fusion of an autoencoder-like model as the discriminator and a proficient generator that harnesses the decoder for data upsampling. We aim to revolutionize feature representations derived from hyperspectral data by harnessing the power of GANs and autoencoders for unsupervised learning. Figure 6.1 visually depicts the working of the proposed model.

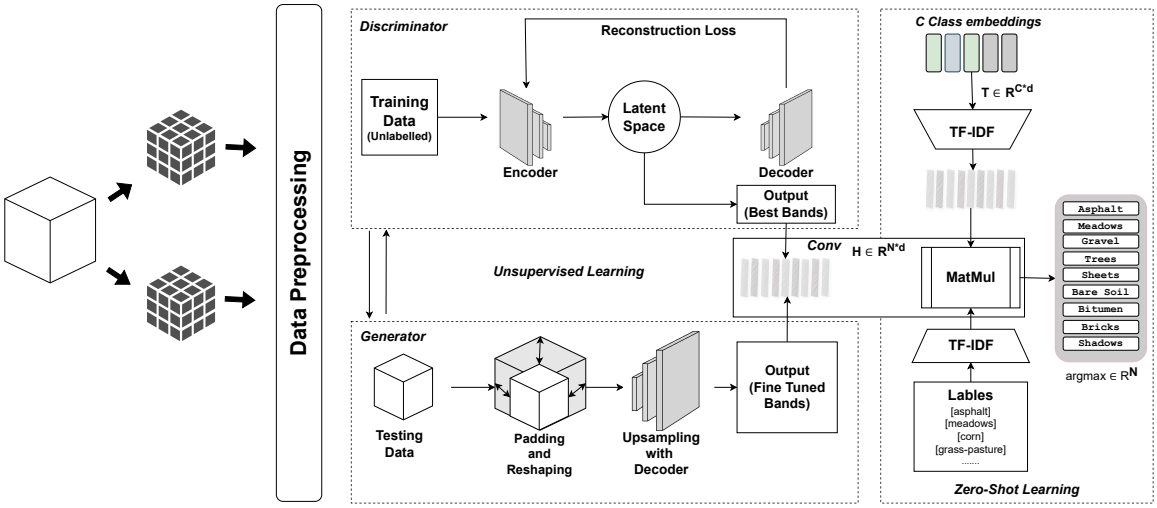


Figure 6.1: Overall working of CONVZ-AEG-TF-HIC

In this section, we propose a novel approach for HSI classification that tackles the challenge of limited labelled samples. Our method seamlessly merges an autoencoder-like model as a discriminator with a powerful generator that leverages its decoder for data augmentation. By harnessing the synergy of GANs and autoencoders, we aim to revolutionize the feature representations derived from HSI data through unsupervised learning. The generator plays a central role, meticulously upsampling the unlabeled testing data to create synthetic samples resembling real-world data. This augmented data effectively enriches feature representations during the unsupervised learning process. The proposed autoencoder-embedded GAN demonstrates remarkable potential in generating realistic data and enhancing feature representations.

Furthermore, we leverage any available labelled samples to combine the extracted features with text embeddings, forming a powerful hybrid model. This enables effective utilization of both labelled and unlabeled data, mitigating the limitations imposed by limited

labelled information. This amalgamation of unsupervised and supervised learning results in a semi-supervised network with superior performance in HSI classification tasks.

Our comprehensive approach, enhanced by a meticulously designed convolutional neural network, effectively generates realistic data, improves feature representations, and addresses the challenges posed by limited labelled data in HSI classification. Overall, the proposed model holds promise for advancing the field of HSI classification by showcasing the benefits of integrating unsupervised and supervised learning for enhanced feature representation and robust classification performance.

6.3.2 Autoencoder Embedded Discriminator

The composition of the handcrafted discriminator is visually depicted in Figure 6.2.

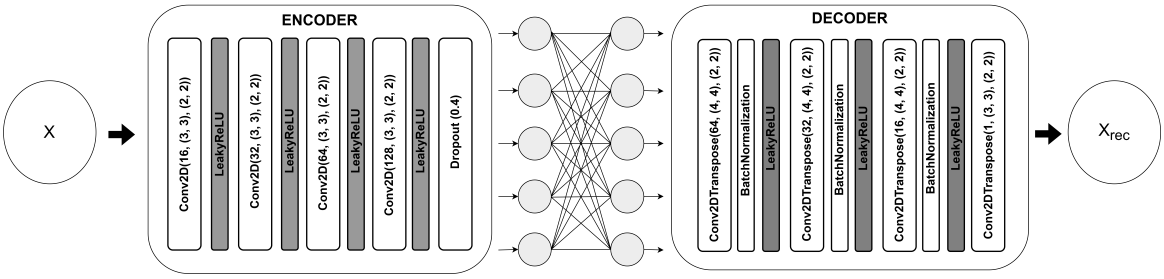


Figure 6.2: Composition of the Discriminator

To initiate the GAN training process, the data X undergoes reshaping into a 4D array with dimensions $X(N, H, W, C)$, where N denotes the number of samples, height, and width represent the dimensions of each sample and C denotes the number of spectral bands involved, depicted in Equation 6.1.

$$\text{Reshaped}_{X_{train_{unsup}}} = X_{train_{unsup}}.\text{reshape}(N, H, W, C) \tag{6.1}$$

The crux of the discriminator model is handcrafted by building an autoencoder. By design, an autoencoder comprises two integral components: an encoder and a decoder. Operating in tandem, the encoder, endeavours to compress the input data X into a compact, lower-dimensional representation, shown in Equation 6.2.

$$Z = E(X, \theta_E, \lambda_1, \lambda_2) + \gamma - \delta \tag{6.2}$$

The equation conveys below meaning :

- **Z**: This represents the latent representation of the hyperspectral input X. In the context of an autoencoder, Z is the lower-dimensional encoding produced by the encoder part of the autoencoder.

- **E**: This denotes the encoder function. The encoder is a neural network that compresses the input data X into a lower-dimensional latent space.
- **X**: This is the hyperspectral input data. It's the original high-dimensional data that we want to encode into a lower-dimensional space.
- θ_E: These are the learnable parameters of the encoder network E. They include weights and biases that are adjusted during the training process.
- λ_1 and λ_2: These are hyperparameters that influence the encoder function E. They might be regularization parameters or other factors that control the behavior of the encoder.
- γ: This is a hyperparameter that adds an intricate regularization to the encoding process. It helps in introducing non-linearity into the latent space.
- δ: Another hyperparameter that introduces subtle shifts in the latent space, further contributing to the non-linearity.

So, the equation can be interpreted as follows:

- The encoder function E takes the input data X and encodes it into a latent representation Z using the parameters θ_E and influenced by the hyperparameters λ_1 and λ_2. After encoding, the latent representation is further adjusted by adding γ and subtracting δ, which introduces additional non-linearity and shifts to the latent space representation.

Conversely, the decoder's objective is to reconstruct the original input, presented as $X_{hat} = D(Z)$, from the compressed latent representation Z, ensuring a seamless reconstruction process. To capture the essence of the original data distribution, the discriminator endeavours to proficiently reconstruct the input data through the intricate interplay of its encoder and decoder components having reconstruction loss presented in Equation 6.3.

$$MSE_{loss} = \frac{1}{N}\sum_{i=1}^{N}(X - X_{hat})^2 \tag{6.3}$$

The discriminator model is trained using the training data $X_{train_{unsup}}$, If the padded data for the test dataset does not exist, the code generates padded data. For each row in the test dataset, the data is padded to match the dimension of the training dataset. Finally, a new sequential model named discriminator is created. The previously trained discriminator model is added to this new sequential model. A flattened layer and a dense layer with sigmoid activation are added to complete the discriminator architecture.

6.3.3 Generator for Synthetic Data

A latent noise vector is generated from a uniform distribution tailored to the training data's dimensions, forming the basis for our specialized generator model. Guided by a specific objective, the generator undergoes epoch-based training to create synthetic data that mirrors the genuine data distribution. During each epoch, random noise acts as input for the generator, producing synthetic images assessed by a discriminator—a pivotal component of our GAN framework.

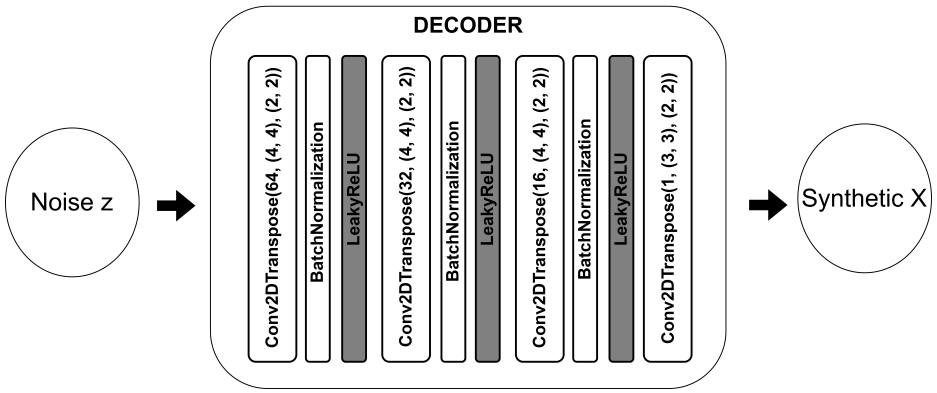


Figure 6.3: Composition of the Generator

Generator and discriminator losses, computed using binary cross-entropy functions as detailed in section 6.3.4, drive gradients that update the generator's weights through an optimizer. This iterative process enhances the generator's ability to create synthetic data with heightened realism and a coherent distribution. We meticulously track and archive generator and discriminator losses to document training progress. Furthermore, these loss profiles are stored for future analysis, complementing our comprehensive approach. The generator $G(.)$ is defined by employing the decoder of the autoencoder illustrated in Figure 6.3, enabling data upsampling. This mechanism enriches the testing data X_{test} by generating samples with increased dimensions, thus enhancing the testing dataset for subsequent learning stages. The generation of synthetic data Y is $Y = G(Z)$.

$$Y = G(Z) \tag{6.4}$$

Create Padded DataFrame

Data preparation is crucial for effective hyperspectral image classification. It involves creating a padded data frame to ensure uniform sample dimensions. A specialized function pads the features to match the training dataset dimensions. The padded rows, along with their target labels, are appended to the list, creating a complete padded dataset stored in a

Algorithm 5 Padded DataFrame Generation

Inputs:
df_{test} : Dataframe containing the test dataset
df_{train} : Dataframe containing the train dataset
Outputs: $padded_{df}$: Dataframe with padding applied
Procedure:

```
Create an empty list named padded_rows
if CSV file does not exist in Datafiles directory then
    for i from 0 to df_test.shape[0] - 1 do
        row ← df_test.iloc[i, : −1].values
        padded_row ← getdata.paddata(row, df_train.shape[1])
        padded_row ← padded_row + [df_test.iloc[i, −1]]
        padded_rows.append(padded_row)
    end for
    padded_df ← Create DataFrame from padded_rows
    Save padded_df in "DataFiles" directory
else
    padded_df ← Read padded DataFrame from "padded_df_
end if
Return padded_df as the final output.
```

Return: $padded_{df}$ as the output

data frame and saved for future use. This approach, formulated in Algorithm 5 addresses challenges related to varied sample dimensions, enhancing the robustness and reliability of the classification model.

6.3.4 GAN Training and Fine-Tuning

This step involves iterative fine-tuning of the generator and discriminator models. The real samples are denoted by $D(X)$ and synthetic samples are denoted by $D(Y)$, where X denotes the real data or ground truth that the discriminator evaluates. The generator loss function denoted as L_{gen}, operates on the synthetic output. It calculates the binary cross-entropy loss between ones indicating real data and the generated output. This loss depicted in equation 6.5 guides the generator to produce data that closely emulates real samples, enhancing the realism of the generated data.

$$L_{gen} = BCE(Ones, D(Y)) \tag{6.5}$$

The discriminator loss function depicted in equation 6.6, defined as L_{disc} leverages two components. The first component is Real loss. It measures the dissimilarity between the predicted probabilities and the true labels. This component calculates the binary cross-entropy loss between the vector of ones representing real data and the output produced for real data. This component guides the discriminator to accurately classify real samples, promoting precise discrimination. $D_{X_{train_{unsup}}}$ represents the output of discriminator $D(.)$ when $X_{train_{unsup}}$ is given as the input. The discriminator's task is to classify whether the input data is real or fake. This is depicted in Equation 6.6.

$$RealLoss = BCE(Ones, D_{X_{train_{unsup}}}) \tag{6.6}$$

Fake loss, denoted in equation 6.7 is this component that calculates the binary cross-entropy loss between zeroes indicating generated data and the output produced for generated data. It aids the discriminator in effectively distinguishing between real and fake samples.

$$FakeLoss = BCE(Zeros, D(Y)) \tag{6.7}$$

The total loss, depicted in equation 6.8 is the sum of real loss and fake loss, encompassing the overall loss for the discriminator during training.

$$L_{disc} = BCE(Ones, D_{X_{train_{unsup}}}) + BCE(Zeros, D(Y)) \tag{6.8}$$

Engaging in unsupervised learning, the discriminator gains valuable insights into the underlying distribution of the input data, acquiring informative and robust feature representations. By reconstructing the data, it effectively captures salient patterns and structures in hyperspectral data, iteratively refining its encoder and decoder during training. The combination of unsupervised learning with an autoencoder-like architecture remarkably enhances feature representations, particularly in scenarios with limited labelled samples. Our approach showcases the potential of autoencoder-embedded GANs for unsupervised feature learning, paving the way for more powerful feature extraction methods in hyperspectral image analysis. The binary cross-entropy loss function serves as the foundational loss for both the generator and discriminator, guiding the models throughout the adversarial training process. Coupled with Adam optimizers, these loss functions enable the GAN to generate high-quality data and refine feature representations, demonstrating great potential for data synthesis and representation enhancement in various applications.

6.3.5 Supervised Training using CONV

The fine-tuned hyperspectral data X_{train} obtained from the handcrafted GAN and the TF-IDF text embeddings are inputted into a custom-built CNN, as shown in Figure 6.4. These text embeddings, derived from class labels and descriptions, enable zero-shot learning, allowing the model to classify unseen classes. These TF-IDF vectors not only encode the textual descriptions but also serve as semantic embeddings for the class labels. During training, the CNN is augmented to incorporate both spectral and semantic information simultaneously, leading to richer features and improved classification performance, especially for unseen classes.

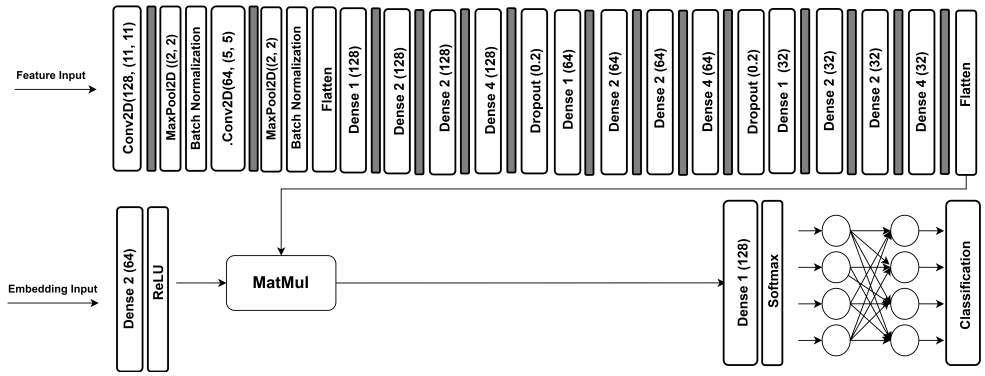


Figure 6.4: Convolutional Layers

CONV Model

The proposed model incorporates two inputs: fine-tuned hyperspectral data $X_{\text{train}}(N, H, W, C)$ and TF-IDF vectors (N_{Des}, V), capturing their interactions. Here N_{Des} represents several samples in the corpus with descriptions, and V represents the size of the vocabulary or the number of unique terms. The hyperspectral data undergoes convolutional and dense layers to extract relevant features, while TF-IDF vectors are dimensionality-reduced through dense layers. The reduced TF-IDF vectors and hyperspectral features are combined using matrix multiplication, as shown by $CombinedFeatures = \text{TF-IDF}@X_{\text{train}}$, where Combined Features are depicted as (N_{Des}, c), having c extracted features enabling the model to learn their relationship. Additional dense layers process the combined features, and a Softmax activation function, in equation 6.9, yields class probabilities P_{ij} for each sample. Here, P_{ij} represents the probability of the i-th sample belonging to the j-th class, and $\text{CombinedFeatures}_{ij}$ represents the combined feature value for the i-th sample and j-th feature. C is the number of classes.

$$P_{ij} = \frac{e^{CombinedFeatures_{ij}}}{\sum_{k=1}^{c} e^{CombinedFeatures_{ik}}} \tag{6.9}$$

$$L = -\frac{1}{N_{\text{des}}} \sum_{i=1}^{N_{\text{des}}} \sum_{j=1}^{C} \text{des['label']}i \cdot \log(Pij) \tag{6.10}$$

After applying the Softmax function, the P_{ij} matrix has shape (N_{des}, C) and represents the predicted class probabilities for each sample. The Sparse Categorical Cross-Entropy loss is used to measure the discrepancy between the true labels L presented by (des['label']) and the predicted class probabilities P_{ij}. The equation for Sparse Categorical Cross-Entropy loss is depicted in equation 6.10. Here, des['label']_i is the true label for the i-th sample, represented as a one-hot encoded vector (1 for the true class, 0 for others). Equation 6.10 computes the loss over all samples and classes, providing a measure of how well the model predicts the correct class for each sample.

6.3.6 Algorithm for CONVZ-AEG-TF-HIC

The proposed model is depicted in Algorithm 6, delineated into four core steps: unsupervised learning, test data preparation, generator model training utilizing GAN, and supervised training. The objective behind GAN for synthetic data generation stands out, enhancing classification accuracy by furnishing diverse samples, particularly in scenarios with limited labelled data. This strategy capitalizes on GANs' unique ability to effectively capture complex data distributions. Moreover, the algorithm integrates Term Frequency Inverse Document Frequency for text embeddings, enriching hyperspectral features with contextual information to enhance classification accuracy. Finally, it incorporates a fine-tuned CNN for supervised learning, processing combined features from autoencoders and TF-IDF embeddings through dense layers and softmax activation, ultimately predicting class labels and achieving the final classification outcomes.

Algorithm 6 CONVZ-AEG-TF-HIC

Step 1: Unsupervised Learning

1. Prepare the training data: $\texttt{X}_{\texttt{train}_{\texttt{unsup}}}(N, H, W, C)$

2. Create and Train the Discriminator Model in two steps: The discriminator denoted as $\texttt{D(X; } \theta_d\texttt{)}$, which takes $\texttt{X}_{\texttt{train}_{\texttt{unsup}}}$ as input and outputs the reconstructed data Xhat ($\texttt{D (X}_{\texttt{train}_{\texttt{unsup}}}, \theta_d\texttt{)}$).

Minimize the reconstruction loss between input and reconstructed images using Equation 6.3.

Step 2: Prepare the Test Data and Padded Data (`padded_df`)

1. Prepare the test data in a format similar to $\texttt{X}_{\texttt{train}_{\texttt{unsup}}}$.

2. Create a padded version of $\texttt{X}_{\texttt{test}_{\texttt{unsup}}}$ to match the dimensions of $\texttt{X}_{\texttt{train}_{\texttt{unsup}}}$ and store it as padded_df.

Step 3: Train the Generator Model

1. Generate random noise z with shape (N,H,W,C) for the Generator.

2. Define the generator model $\texttt{G(z; } \theta_g\texttt{)}$ using parameters θ_g to produce synthetic data $\texttt{Y=G(z,}\theta_g\texttt{)}$ resembling $\texttt{X}_{\texttt{train}_{\texttt{unsup}}}$.

3. Reconstruct data $\texttt{X}_{\texttt{hat}}$ using the Discriminator: $\texttt{X}_{\texttt{hat}} = \texttt{D(X}_{\texttt{train}_{\texttt{unsup}}}\texttt{ ,}\theta_d\texttt{)}$.

4. Calculate the fake output from the Discriminator: $\texttt{D(Y) = D(Y,}\theta_d\texttt{)}$.

5. Compute the generator loss $\texttt{L}_{\texttt{gen}}$ and Discriminator loss $\texttt{L}_{\texttt{disc}}$ using Equation 6.8. Update parameters and save the model weights.

Step 4: Supervised Training

1. Generate samples for training data c using saved Generator weights, denoted as $\texttt{train}_{\texttt{data}_{\texttt{sup}}}$.

2. Generate fine-tuned samples for test data $\texttt{X}_{\texttt{test}_{\texttt{unsup}}}$ using saved Generator weights, denoted as $\texttt{test}_{\texttt{data}_{\texttt{sup}}}$.

3. Load ground truth labels for training data $\texttt{train}_{\texttt{gt}}$ and test data $\texttt{test}_{\texttt{gt}}$.

4. Prepare TF-IDF text embeddings for each sample.

5. Create a Supervised Learning classifier using synthetic samples $\texttt{train}_{\texttt{data}_{\texttt{sup}}}$, $\texttt{test}_{\texttt{data}_{\texttt{sup}}}$, and text embeddings.

6. Combine features obtained by matrix multiplication of text embeddings and synthetic samples. Pass through additional dense layers and softmax using Equation 6.10.

7. Train the Classifier using Sparse Categorical Cross-Entropy loss and Adam optimizer to predict class labels. Evaluate the model on test data for final classification results. Evaluate the trained Classifier on the test data to get the final classification results.

6.4 Experimental Results and Discussion

This section encompasses the datasets selected for the model, outlining their characteristics. Additionally, it entails the outcomes derived from the proposed model, coupled with an analysis and discussion of these findings.

6.4.1 Experimental Settings

The GAN was trained using the Binary Cross Entropy loss function with KL-Divergence for optimization. The GAN employs the Adam optimizer with a learning rate set to 0.0002 and an exponential decay of 0.5 to enhance convergence. During training, the GAN operates with a batch size of 32 and undergoes 10 epochs, with each epoch consisting of 20 steps. In parallel, the discriminator model employs real and fake loss and employs the Sigmoid activation function to discern between real and generated data samples effectively. The CONV model employs flattened and fully connected layers with Softmax activation.

6.4.2 Comparison with other models

In this section, we compare the results achieved by the proposed model with state-of-the-art and customized models. The models presented in the comparative table encompass a diverse range of techniques that address the complexities of hyperspectral image classification. Each model offers unique attributes and innovations tailored to specific challenges within the domain. The GAN-based Deep Subspace clustering model, also called as *DSC DAG* introduced by Yu et al. [330] is a novel deep semi-supervised learning approach using GAN embedded with autoencoders, while the One-dimensional GAN *1DGAN*, Two-dimensional GAN (*2DGAN*), and Three-dimensional GAN (*3DGAN*) models leverage variations of Generative Adversarial Networks (GANs) to enhance data representation. The Zero-Shot learning based *Zeg-CLIP* model [335] utilizes zero-shot learning principles, and an adaptive-based learning technique *ALE* were considered. The adapted *Z-AEG-TF-HIC* model represents a variant that omits the use of CONV layers. In a similar vein, the *Z-AEG-CV-HIC* model replaces term-frequency and substitutes the count vectorizer. Furthermore, the *S-CONV-Z-AEG-TF-HIC* and *S-CONV-Z-AEG-CV-HIC* models integrate simplified convolutional architectures, maintaining the Term Frequency-Inverse Document Frequency (*TF-IDF*) and count vectorizer approaches respectively. In contrast, our proposed approach, the *CONVZ-AEG-TF-HIC* model, harmonizes a variety of techniques to deliver a comprehensive solution. This fusion adeptly combines GANs, autoencoders, and convolutional networks, resulting in enhanced feature representations, refined classification capabilities, and adaptable performance across varying domains.

SA to PU:

The outcomes showcased in Table 6.2 emphasize the superiority of our proposed model, **CONVZ-AEG-TF-HIC**, in transferring knowledge from the source domain (SA) to the target domain (PU). Notably, our model achieves an outstanding OA of 78.20, AA of 89.39 and KC of 69.75 surpassing other models included in the comparison. The potentially higher complexity and capacity of the **CONVZ-AEG-TF-HIC** model compared to other

models enable it to capture intricate patterns and relationships in the data, contributing to its superior performance in domain adaptation from IP to PU. This exceptional performance can be attributed to the model's adeptness at effectively aligning features across domains, as evidenced by its highest OA and KC values among state-of-the-art models. Conversely, models like '1D GAN' or Zeg-CLIP struggle to adapt, exhibiting the lower OA - a key parameter to assess accuracy, indicative of challenges in domain adaptation. The remarkable results of our proposed model underscore its robustness and efficacy in handling the inherent differences between source and target domains, positioning it as a promising solution for semi-supervised and cross-domain hyperspectral image classification tasks.

Table 6.2: Results from SA to PU

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	55.31	37.42	70.60	68.15	53.21	73.05	57.31	47.98	39.79	53.99	38.44	53.37
2	79.75	52.06	62.37	66.90	71.60	43.62	41.87	34.24	49.11	50.80	46.57	91.37
3	54.68	78.74	80.00	48.14	51.36	57.37	47.69	56.17	86.02	74.98	76.24	74.69
4	79.09	43.92	55.80	34.34	39.08	48.98	65.72	76.56	48.49	75.67	42.43	95.62
5	44.94	64.80	48.98	45.83	65.71	41.09	50.66	78.79	42.45	67.54	45.36	100
6	85.83	55.64	85.21	63.73	80.91	53.01	52.03	85.79	58.95	75.04	62.56	96.59
7	79.38	51.50	64.38	38.10	68.59	88.51	60.33	42.16	51.68	84.85	81.13	99.47
8	53.12	68.40	68.57	35.31	45.84	59.03	36.11	49.83	60.44	87.56	61.94	93.5
9	70.80	68.46	78.53	66.27	69.33	51.89	36.21	82.47	55.83	84.47	79.11	100
OA	72.35	77.85	75.05	70.79	70.77	73.38	66.81	75.82	59.33	75.27	76.32	**78.2**
AA	66.99	57.88	68.27	51.86	60.63	57.39	49.77	61.56	54.75	72.77	59.31	**89.39**
KC	77.71	66.82	67.11	60.05	68.11	67.46	60.35	65.08	69.30	65.86	68.46	**69.75**

Table 6.3: Results from IP to PU

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	69.66	77.33	82.18	60.81	60.24	54.72	35.85	72.63	69.08	64.94	82.37	95.64
2	84.73	74.01	68.40	77.37	42.67	40.74	43.80	43.13	36.67	59.94	72.28	99.28
3	82.33	77.36	78.25	69.82	43.66	72.01	38.77	30.12	50.79	61.01	57.19	63.74
4	78.75	84.53	86.62	81.27	68.33	45.67	67.88	66.62	72.13	74.77	82.43	95.23
5	31.38	63.88	56.51	60.33	32.55	45.15	47.12	72.75	44.93	38.16	32.35	100
6	78.51	54.96	39.85	36.03	31.55	37.81	50.63	65.62	66.95	40.12	85.42	84.01
7	72.01	71.64	46.21	41.87	73.06	63.40	78.84	50.24	55.59	57.29	42.14	96.01
8	72.26	35.31	31.09	42.29	32.80	44.97	63.33	86.68	64.41	60.04	32.01	96.33
9	49.23	78.92	66.28	47.12	80.44	39.30	85.65	45.16	35.65	40.04	31.97	100
OA	70.67	74.13	71.61	73.32	70.83	75.72	78.78	75.27	73.71	78.22	79.13	**79.74**
AA	68.76	68.66	61.71	57.44	51.70	49.31	56.87	59.22	55.13	55.15	57.57	**92.25**
KC	61.80	67.61	63.18	63.04	67.99	60.82	64.98	63.28	68.55	68.48	71.95	**71.22**

IP to PU:

Table 6.3 presents the outcomes of knowledge transfer from the source domain (IP) to the target domain (PU) across various models. Notably, the proposed **CONVZ-AEG-TF-HIC** model demonstrates remarkable performance, exhibiting an OA of 79.74, AA of 92.25, and KC of 71.22, surpassing all other models evaluated. This would have happened because of several reasons including feature alignment, ensuring that representations learned from the IP domain are well-adapted to the characteristics of the PU domain. Secondly, the model exhibits adaptability to domain shifts, enabling it to handle variations between IP and PU datasets and utilize domain-specific information for classification effectively. Moreover, the model's mechanism for representation learning is robust, allowing it to extract meaningful features from IP data and generalize effectively to PU data, thereby enhancing classification accuracy. Leveraging semi-supervised learning, the model benefits from both labelled and unlabeled data during training, exploiting the abundance of information in the IP dataset to improve performance on PU data. These reasons collectively position the **CONVZ-AEG-TF-HIC** model as a highly effective solution for knowledge transfer and domain adaptation tasks in hyperspectral image classification, showcasing its potential for real-world applications.

Table 6.4: Results from SA to IP

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	74.16	34.67	55.22	69.79	43.42	40.03	75.97	82.99	61.92	52.48	83.04	67.39
2	30.28	57.46	88.75	43.25	88.89	79.34	56.77	30.30	44.37	78.60	64.35	93.9
3	37.71	87.41	48.99	83.77	38.93	35.77	86.50	82.30	46.62	49.58	71.94	63.49
4	41.59	48.43	33.85	51.42	35.91	31.87	60.68	83.70	79.69	53.27	51.83	90.29
5	50.58	55.73	74.05	48.87	75.77	36.64	65.86	43.35	84.78	41.90	67.37	98.55
6	60.35	55.72	35.07	59.29	61.21	36.15	43.02	61.12	42.01	73.82	46.16	98.35
7	42.51	66.98	63.79	30.97	39.82	83.44	37.78	67.36	70.44	63.33	63.00	89.28
8	31.35	35.62	65.23	71.71	39.93	60.43	54.30	75.40	42.33	31.53	68.54	100
9	74.66	42.62	36.41	48.16	85.83	42.30	64.49	42.58	70.63	35.50	50.74	100
10	35.63	46.05	62.72	79.79	82.75	52.66	73.86	32.73	72.58	59.71	44.96	83.74
11	85.91	87.68	66.76	74.39	37.30	39.46	86.13	39.05	45.99	49.27	34.29	98.53
12	61.21	44.61	43.28	76.53	86.66	40.69	85.76	52.90	46.01	57.70	39.07	97.8
13	45.24	53.85	68.49	74.17	75.38	67.84	41.51	50.08	38.46	52.31	73.39	100
14	69.62	83.23	75.10	41.96	50.77	80.68	62.64	38.80	86.07	48.54	74.63	98.89
15	35.54	36.01	64.08	65.91	35.69	62.89	46.75	87.10	63.28	48.64	41.33	79.79
16	47.96	57.05	82.72	35.37	51.58	44.25	41.62	39.22	41.32	73.52	79.91	95.69
OA	86.72	85.61	80.66	87.98	86.45	90.41	74.38	79.11	83.95	86.01	92.26	**92.35**
AA	51.52	55.82	60.28	59.71	58.11	52.15	61.48	56.81	58.53	54.36	59.66	**90.98**
KC	82.74	83.78	73.76	84.16	84.08	76.78	85.33	85.52	70.25	80.50	89.28	**91.21**

SA to IP:

In Table 6.4, we present results from the source domain (SA) to the target domain (IP). The proposed **CONVZ-AEG-TF-HIC** model demonstrates remarkable performance, consistently achieving the highest values in the final column. With an OA of 92.35, an AA of 90.98, and a KC of 91.21. This performance may be attributed to the model's architecture having effective feature alignment between the domains SA and IP, facilitating superior adaptation to the target domain characteristics. Additionally, the similarities between the IP and SA datasets likely contribute to the model's strong performance, as it can leverage common patterns and structures between the two domains. Overall, the **CONVZ-AEG-TF-HIC** model emerges as a standout performer for SA to IP.

Table 6.5: Results from PU to PC

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	61.59	70.19	66.98	85.90	73.25	48.62	45.46	84.08	37.91	86.02	50.02	99.27
2	76.67	85.03	64.20	87.68	44.28	68.50	65.61	83.46	80.44	79.03	74.61	85.48
3	37.68	56.01	46.57	40.64	46.14	30.92	78.22	56.67	51.04	68.96	77.05	96.99
4	41.82	59.47	64.10	80.92	71.88	63.76	45.12	82.74	48.09	64.02	40.08	77.61
5	34.33	39.00	65.47	41.33	69.73	57.78	74.81	65.76	31.05	53.85	78.86	96.36
6	79.13	41.79	74.23	32.45	82.42	63.35	82.88	78.54	61.55	88.24	51.69	96.74
7	64.76	42.21	86.09	58.39	77.75	75.25	32.13	79.46	34.36	67.64	75.33	00.00
8	30.28	71.04	41.18	40.52	84.12	41.48	73.83	70.19	59.80	52.70	44.13	99.71
9	53.02	59.49	71.31	78.11	73.84	78.86	46.31	66.04	49.22	59.77	52.21	100
OA	46.84	61.07	63.02	66.88	68.40	48.48	60.05	74.81	52.07	75.92	76.81	**48.89**
AA	53.25	58.25	64.46	60.66	69.27	58.72	60.49	74.10	50.38	68.92	60.44	**83.57**
KC	39.84	54.85	61.25	59.91	57.80	42.14	58.80	65.87	48.39	65.93	70.81	**42.07**

PU to PC

The outcomes presented in Table 6.5 showcase the results from the source domain (PU) to the target domain (PC) through various models. The proposed model achieves comparatively lower results in this scenario, possibly suggesting some difficulty in aligning features between these two domains. Specifically, class 7 (*Shadows*) achieved the lowest accuracy due to the class imbalance problem. This class had the lowest number of samples amongst other classes and therefore failed to generalize. The inherent differences in spatial structures between PU and PC datasets may contribute to the observed performance gap. Furthermore, variations in environmental conditions such as disparities in land cover types, surface materials, and geographical characteristics between the PU and PC regions introduce additional complexities for knowledge transfer between the two domains.

6.4.3 Same-Domain Datasets

The subsections below depict the results when trained and tested on same domain datasets.

- In the IP to IP dataset (Table 6.6), the model achieves exceptional results with an OA of 91.83, AA of 93.51, and KC of 90.62. Trained and tested on identical datasets, it effectively captures domain-specific patterns, resulting in superior accuracy without the need for domain adaptation.

- In the SA to SA dataset, the model demonstrates remarkable performance as depicted in (Table 6.7), boasting an OA of 85.9, an AA of 88.01, and a KC of 84.2. However, a notable discrepancy arises in the classification of the vineyard vertical trellis class, exhibiting considerably lower accuracy compared to other categories. This discrepancy can be attributed to the intricate spectral signature of the vineyard vertical trellis, which poses a unique challenge for accurate classification. Unlike other classes, the spectral patterns of the vineyard vertical trellis may exhibit complexities that overlap with those of neighbouring categories. Consequently, the model's ability to discern subtle spectral variations among classes is hindered, resulting in misclassifications and diminished accuracy.

- On the PC to PC dataset (Table 6.8), the model exhibits slightly lower performance, with an OA of 53.57, AA of 89.07, and KC of 47.28. Despite this, its ability to transfer knowledge within the same domain remains evident, albeit with some variability in results. Despite being within the same domain, variations in spectral signatures, environmental conditions, or sensor characteristics can influence classification accuracy. Spatial variability, such as differences in soil moisture, vegetation density, or surface roughness, can influence spectral responses even within the confines of Pavia Centre's landscape. Temporal fluctuations, stemming from changes in weather patterns, seasonal transitions, or human interventions like agricultural practices, further contribute to spectral variability. Moreover, The model's architecture and complexity might affect its ability to generalize across different datasets effectively. If the model is overly complex or prone to overfitting, it may fail to capture the underlying patterns resulting in reduced performance.

- For the PU to PU dataset (Table 6.9), the model maintains high accuracy levels with an OA of 80.23, AA of 89.05, and KC of 72.18. Its proficiency in transferring domain knowledge underscores its reliability and robustness across diverse scenarios.

Table 6.6: Results from IP to IP

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	78.74	32.12	54.87	9.51	34.70	43.75	35.88	33.59	41.75	20.23	40.00	93.47
2	51.72	49.06	37.80	29.27	51.44	35.80	41.34	14.96	38.01	17.95	81.16	88.65
3	26.85	54.83	88.76	43.34	68.25	31.27	31.78	90.42	44.79	97.92	91.61	64.45
4	55.68	31.42	95.51	74.40	44.87	76.55	68.34	10.03	22.97	75.68	70.53	85.23
5	66.73	40.64	17.77	43.42	78.34	47.97	36.93	3.60	29.31	22.76	41.51	99.17
6	63.35	72.60	60.55	27.94	38.77	55.18	76.41	95.89	99.49	76.42	59.34	98.76
7	39.70	82.84	54.71	49.00	39.70	34.19	46.71	26.22	24.16	51.21	52.94	100
8	24.48	96.06	70.77	73.30	58.28	73.92	75.53	23.67	89.50	87.21	90.59	100
9	85.16	36.49	33.33	39.93	39.94	39.58	43.17	66.12	89.73	68.61	82.24	100
10	43.45	81.20	81.83	85.96	51.19	67.44	62.45	5.28	72.28	21.80	43.92	89.5
11	34.85	96.28	73.76	58.39	43.66	39.58	74.06	90.01	28.01	48.24	21.10	99.18
12	26.16	32.91	87.53	42.31	57.06	43.29	56.71	87.94	29.36	5.79	98.78	98.14
13	50.22	52.16	94.65	90.82	65.04	59.94	40.05	85.58	55.46	57.59	16.19	99.51
14	6.13	53.53	4.68	47.34	52.62	49.48	47.67	73.51	83.95	84.16	68.66	90.35
15	34.52	13.04	41.57	1.40	62.89	59.22	70.83	33.12	67.85	13.74	95.20	93
16	60.54	33.39	37.88	611.66	70.96	35.83	75.79	84.48	21.12	35.30	70.12	96.77
OA	83.10	82.84	83.02	82.97	83.08	82.90	82.98	82.95	82.92	83.18	83.06	**91.83**
AA	46.77	53.66	58.50	83.00	53.61	49.56	55.23	51.53	52.36	49.04	63.99	**93.51**
KC	80.89	80.96	80.94	81.11	81.01	80.99	81.10	80.92	81.10	81.00	80.97	**90.62**

6.4.4 Overall Analysis

Our proposed semi-supervised model exhibits competitive performance, showcasing its potential in hyperspectral data analysis. This approach effectively tackles the challenge of limited labelled samples, a significant bottleneck stemming from the time-consuming and labour-intensive nature of hyperspectral data annotation. While supervised models typically achieve higher accuracy with abundant labelled data, our model demonstrates the advantages of semi-supervised learning.

In comparison to our last published semi-supervised work [332], which is a few-shot learning model demonstrating excellent classification accuracy but constrained to work only on cross-domain datasets generated by the same sensors with identical spectra or bands, our current model can handle cross-domain data with different spectra. Leveraging semi-supervised learning, it provides a strong foundation for generalization and adaptability to unseen samples using zero-shot learning techniques with text embeddings. The model's ability to perform well on unseen data, despite lacking explicit training on them, underscores a significant strength of our approach. It reflects the growing trend towards semi-supervised and few-shot learning models in hyperspectral data analysis as researchers aim to overcome the challenges of data scarcity and annotation costs.

In summary, our findings affirm the potential of the proposed model in navigating the complexities of hyperspectral data analysis. Its semi-supervised architecture and ability to generalize to unseen samples signify promising advancements in overcoming data limita-

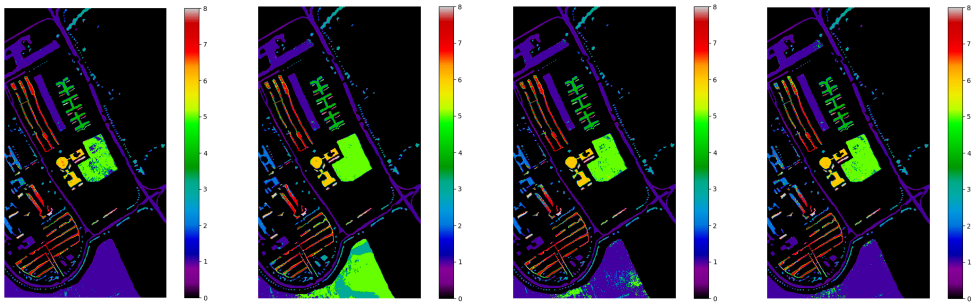
Table 6.7: Results from SA to SA

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	44.85	40.00	46.79	77.28	80.61	50.25	37.17	56.35	53.42	65.55	69.51	76.4
2	63.43	64.91	82.84	42.70	73.38	77.73	68.58	42.33	58.51	82.73	82.67	99.97
3	81.45	84.99	66.63	31.95	40.53	36.54	49.79	48.78	39.41	59.68	55.71	91.34
4	43.76	77.64	58.11	38.85	74.63	87.27	77.36	76.40	72.92	64.11	58.36	99.28
5	78.14	52.28	86.64	54.63	78.76	85.65	30.71	75.51	79.49	40.67	81.69	99.43
6	61.45	84.57	46.04	40.50	85.06	46.50	44.95	43.85	62.09	58.75	32.43	99.97
7	30.08	86.11	88.71	65.90	79.70	43.58	52.02	48.68	66.45	78.40	39.40	99.97
8	35.47	78.10	59.46	41.02	85.85	51.63	81.56	39.96	65.14	42.15	62.19	96.57
9	53.69	84.22	32.95	73.17	67.87	88.86	37.25	70.06	57.19	39.41	86.58	99.17
10	65.10	73.21	56.48	63.98	86.16	60.42	79.51	32.70	54.17	81.66	60.45	94.38
11	65.61	74.50	35.56	67.58	67.79	43.88	40.40	61.84	38.51	64.80	88.65	94.38
12	50.53	61.03	73.89	73.54	43.56	49.62	87.24	35.37	76.51	81.24	61.13	99.94
13	49.19	44.68	73.26	36.43	30.57	83.64	36.75	39.24	80.20	66.73	53.05	98.58
14	65.29	46.79	67.73	42.91	67.14	71.14	71.14	49.76	59.77	60.78	57.80	98.69
15	55.63	70.72	74.60	49.40	72.68	56.02	83.34	38.05	77.11	70.46	69.42	60.04
16	54.34	32.65	45.28	59.65	45.49	83.31	82.89	44.68	69.32	51.39	85.36	00.00
OA	67.91	79.70	72.39	85.17	81.61	84.16	81.22	77.18	81.72	84.73	83.95	**85.9**
AA	56.13	66.02	62.18	53.72	67.49	63.50	60.04	50.22	63.14	63.03	65.27	**88.01**
KC	64.56	70.44	68.04	81.77	75.08	79.69	78.32	70.13	80.78	82.59	83.49	**84.2**

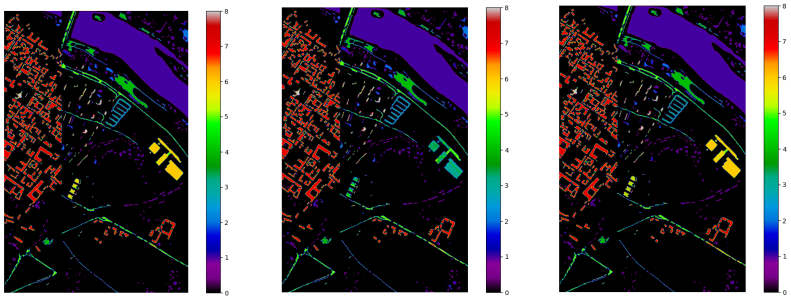
tions. The CONVZ-AEG-TF-HIC model represents a breakthrough solution addressing the challenge of limited labelled data in cross-domain knowledge transfer. By integrating semi-supervised and zero-shot learning techniques, it transcends data limitations and achieves exceptional results across diverse datasets, encompassing both same and cross-domain scenarios. This state-of-the-art performance underscores its effectiveness in navigating data scarcity and excelling in cross-domain knowledge transfer tasks.

Classification Maps

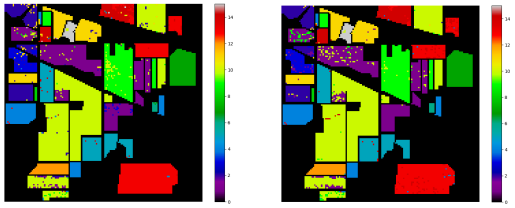
Classification maps are visual representations that illustrate the results of classification tasks between different domains. In each of the mentioned cases, classification maps provide insights into how well a model can classify instances from the source domain (IP, PC, SA, PU) to the target domain (PU). It was observed that PC to PU had major misclassifications in comparison to any other domain.



(a) IP to PU (b) PC to PU (c) SA to PU (d) PU to PU



(e) IP to PC (f) PU to PC (g) PC to PC



(h) SA to IP (i) IP to IP

Figure 6.5: Classification Maps

Table 6.8: Results from PC to PC

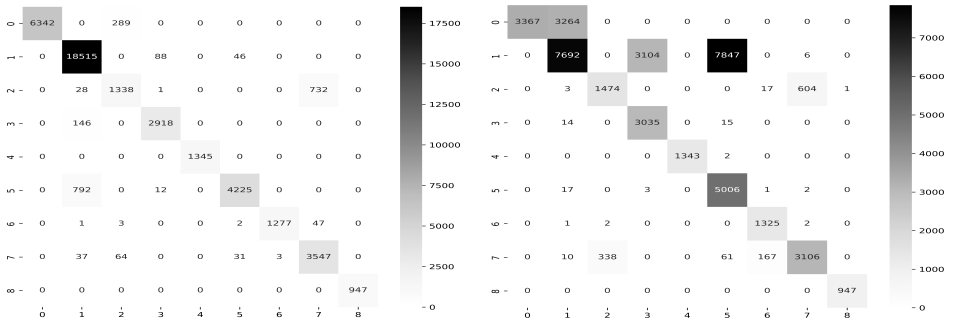
	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	55.26	54.81	75.51	30.23	45.97	85.84	87.72	49.39	59.34	62.26	67.76	65.26
2	85.95	66.11	50.72	70.46	50.18	49.11	48.86	39.75	80.62	83.00	44.62	98.09
3	87.04	51.72	36.75	51.42	55.16	65.57	54.06	87.01	65.94	83.33	82.71	84.17
4	47.96	41.84	35.31	75.80	67.96	32.62	59.29	51.39	44.31	74.63	42.09	69.6
5	58.46	54.14	63.32	31.78	38.78	54.65	65.84	72.37	35.37	31.31	75.42	97.96
6	70.11	85.21	61.33	56.58	40.54	50.23	62.04	68.56	80.39	66.82	31.67	93.41
7	66.08	52.23	76.14	43.57	46.61	64.10	65.13	79.97	56.68	39.90	57.41	93.55
8	63.84	76.27	60.04	40.80	47.64	87.34	52.80	41.14	88.46	79.13	64.55	99.67
9	75.84	62.81	59.31	87.56	65.00	54.45	40.97	70.63	77.57	62.05	75.05	99.96
OA	59.04	27.92	33.93	41.41	55.35	34.02	54.44	46.65	43.86	64.07	40.81	**53.57**
AA	67.84	60.57	57.60	54.24	50.87	60.44	59.64	62.24	65.41	64.72	60.14	**89.07**
KC	52.73	12.73	26.99	38.07	43.64	33.72	50.93	38.86	41.16	59.00	36.15	**47.28**

Table 6.9: Results from PU to PU

	DSC DAG	1DGAN	2DGAN	3DGAN	Zeg-CLIP	ALE	Z-AEG-TF-HIC	CONV-Z-AEG-CV-HIC	Z-AEG-CV-HIC	S-CONV-Z-AEG-TF-HIC	S-CONV-Z-AEG-CV-HIC	**CONVZ-AEG-TF-HIC**
1	55.01	51.73	71.02	81.17	45.23	61.47	87.27	64.73	49.34	61.06	74.01	51.35
2	40.26	31.45	43.81	34.13	43.97	43.91	39.33	35.77	36.83	42.08	58.81	97.49
3	35.12	51.81	67.81	58.87	77.14	35.10	30.21	65.38	60.99	54.68	33.99	77.65
4	32.78	33.48	72.27	34.43	39.88	82.49	46.98	55.61	49.23	70.74	79.73	90.73
5	36.53	82.10	86.46	44.96	51.30	69.42	53.56	62.68	80.78	38.72	78.86	99.92
6	54.20	84.42	44.59	80.48	34.91	88.02	52.14	35.39	53.99	60.04	48.37	96.34
7	45.62	60.52	74.69	58.42	76.89	80.60	77.69	53.40	81.43	82.62	75.21	99.09
8	37.83	69.74	75.42	70.54	72.96	33.26	68.52	45.53	74.52	40.56	50.60	88.94
9	55.03	79.02	58.46	37.64	54.92	79.78	33.44	64.47	35.20	32.82	67.21	100
OA	69.94	67.59	79.78	76.67	76.70	72.59	73.16	78.27	71.52	76.60	72.19	**80.23**
AA	43.60	60.47	66.06	55.63	55.25	63.78	54.35	53.66	58.03	53.70	62.98	**89.05**
KC	66.08	65.96	71.95	65.12	71.66	68.90	71.49	71.95	74.10	68.71	70.80	**72.18**

Confusion Matrix

A confusion matrix is a vital tool for assessing the performance of a machine learning model in classification tasks. It offers a clear breakdown of the model's predictions, highlighting where it correctly and incorrectly classifies instances. The confusion matrices are included in Figure 6.6.



(a) IP to PU

(b) PC to PU

(c) SA to PU

(d) PU to PU

(e) IP to PC

(f) PU to PC

(g) PC to PC

(h) SA to IP

(i) IP to IP

Figure 6.6: Confusion Matrices

6.5 Summary

This chapter presented a comprehensive framework that leverages the synergistic potential of hybrid CNN-GAN-autoencoder models, text embeddings, and zero-shot learning for advanced hyperspectral image classification. The proposed model not only addresses the challenges of limited labelled data but also enhances feature representations through unsupervised and semi-supervised learning paradigms. By seamlessly combining autoencoder-based latent representations with GAN-generated synthetic data, the model demonstrates remarkable performance improvements. In the ever-evolving landscape of knowledge transfer, the **CONVZ-AEG-TF-HIC** model emerges as a beacon of effectiveness and versatility, offering a pioneering solution to the challenges posed by limited data availability. With a meticulous focus on preserving domain expertise, the model seamlessly transfers nuanced insights, ensuring that the transferred knowledge remains both accurate and relevant. Its consistently impressive OA and AA metrics underscore its unwavering ability to facilitate decision-making with precision, particularly when data is scarce. Notably, the model's adeptness at robust feature extraction captures the intricacies of the source domain, translating them into actionable insights in the target domain. This prowess is further fortified by its exceptional KC, signifying resilience in the face of real-world data complexity. The model's end-to-end approach and adaptability across diverse domains enhance its appeal, while its consistent outperformance against existing models solidifies its status as a cutting-edge solution. Ultimately, the **CONVZ-AEG-TF-HIC** model not only revolutionizes and democratizes knowledge transfer but also empowers industries by equipping them with a powerful tool for accurate, adaptable, and impactful knowledge integration, even in data-scarce scenarios.

CHAPTER 7

UNLOCKING THE POTENTIAL OF UNLABELED DATA WITH CONVOLUTIONAL AUTOENCODERS AND GRAPH CONVOLUTIONAL LAYERS

To further address the limitation of labelled data, this chapter also presents a new semi-supervised framework that harmoniously combines unsupervised feature learning with the employment of graph convolutional networks. This approach harnesses the latent knowledge hidden within vast pools of unlabeled hyperspectral data using autoencoders, which extract meaningful features. These features are then incorporated into a GCN based architecture, leveraging spatial relationships among neighbouring pixels. The fusion of unsupervised autoencoder-based learning and graph-based techniques enables our model to achieve remarkable classification accuracy, even in scenarios with minimal labelled samples. Through extensive experimentation, we demonstrate the superior performance and robustness of our methodology across a spectrum of hyperspectral imaging datasets. This work is a significant step in the realm of semi-supervised hyperspectral image analysis, unlocking the potential of unlabeled data and empowering accurate classification in data-scarce environments.

7.1 Introduction

In line with advancements mentioned in preceding chapters, this chapter introduces a novel semi-supervised technique that amalgamates autoencoders and graph-based deep learning. The proposed methodology transcends the boundaries of unsupervised learning by seamlessly fusing supervised classification into the pipeline. Here, graph-based convolutional layers are introduced, and engineered to exploit intricate spatial dependencies within the hyperspectral data. By capitalizing on the learned representations stemming from the un-

supervised phase, the approach bridges the chasm between labelled and unlabelled data, effectively giving birth to a powerful semi-supervised model. Below are the contributions of the proposed model:

- By incorporating AE, our model autonomously uncovers critical patterns and representations within hyperspectral data, reducing its dependency on labelled examples.
- Leveraging GCN, our approach harnesses spatial dependencies within hyperspectral data, enhancing classification accuracy and bridging the gap between limited annotations and model performance.
- Our methodology seamlessly integrates unsupervised and supervised learning, creating a powerful semi-supervised model that maximizes the utility of both labelled and unlabelled data. The presented model attains cutting-edge outcomes on standard hyperspectral image classification datasets, affirming the model's credibility.

Throughout the subsequent sections, we embark on a detailed examination of this methodology, offering an in-depth exploration of the distinct intricacies underlying each component's functionality and the cohesive synergy that unites them. Section 7.2 meticulously elucidates the proposed methodology, elucidating the step-by-step process employed and the rationale behind each strategic decision. Moving forward, Section 7.3 serves as a comprehensive platform for the presentation of the results obtained, along with a meticulous evaluation of the model's performance based on an array of carefully selected metrics. This segment provides a comprehensive and insightful analysis, offering a nuanced understanding of the methodology's effectiveness in practical applications. Lastly, in Section 7.4, we offer a succinct yet comprehensive conclusion that encapsulates the key findings and implications derived from this study. This section serves as a comprehensive synthesis of the research journey, summarizing the key insights garnered and providing potential avenues for future exploration and development in this dynamic field.

7.2 Proposed Methodology

In this research, we emphasize the importance of careful data preprocessing to prepare raw hyperspectral data for analysis and modelling. This is done by introducing a custom function called extract pixels that reshapes the data into a 2D format, merges it with class labels, and stores it as a structured dataset which facilitates further analysis. This enriched data frame containing extracted pixel values and associated class labels serves as the foundation for subsequent data analysis and model development.

7.2.1 Unsupervised Learning

The pre-processed data in equation 7.1 is input by removing noisy data. Figure 7.1 shows the handcrafted autoencoder trained using unsupervised ninety percent data.

$$X_{preprocessed} = RemoveNoise(X, y) \tag{7.1}$$

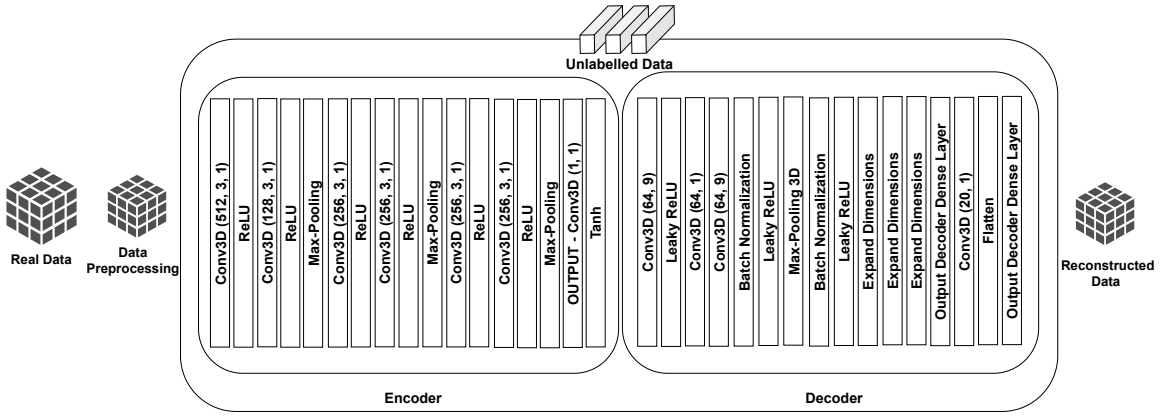


Figure 7.1: Learning using unlabelled data

Initially, all the class labels are eliminated, and the data is left with only the raw pixel values. Then, the data is reshaped to ensure it is ready for the subsequent steps in the process. The first step involves encoding the input hyperspectral data to transform it into a condensed latent space representation. This transformation is aimed at simplifying the data while retaining its essential information. The decoder assumes a vital role in reconstructing the initial data from this condensed representation. The main goal during the training of the autoencoder is to reduce the differences between the input data and the reconstructed data. This is typically achieved using a widely used measure, the mean squared error, which helps in quantifying the overall dissimilarity between the two sets of data.

To capture relevant features, a sequence of 3D convolutional layers is utilized. These specific layers employ the rectified linear unit (ReLU) activation functions shown in equation 7.2 to effectively grasp crucial patterns and connections within the data. Following the convolutional layers, max-pooling layers are used to effectively decrease the dimensions of the data. This reduction in dimensionality aids in simplifying the subsequent processing steps and enhances the overall efficiency of the model.

$$\sigma(x) = \max(0, x) \tag{7.2}$$

The output layer is configured with the necessary dimensions and utilizes a tanh activation function. For training stability, batch normalization is applied, and an inception block is incorporated to capture intricate patterns. The customized autoencoder model is optimized using Adam optimizer and the Mean Squared Error (MSE) loss function with MSE

serving as the evaluation metric, as illustrated in equation 7.3:

$$\mathrm{MSE}(\mathrm{Y}_{\mathrm{true}}, \mathrm{Y}_{\mathrm{predicted}}) = \frac{1}{n}\left(\sum_{i=1}^{n}(\mathrm{y}^{i}_{\mathrm{True}} - \mathrm{y}^{i}_{\mathrm{Predicted}})^2\right) \tag{7.3}$$

It's important to note that the autoencoder model undergoes training using unsupervised data, meaning the input lacks any labels. To prepare the training data, the raw pixel values are organized to fit the required shape of the model. After this initial setup, the model's encoder component is employed to rebuild the hyperspectral data into a reconstructed form denoted as $\mathrm{X}_{\mathrm{reconstructed}}$. The resulting reconstructed data is then stored in a data frame, ready for subsequent stages of processing. This type of architecture is commonly employed for tasks involving dimensionality reduction and feature extraction. Its underlying principle lies in the ability of the latent space representation to effectively capture crucial information from the initial data, allowing for a more concise and meaningful understanding of the dataset.

7.2.2 Supervised Learning using Graph-Based Convolutional Layers

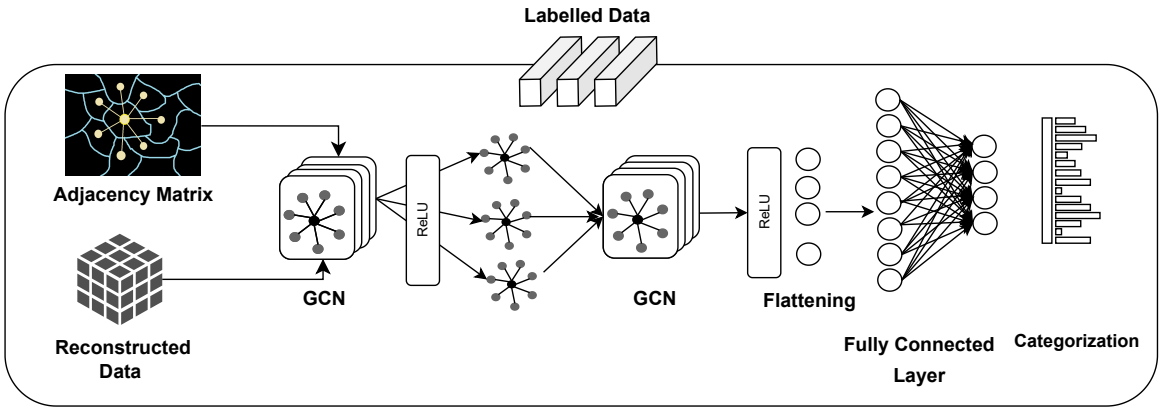


Figure 7.2: Handcrafted GCN for Supervised Learning

Figure 7.2 shows the handcrafted network trained with supervised data. The fine-tuned bands Xunsup are inputted with labelled data X_{sup} , with class labels 0 filtered out represented in the equation 7.4:

$$\mathrm{X}_{\mathrm{sup}},\ y = \mathrm{Filter}(\mathrm{X}_{\mathrm{sup}},\ y,\ \mathrm{class} \neq 0) \tag{7.4}$$

The feature input, denoted as X, is dependent on $\mathrm{num}_{\mathrm{points}}$ and $\mathrm{num}_{\mathrm{features}}$ where $\mathrm{num}_{\mathrm{points}}$ represents the number of data points, and $\mathrm{num}_{\mathrm{features}}$ is the number of features. The data array is normalized using the min-max scaling technique, ensuring that the values are within the range of 0 to 1. This step is often performed to standardize the input data and improve the training process as shown in equation 7.5:

$$X^*_{\text{sup}} = \frac{X_{\text{sup}} - \min(X_{\text{sup}})}{\max(X_{\text{sup}}) - \min(X_{\text{sup}})} \tag{7.5}$$

The graph-based convolution operation in 7.6 is defined as H where X is input data, A is the adjacency matrix computed from pairwise Euclidean distances expressed in equation 7.7 and W represents learnable weight matrix in the graph convolution layer and σ is the activation function.

$$\mathrm{H} = \sigma(\mathrm{A} \cdot \mathrm{X} \cdot \mathrm{W}) \tag{7.6}$$

$$\mathrm{ED} = \sqrt{(\mathrm{x}_2 - \mathrm{x}_1)^2 + (\mathrm{y}_2 - \mathrm{y}_1)^2} \tag{7.7}$$

$$\mathrm{H}_1 = \mathrm{GCN}_1(\mathrm{X}, \mathrm{A}) \tag{7.8}$$

The proposed architecture is centred around the idea of utilizing GCN as part of a CNN. This integration is designed to leverage the inherent structural information within a graph. The graph convolution layers take as input both the feature vectors for each pixel as well as an adjacency matrix that encodes the connections between neighbouring pixels. By performing a multiplication between the adjacency matrix and the feature vectors, the output at each node (pixel) gets influenced by its neighbours, effectively capturing spatial patterns and dependencies.

The key component in this architecture is the GraphConvolution3D layer. This layer operates on the input features X alongside an adjacency matrix A, as shown in equation 7.8 which is a way of representing connections between different elements in a graph. The use of a 3D layer with an output dimension of 128 suggests that it processes data in a three-dimensional format, likely to capture spatial relationships within the data. In practical terms, the GraphConvolution3D layer performs graph convolution, which involves a mathematical operation where the input features and the adjacency matrix are multiplied together. This process enables the model to take into account the interconnections among various elements in the data, thereby allowing it to comprehend intricate relationships that might exist between different components of the input. Following the graph convolution step, a learnable kernel is applied to the results to obtain the final output hidden representation. This indicates that the network learns to weigh the importance of different elements based on their relationships in the graph.

Incorporating non-linear characteristics into the model is achieved by employing the Rectified Linear Unit (ReLU) activation function. This particular function is implemented on each element of the output from the preceding step, enabling the model to detect intricate, non-linear patterns within the data. This addition helps the network to model more

intricate relationships and patterns that might not be linear.

Another GraphConvolution3D layer with an output dimension of 64 is applied to the hidden output along with the adjacency matrix to perform another graph convolution as shown in 7.9. The ReLU activation function is employed element-wise on the output to introduce non-linear characteristics.

$$\mathrm{H}_2 = \mathrm{GCN}_2(\mathrm{H}_1, \mathrm{A}) \tag{7.9}$$

We predict class labels, as illustrated in equation 7.11, by applying a fully connected layer to the flattened output obtained from the graph-based CNN, as shown in equation 7.10.

$$\mathrm{H}_3 = \mathrm{Flatten}(\mathrm{H}_2) \tag{7.10}$$

$$\mathrm{y}_{\mathrm{pred}} = \mathrm{Num}_{\mathrm{classes}}(\mathrm{H}_3) \tag{7.11}$$

The model is combined using Adam as the optimizer with a learning rate parameter 0.0001 and mean square loss error. We train the model using X_{sup} and labels y. We predict class labels for the input data X_{sup} using the trained model.

7.3 Results

The autoencoder component serves a pivotal role in the proposed methodology by learning condensed representations of hyperspectral data, capturing essential features and underlying variations. Convolutional layers within the autoencoder encode local spectral patterns and spatial motifs, with higher layers capturing more abstract concepts. Additionally, unsupervised pretraining initializes the model effectively, facilitating knowledge transfer to the supervised graph convolution component. These features encode discriminative spectral-spatial patterns and aid in boosting overall classification performance when combined with spatial dependencies modelled by the graph convolution component.

To evaluate the efficiency of our suggested method, we conducted experiments using three authentic HSI: IP, PU, and SA. These three datasets offer a diverse range of HSI, each with its distinct characteristics. This diversity enables a comprehensive evaluation of our methodology's performance across various scenarios. Three widely recognized parameters were employed to evaluate the precision of the classification, which included OA, AA and KC.

7.3.1 Parameters Setting

In the noise suppression phase, a series of 14 convolutional layers are employed, each with a 3x3x3 kernel size, ReLU activation function, 'same' padding, and a stride of 1. Four max-pooling layers with a pool size of 2x2x2 are used to down-sample the data. Following noise suppression, the main auto-encoding phase consists of three convolutional layers with kernel sizes of 9x9x9, 1x1x1, and 9x9x9, respectively. Leaky ReLU activation is applied, and batch normalization is used to improve training stability. A max-pooling layer further reduces the feature dimensionality.

The autoencoder is specifically designed with a loss function-centered MSE and is fine-tuned using the Adam optimizer with a learning rate of 0.0001. Additionally, the model incorporates the GraphConvolution3D module to integrate graph-based information, consisting of two graph convolutional layers with output dimensions of 128 and 64, respectively, followed by ReLU activation functions. Similar to the auto-encoder, this part of the model is compiled using the Adam optimizer and MSE loss. Training is performed over 100 epochs with a batch size of 16.

7.3.2 Comparison with Other Models

Our proposed model exhibits several noteworthy advantages compared to existing methods, as demonstrated by the outcomes of our experiments conducted on the IP, PU and SA datasets. When compared to state-of-the-art techniques, such as GCN [336], DBGCN [336], DAE [263] and 3DCAE [337], our model consistently demonstrates superior classification capabilities.

As indicated in Table 7.1, our model achieves a remarkable OA of 94.96% on the IP dataset, surpassing the performance of all other methods. This high OA value emphasizes the model's exceptional capacity to precisely categorize pixels across all classes. Furthermore, the AA of 89.37%, while marginally lower than that of DBGCN[336], still outperforms GCN[336], DAE[263] and 3DCAE[337]. Additionally, the KC of 90.34% demonstrates a substantial level of agreement between the classifications made by the model and the ground truth.

Moving on to the PU dataset shown in Table 7.2, our model once again outperforms all other methods with an impressive OA of 97.57%. This result showcases our model's superior capability in accurately classifying pixels across all categories. The AA of 92.53% further emphasizes the model's remarkable performance, exceeding that of GCN[336], DBGCN[336], DAE[263] and 3DCAE[337]. Moreover, the proposed model's KC outperforms all compared methods, including DBGCN[336], indicating its high reliability and robustness.

On the SA dataset, our proposed model attains remarkable classification results shown

Table 7.1: Classification Accuracy for IP

Classes	**GCN**[336]	**DBGCN**[336]	**DAE**[263]	**3DCAE**[337]	**Proposed**
1	57.89	100.00	35.52	90.48	100.00
2	56.62	73.73	61.13	92.49	84.10
3	30.90	90.27	53.28	90.37	62.04
4	86.46	100.00	63.53	86.90	88.60
5	37.36	65.26	63.26	94.25	96.68
6	77.98	97.65	88.31	97.07	98.08
7	60.00	100.00	30.98	91.26	96.42
8	92.34	100.00	95.65	97.79	100.00
9	83.33	100.00	48.89	75.91	75.00
10	64.11	87.24	75.15	87.34	86.31
11	21.99	68.41	78.78	90.24	95.31
12	27.86	55.73	49.03	95.76	95.78
13	94.92	100.00	89.97	97.49	99.51
14	95.86	100.00	91.65	96.03	95.81
15	44.71	99.74	54.61	90.48	61.65
16	92.94	100.00	85.92	98.82	94.62
OA	53.51	82.30	73.16	92.35	**94.96**
AA	64.03	89.88	66.60	92.04	**89.37**
KC	48.44	80.11			**90.34**

Table 7.2: Classification Accuracy for PU

	GCN[336]	**DBGCN**[336]	**DAE**[263]	**3DCAE**[337]	**Proposed**
1	43.30	78.49	93.69	95.21	100
2	54.41	90.46	96.41	96.06	98.55
3	87.31	90.28	71.88	91.32	57.21
4	83.63	89.65	96.70	98.28	96.57
5	99.33	99.93	99.37	95.55	100.00
6	61.79	100.00	78.83	95.30	86.75
7	95.68	96.29	76.83	95.14	97.21
8	35.43	59.75	88.76	91.38	96.52
9	100.00	92.21	96.72	99.96	100.00
OA	59.23	87.53	91.57	95.39	**97.57**
AA	73.33	88.56	88.80	95.36	**92.53**
KC	49.99	83.62			**94.09**

in Table 7.3, boasting an impressive OA of 92.06%. The AA of 90.09% further underscores the model's remarkable performance, outperforming S2GCN[338], SSCNN[339], and DAE[263]. Although 3DCAE[337] achieves a slightly higher AA, the proposed model remains highly competitive and surpasses S2GCN [338] and SSCNN [339]. The KC of 87.38% signifies a substantial level of agreement between the model's classifications and the ground truth, underscoring the robustness and reliability of our proposed model. Although certain other methods achieve similar KC values, our model consistently delivers competitive performance.

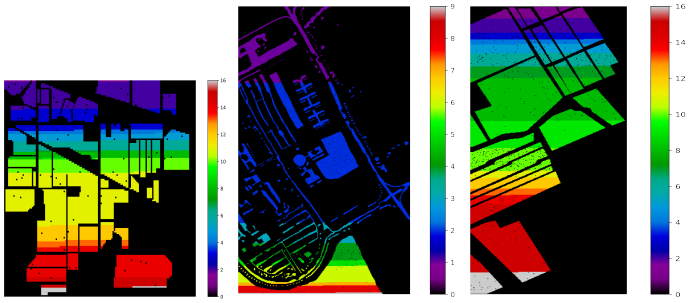
In summary, our proposed model consistently surpasses existing methods in both OA and AA across multiple hyperspectral datasets. These results underscore the model's effectiveness in achieving precise pixel classification. The presented hyperspectral image

Table 7.3: Classification Accuracy for SA

	GCN[336]	**DBGCN**[336]	**DAE**[263]	**3DCAE**[337]	**Proposed**
1	99.01	98.39	96.51	100.00	100.00
2	99.18	94.98	98.35	99.29	100.00
3	97.15	92.74	95.08	97.13	99.79
4	99.11	82.68	98.57	97.91	99.64
5	97.55	97.62	97.19	98.26	97.19
6	99.32	99.49	99.50	99.98	99.97
7	90.06	97.69	98.73	99.64	100.00
8	70.68	83.26	83.83	91.58	90.29
9	98.32	98.86	97.67	99.28	96.22
10	90.97	92.57	92.55	96.65	94.44
11	98.00	95.74	90.89	97.74	94.66
12	99.56	93.97	99.16	98.84	99.89
13	97.83	91.28	96.87	99.26	99.23
14	95.75	92.63	95.54	97.49	98.31
15	70.36	66.81	74.78	87.85	71.84
16	96.90	97.26	87.50	98.34	00.00
OA	88.39	89.37	91.04	95.81	**92.06**
AA	94.30	92.25	93.92	97.95	**90.09**
KC	87.00	88.20			**87.38**

classification model offers several key advantages that distinguish it from existing methods. Firstly, the incorporation of noise suppression at the initial phase significantly enhances the model's ability to handle noisy hyperspectral data, resulting in more accurate classifications. Secondly, the auto-encoding component of the model aids in capturing essential spectral features and reducing data dimensionality, ultimately improving classification performance. The model's adaptive encoding dimensions further optimize its flexibility across various datasets. Moreover, the implementation of the Inception block introduces multi-scale features, enhancing the model's ability to capture complex patterns. Additionally, the proposed model showcases remarkable robustness and stability, as demonstrated by the results obtained through ten repeated experiments with random training pixels. Overall, these advantages collectively position the model as a superior choice for hyperspectral image analysis, offering enhanced accuracy and adaptability compared to existing methods. the incorporation of a GCN in the model's architecture is a pivotal strength. The GCN component makes use of the underlying graph structure found in hyperspectral data, enabling the model to effectively grasp spatial dependencies and connections among adjacent pixels. This feature significantly improves the model's capacity to identify delicate spectral variations and patterns within the data, which is especially crucial in comprehending complex and intricate hyperspectral images. By effectively integrating GCN, the suggested model effectively utilizes both spectral and spatial information, leading to enhanced classification accuracy and resilience, thereby establishing itself as a cutting-edge solution for hyperspectral image analysis.

Figure 7.3 demonstrates the obtained classification maps for the IP, PU, and SA datasets.



(a) IP (b) PU (c) SA

Figure 7.3: Classification Maps

These maps depict the different land cover types, surface materials, or other categories present in the data, with the colours or shades representing the various classes identified during the classification process.

7.4 Summary

This chapter introduced a groundbreaking approach for effectively classifying hyperspectral images. By merging AE and GCN, we achieved highly promising outcomes when tested against various standard datasets. The integration of AE within our model proved to be a critical factor in enhancing the precision of the classification process. This integration allowed us to refine the initial data by minimizing noise and emphasizing clear and valuable features, consequently fortifying the overall reliability of the model and its capacity to extract vital spectral information.

Moreover, the incorporation of a GCN contributed to an enhanced understanding of spatial relationships within hyperspectral data, adding a valuable layer to the model's ability to discern intricate spatial connections. This strategic fusion of spectral and spatial insights enabled our model to thrive, especially in scenarios where spatial associations played a pivotal role, ultimately leading to highly accurate classifications. To conclude, our proposed model marks a significant stride in the realm of hyperspectral image classification, showcasing exceptional performance in various classification tasks.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Over the last few years, researchers have continued to innovate in pursuit of enhanced HSIC in remote sensing and geospatial analysis, spanning vital fields like agriculture, environmental monitoring, and urban planning. However, conventional methods, reliant on manually crafted features, face formidable challenges in handling the intricate non-linearities intrinsic to hyperspectral data. This limitation has sparked interest in the infusion of deep learning techniques for HSIC. Yet, the complexities of hyperspectral data give rise to challenges like the Hughes phenomenon, characterized by high dimensionality, high convergence time, and a scarcity of labelled samples. To address these challenges, this thesis has introduced multiple novel deep learning architectures, namely Xcep-Dense, Deep Siamese, 3D-CAE-SIAM-ATT, CONVZ-AEG-TF-HIC, and AE-GCN, both supervised or semi-supervised, for HSIC.

8.1 Research Contributions

In this thesis, we have conducted a detailed analysis of existing deep learning models and proposed techniques for the classification of HSI. The contributions of this thesis are outlined in the following subsections :

8.1.1 Research Contribution I

The thesis commenced with an introductory overview of the significance of HSIC and the burgeoning interest in leveraging DL techniques for enhanced classification accuracy. It examines each DL model in detail, finding that the CNN stands out for its overall accuracy compared to other models such as LSTM, AE, GAN, and GCN. The study highlights the utilization of three major benchmark datasets—Indian Pines, Pavia University, and Sali-

nas—in existing literature for performance evaluation. The analysis reveals a prevalent reliance on supervised learning in most models, underscoring the study's recommendation for exploring semi-supervised, unsupervised, or alternative learning strategies to effectively address the challenge of limited labelled samples. Overall, the findings shed light on the landscape of DL breakthroughs in HSI classification, highlighting key challenges and suggesting practical solutions.

8.1.2 Research Contribution II

Xcep-Dense: A Novel Lightweight Extreme Inception Model for Hyperspectral Image Classification: This study proposed a lightweight Xcep-Dense network aimed at addressing the issue of extensive parameters present in current deep learning models. By utilizing depth-wise and pointwise separable convolutions, the network effectively reduces parameters while maintaining comparable classification accuracy. Integration of dense modules enhances network depth, facilitating robust extraction of spatial-spectral properties. Notably, this approach demonstrated superior classification accuracy, surpassing state-of-the-art models even with limited training data. The proposed network's efficiency lies in its ability to achieve excellent classification performance with fewer parameters, leading to faster convergence and reduced overfitting.

8.1.3 Research Contribution III

Deep Siamese Network with Handcrafted Feature Extraction for Hyperspectral Image Classification: Deep Siamese, a few-shot learning model contributes significantly by addressing challenges associated with limited training data. Through the utilization of Siamese networks and novel feature generation techniques, the model achieves state-of-the-art classification accuracy while mitigating overfitting and gradient vanishing. Its robustness to imbalanced data and superior feature extraction capabilities make it a valuable asset in hyperspectral image classification.

8.1.4 Research Contribution IV

A 3D Convolutional Autoencoder embedded Siamese Attention Network for Classification of Hyperspectral Images: The 3D-CAE-SIAM-ATT model presents a novel semi-supervised deep learning approach for hyperspectral image classification. The convolutional autoencoder is trained in an unsupervised manner to refine the representation, solving the problem of limited samples. The handcrafted Siamese network enhances the feature separability between samples, by increasing the distance in samples belonging to different classes and minimizing the distance in samples belonging to the same class. The

attention layers aided in extracting the informative spectral-spatial features by suppressing the noisy or less important ones. By integrating autoencoders, Siamese networks, and attention layers, the model demonstrates exceptional performance across both same-domain and cross-domain datasets.

8.1.5 Research Contribution V

Unifying Autoencoder-Enhanced GANs with CNNs and Zero-Shot Learning for HSIC: This work presents a comprehensive framework that leverages hybrid CNN-GAN-autoencoder models, text embeddings, and zero-shot learning for advanced hyperspectral image classification. By seamlessly combining autoencoder-based latent representations with GAN-generated synthetic data, the model enhances feature representations across domains and sensors. Its exceptional performance metrics underscore its effectiveness and versatility across diverse domains, making it a pioneering solution in the hyperspectral image analysis domain.

8.1.6 Research Contribution VI

Unlocking the Potential of Unlabeled Data with Convolutional Autoencoders and Graph Convolutional Layers: This work introduces a groundbreaking approach to hyperspectral image classification by merging Autoencoders with Graph Convolutional Networks. This integration enhances feature extraction and spatial understanding within hyperspectral data, leading to highly accurate classifications. The model's ability to seamlessly integrate unsupervised and supervised learning makes it a valuable asset in scenarios with limited labelled data.

In conclusion, this thesis serves as a testament to the transformative potential of DL in advancing the field of HSIC. Through empirical experimentation and critical analysis, we have elucidated the strengths, limitations, and future avenues of exploration for eachDL model, offering valuable insights for researchers and practitioners alike. As we embark on the next frontier of HSIC research, it is imperative to embrace interdisciplinary collaboration, leverage emerging technologies, and continually push the boundaries of innovation to unlock new possibilities in remote sensing, environmental monitoring, agriculture, defence, security and beyond. As we navigate the complexities of the digital age, let us remain steadfast in our commitment to harnessing the power of DL for the betterment of humanity.

8.2 Future Scope

The future scope of HSIC research extends across multiple dimensions, offering avenues for innovation and exploration. One promising direction lies in transfer learning, where

models like Deep Siamese and Xcep-Dense can benefit from leveraging knowledge gained across different domains and sensor modalities. By embracing transfer learning techniques, researchers can enhance model generalization and adaptability, enabling seamless knowledge transfer and improving classification accuracy across diverse datasets and applications. Additionally, there is significant potential for optimizing model convergence speed and scalability, particularly in frameworks like 3D-CAE-Siam-ATT. fine-tuning model architectures and exploring optimization strategies can help accelerate convergence and extend the applicability of these models to a broader range of domains and disciplines beyond remote sensing and environmental monitoring. Furthermore, the integration of graph convolutional networks with other DL architectures, as demonstrated in AE-GCN, holds promise for unlocking new insights and enhancing classification accuracy across diverse multimodal hyperspectral datasets. By embracing these future directions and fostering a culture of collaboration and innovation, the HSIC research community can chart a course towards transformative discoveries and impactful applications, ultimately empowering humanity to address pressing challenges and realize the full potential of hyperspectral imaging technologies.

BIBLIOGRAPHY

[1] S. Nicholas M. Short. (2011) Technical and historical perspectives of remote sensing. Accessed on: 28 December 2023. [Online]. Available: https://web.archive.org/web/20110703101153/http://rst.gsfc.nasa.gov/Intro/Part2_24.html

[2] B. P. Mondal, "Hyper-spectral analysis of soil properties for soil management," *Advances in agriculture for sustainable development. Srijan Samiti H*, no. 498, pp. 59–65, 2018.

[3] E. Rosenberg and E. Rosenberg, "Computing the correlation dimension," *Fractal Dimensions of Networks*, pp. 195–219, 2020.

[4] A. Ifarraguerri and C.-I. Chang, "Multispectral and hyperspectral image analysis with convex cones," *IEEE transactions on geoscience and remote sensing*, vol. 37, no. 2, pp. 756–770, 1999.

[5] B. Rasti, "Sparse hyperspectral image modeling and restoration," Ph.D. dissertation, 12 2014.

[6] M. Mateen, J. Wen, M. A. Akbar *et al.*, "The role of hyperspectral imaging: A literature review," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 8, 2018.

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

[8] Y. Qu, H. Qi, and C. Kwan, "Unsupervised sparse dirichlet-net for hyperspectral image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2511–2520.

[9] B. Zhang, D. Wu, L. Zhang, Q. Jiao, and Q. Li, "Application of hyperspectral remote sensing for environment monitoring in mining areas," *Environmental Earth Sciences*, vol. 65, pp. 649–658, 2012.

[10] S. Meng, X. Wang, X. Hu, C. Luo, and Y. Zhong, "Deep learning-based crop mapping in the cloudy season using one-shot hyperspectral satellite imagery," *Computers and Electronics in Agriculture*, vol. 186, p. 106188, 2021.

[11] L. Hashemi-Beni and A. Gebrehiwot, "Deep learning for remote sensing image classification for agriculture applications," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 51–54, 2020.

[12] W. Yang, T. Nigon, Z. Hao, G. D. Paiao, F. G. Fernández, D. Mulla, and C. Yang, "Estimation of corn yield based on hyperspectral imagery and convolutional neural network," *Computers and Electronics in Agriculture*, vol. 184, p. 106092, 2021.

[13] E. K. Hege, D. O'Connell, W. Johnson, S. Basty, and E. L. Dereniak, "Hyperspectral imaging for astronomy and space surveillance," in *Imaging Spectrometry IX*, vol. 5159. SPIE, 2004, pp. 380–391.

[14] P. W. Yuen and M. Richardson, "An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition," *The Imaging Science Journal*, vol. 58, no. 5, pp. 241–253, 2010.

[15] G. Lu and B. Fei, "Medical hyperspectral imaging: a review," *Journal of biomedical optics*, vol. 19, no. 1, pp. 010 901–010 901, 2014.

[16] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1264, 2018.

[17] D. Liu, X.-A. Zeng, and D.-W. Sun, "Recent developments and applications of hyperspectral imaging for quality evaluation of agricultural products: a review," *Critical reviews in food science and nutrition*, vol. 55, no. 12, pp. 1744–1757, 2015.

[18] G. M. Foody and A. Mathur, "A relative evaluation of multiclass image classification by support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 6, pp. 1335–1343, 2004.

[19] L. D. Medus, M. Saban, J. V. Francés-Víllora, M. Bataller-Mompeán, and A. Rosado-Muñoz, "Hyperspectral image classification using cnn: Application to industrial food packaging," *Food Control*, vol. 125, p. 107962, 2021.

[20] Q. Zhuang, Y. Peng, D. Yang, Y. Wang, R. Zhao, K. Chao, and Q. Guo, "Detection of frozen pork freshness by fluorescence hyperspectral image," *Journal of Food Engineering*, vol. 316, p. 110840, 2022.

[21] T.-M. Tu, H.-C. Shyu, C.-H. Lee, and C.-I. Chang, "An oblique subspace projection approach for mixed pixel classification in hyperspectral images," *Pattern recognition*, vol. 32, no. 8, pp. 1399–1408, 1999.

[22] S. W. Wharton, "A spectral-knowledge-based approach for urban land-cover discrmination," *IEEE Transactions on Geoscience and Remote Sensing*, no. 3, pp. 272–282, 1987.

[23] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using svms and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.

[24] P. Liu, K.-K. R. Choo, L. Wang, and F. Huang, "Svm or deep learning? a comparative study on remote sensing image classification," *Soft Computing*, vol. 21, pp. 7053–7065, 2017.

[25] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS journal of photogrammetry and remote sensing*, vol. 114, pp. 24–31, 2016.

[26] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308–6325, 2020.

[27] G. Shukla, R. D. Garg, H. S. Srivastava, and P. K. Garg, "An effective implementation and assessment of a random forest classifier as a soil spatial predictive model," *International Journal of Remote Sensing*, vol. 39, no. 8, pp. 2637–2669, 2018.

[28] U. B. Gewali, S. T. Monteiro, and E. Saber, "Machine learning based hyperspectral image analysis: a survey," *arXiv preprint arXiv:1802.08701*, 2018.

[29] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2012.

[30] M. Rao, P. Tang, and Z. Zhang, "A developed siamese cnn with 3d adaptive spatial-spectral pyramid pooling for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 12, p. 1964, 2020.

[31] W. Wang, Y. Chen, X. He, and Z. Li, "Soft augmentation-based siamese cnn for hyperspectral image classification with limited training samples," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[32] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "Feedback attention-based dense cnn for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

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

[34] J. Yin, C. Qi, Q. Chen, and J. Qu, "Spatial-spectral network for hyperspectral image classification: A 3-d cnn and bi-lstm framework," *Remote Sensing*, vol. 13, no. 12, p. 2353, 2021.

[35] J. Zhao, L. Hu, Y. Dong, L. Huang, S. Weng, and D. Zhang, "A combination method of stacked autoencoder and 3d deep residual network for hyperspectral image classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 102, p. 102459, 2021.

[36] S. Zhou, Z. Xue, and P. Du, "Semisupervised stacked autoencoder with cotraining for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3813–3826, 2019.

[37] F. Xie, Q. Gao, C. Jin, and F. Zhao, "Hyperspectral image classification based on superpixel pooling convolutional neural network with transfer learning," *Remote sensing*, vol. 13, no. 5, p. 930, 2021.

[38] Y. Liu, L. Gao, C. Xiao, Y. Qu, K. Zheng, and A. Marinoni, "Hyperspectral image classification based on a shuffled group convolutional neural network with transfer learning," *Remote Sensing*, vol. 12, no. 11, p. 1780, 2020.

[39] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212–216, 2017.

[40] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 3318–3329, 2019.

[41] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, "Classification of hyperspectral imagery using a new fully convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 292–296, 2018.

[42] R. Vaddi and P. Manoharan, "Hyperspectral image classification using cnn with spectral and spatial features integration," *Infrared Physics & Technology*, vol. 107, p. 103296, 2020.

[43] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geoscience and remote sensing letters*, vol. 12, no. 12, pp. 2438–2442, 2015.

[44] X. Wang, K. Tan, and Y. Chen, "Capsnet and triple-gans towards hyperspectral classification," in *2018 Fifth International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*. IEEE, 2018, pp. 1–4.

[45] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network," *Remote Sensing*, vol. 12, no. 3, p. 582, 2020.

[46] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, "Capsule networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2145–2160, 2018.

[47] B. Alotaibi and M. Alotaibi, "A hybrid deep resnet and inception model for hyperspectral image classification," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 88, no. 6, pp. 463–476, 2020.

[48] M. Rao, L. Tang, P. Tang, and Z. Zhang, "Es-cnn: an end-to-end siamese convolutional neural network for hyperspectral image classification," in *2019 Joint Urban Remote Sensing Event (JURSE)*. IEEE, 2019, pp. 1–4.

[49] L. Huang and Y. Chen, "Dual-path siamese cnn for hyperspectral image classification with limited training samples," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 3, pp. 518–522, 2020.

[50] J. Miao, B. Wang, X. Wu, L. Zhang, B. Hu, and J. Q. Zhang, "Deep feature extraction based on siamese network and auto-encoder for hyperspectral image classification," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 397–400.

[51] F. Deng, S. Pu, X. Chen, Y. Shi, T. Yuan, and S. Pu, "Hyperspectral image classification with capsule network using limited training samples," *Sensors*, vol. 18, no. 9, p. 3153, 2018.

[52] Q. Xu, D. Wang, and B. Luo, "Faster multiscale capsule network with octave convolution for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 2, pp. 361–365, 2020.

[53] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.

[54] N. Audebert, B. Le Saux, and S. Lefèvre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE geoscience and remote sensing magazine*, vol. 7, no. 2, pp. 159–173, 2019.

[55] P. Ranjan and A. Girdhar, "Xcep-dense: a novel lightweight extreme inception model for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 43, no. 14, pp. 5204–5230, 2022.

[56] P. Ranjan and G. Gupta, "A cross-domain semi-supervised zero-shot learning model for the classification of hyperspectral images," *Journal of the Indian Society of Remote Sensing*, pp. 1–15, 2023.

[57] P. Ranjan and A. Girdhar, "Deep siamese network with handcrafted feature extraction for hyperspectral image classification," *Multimedia Tools and Applications*, pp. 1–26, 2023.

[58] S. Jia, S. Jiang, Z. Lin, M. Xu, W. Sun, Q. Huang, J. Zhu, and X. Jia, "A semisupervised siamese network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2021.

[59] C. Liu, L. He, Z. Li, and J. Li, "Feature-driven active learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 341–354, 2017.

[60] M. Ahmad, A. K. Bashir, and A. M. Khan, "Metric similarity regularizer to enhance pixel similarity performance for hyperspectral unmixing," *Optik*, vol. 140, pp. 86–95, 2017.

[61] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS journal of photogrammetry and remote sensing*, vol. 66, no. 3, pp. 247–259, 2011.

[62] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1741–1754, 2018.

[63] S. Jia, J. Liao, M. Xu, Y. Li, J. Zhu, W. Sun, X. Jia, and Q. Li, "3-d gabor convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[64] H. Zhang, C. Gong, Y. Bai, Z. Bai, and Y. Li, "3-d-anas: 3-d asymmetric neural architecture search for fast hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–19, 2021.

[65] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-d cnn for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.

[66] Z. Meng, L. Jiao, M. Liang, and F. Zhao, "A lightweight spectral-spatial convolution module for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[67] C. Mu, Z. Dong, and Y. Liu, "A two-branch convolutional neural network based on multi-spectral entropy rate superpixel segmentation for hyperspectral image classification," *Remote Sensing*, vol. 14, no. 7, p. 1569, 2022.

[68] S. Mei, X. Chen, Y. Zhang, J. Li, and A. Plaza, "Accelerating convolutional neural network-based hyperspectral image classification by step activation quantization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2021.

[69] Z. Ge, G. Cao, Y. Zhang, X. Li, H. Shi, and P. Fu, "Adaptive hash attention and lower triangular network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–19, 2021.

[70] J. Yue, D. Zhu, L. Fang, P. Ghamisi, and Y. Wang, "Adaptive spatial pyramid constraint for hyperspectral image classification with limited training samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[71] J. Wang, S. Guo, R. Huang, L. Li, X. Zhang, and L. Jiao, "Dual-channel capsule generation adversarial network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[72] J. Liu, K. Zhang, S. Wu, H. Shi, Y. Zhao, Y. Sun, H. Zhuang, and E. Fu, "An investigation of a multidimensional cnn combined with an attention mechanism model to resolve small-sample problems in hyperspectral image classification," *Remote Sensing*, vol. 14, no. 3, p. 785, 2022.

[73] W. Wei, S. Xu, L. Zhang, J. Zhang, and Y. Zhang, "Boosting hyperspectral image classification with unsupervised feature learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[74] Y.-L. Chang, T.-H. Tan, W.-H. Lee, L. Chang, Y.-N. Chen, K.-C. Fan, and M. Alkhaleefah, "Consolidated convolutional neural network for hyperspectral image classification," *Remote Sensing*, vol. 14, no. 7, p. 1571, 2022.

[75] K. Yang, H. Sun, C. Zou, and X. Lu, "Cross-attention spectral–spatial network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[76] L. Zou, Z. Zhang, H. Du, M. Lei, Y. Xue, and Z. J. Wang, "Da-imrn: Dual-attention-guided interactive multi-scale residual network for hyperspectral image classification," *Remote Sensing*, vol. 14, no. 3, p. 530, 2022.

[77] J. Zhao, Z. Ba, X. Cao, J. Feng, and L. Jiao, "Deep mutual-teaching for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[78] H. Zhai, J. Zhao, and H. Zhang, "Double attention based multilevel one-dimensional convolution neural network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 3771–3787, 2022.

[79] Y. Xu, Z. Li, W. Li, Q. Du, C. Liu, Z. Fang, and L. Zhai, "Dual-channel residual network for hyperspectral image classification with noisy labels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.

[80] C. Shah, Q. Du, and Y. Xu, "Enhanced tabnet: Attentive interpretable tabular learning for hyperspectral image classification," *Remote Sensing*, vol. 14, no. 3, p. 716, 2022.

[81] W. Guo, H. Ye, and F. Cao, "Feature-grouped network with spectral–spatial connected attention for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[82] S. Li, X. Luo, Q. Wang, L. Li, and J. Yin, "H2an: Hierarchical homogeneity-attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[83] Y. Cui, J. Xia, Z. Wang, S. Gao, and L. Wang, "Lightweight spectral–spatial attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[84] B. Xi, J. Li, Y. Li, R. Song, Y. Xiao, Y. Shi, and Q. Du, "Multi-direction networks with attentional spectral prior for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[85] M. E. Paoletti, S. Moreno-Alvarez, and J. M. Haut, "Multiple attention-guided capsule networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–20, 2021.

[86] M. Zhu, J. Fan, Q. Yang, and T. Chen, "Sc-eadnet: A self-supervised contrastive efficient asymmetric dilated network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[87] J. Yue, L. Fang, H. Rahmani, and P. Ghamisi, "Self-supervised learning with adaptive distillation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[88] Z. Gao, L. Tong, J. Zhou, B. Qian, J. Yu, and C. Xiao, "Stochastic depth residual network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[89] Z. Meng, F. Zhao, M. Liang, and W. Xie, "Deep residual involution network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 16, p. 3055, 2021.

[90] H. Yan, J. Wang, L. Tang, E. Zhang, K. Yan, K. Yu, and J. Peng, "A 3d cascaded spectral–spatial element attention network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 13, p. 2451, 2021.

[91] S. Jia, Z. Lin, M. Xu, Q. Huang, J. Zhou, X. Jia, and Q. Li, "A lightweight convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4150–4163, 2020.

[92] J. Xi, O. K. Ersoy, J. Fang, M. Cong, T. Wu, C. Zhao, and Z. Li, "Wide sliding window and subsampling network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 7, p. 1290, 2021.

[93] Q. Sun, X. Liu, and S. Bourennane, "Unsupervised multi-level feature extraction for improvement of hyperspectral classification," *Remote Sensing*, vol. 13, no. 8, p. 1602, 2021.

[94] X. Ding, Y. Li, J. Yang, H. Li, L. Liu, Y. Liu, and C. Zhang, "An adaptive capsule network for hyperspectral remote sensing classification," *Remote Sensing*, vol. 13, no. 13, p. 2445, 2021.

[95] Z. Zhao, D. Hu, H. Wang, and X. Yu, "Center attention network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3415–3425, 2021.

[96] X. Liu, C. Zhang, Z. Cai, J. Yang, Z. Zhou, and X. Gong, "Continuous particle swarm optimization-based deep learning architecture search for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 6, p. 1082, 2021.

[97] Y. Qing and W. Liu, "Hyperspectral image classification based on multi-scale residual network with attention mechanism," *Remote Sensing*, vol. 13, no. 3, p. 335, 2021.

[98] W. N. Khotimah, M. Bennamoun, F. Boussaid, F. Sohel, and D. Edwards, "A high-performance spectral-spatial residual network for hyperspectral image classification with small training data," *Remote Sensing*, vol. 12, no. 19, p. 3137, 2020.

[99] H. Gao, Z. Chen, and C. Li, "Hierarchical shrinkage multiscale network for hyperspectral image classification with hierarchical feature fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 5760–5772, 2021.

[100] Z. Xue, X. Yu, B. Liu, X. Tan, and X. Wei, "Hresnetam: Hierarchical residual network with attention mechanism for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3566–3580, 2021.

[101] Z. Meng, L. Jiao, M. Liang, and F. Zhao, "Hyperspectral image classification with mixed link networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2494–2507, 2021.

[102] M. E. Paoletti, J. M. Haut, N. S. Pereira, J. Plaza, and A. Plaza, "Ghostnet for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 378–10 393, 2021.

[103] B. Yang, H. Li, and Z. Guo, "Learning a deep similarity network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1482–1496, 2020.

[104] T. Alipour-Fard, M. E. Paoletti, J. M. Haut, H. Arefi, J. Plaza, and A. Plaza, "Multi-branch selective kernel networks for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 6, pp. 1089–1093, 2020.

[105] H. Xu, W. Yao, L. Cheng, and B. Li, "Multiple spectral resolution 3d convolutional neural network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 7, p. 1248, 2021.

[106] J. Wang, X. Song, L. Sun, W. Huang, and J. Wang, "A novel cubic convolutional neural network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4133–4148, 2020.

[107] H. Huang, C. Pu, Y. Li, and Y. Duan, "Adaptive residual convolutional neural network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2520–2531, 2020.

[108] B. Fang, Y. Bai, and Y. Li, "Combining spectral unmixing and 3d/2d dense networks with early-exiting strategy for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 5, p. 779, 2020.

[109] Q. Xu, Y. Xiao, D. Wang, and B. Luo, "Csa-mso3dcnn: Multiscale octave 3d cnn with channel and spatial attention for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 1, p. 188, 2020.

[110] H. Guo, J. Liu, J. Yang, Z. Xiao, and Z. Wu, "Deep collaborative attention network for hyperspectral image classification by combining 2-d cnn and 3-d cnn," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4789–4802, 2020.

[111] X. Li, M. Ding, and A. Pižurica, "Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 4, pp. 2615–2629, 2019.

[112] B. Pan, X. Xu, Z. Shi, N. Zhang, H. Luo, and X. Lan, "Dssnet: A simple dilated semantic segmentation network for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 11, pp. 1968–1972, 2020.

[113] Y. Wang, B. Liang, M. Ding, and J. Li, "Dual-branch dense residual network for hyperspectral imagery classification," *International Journal of Remote Sensing*, vol. 41, no. 7, pp. 2581–2602, 2020.

[114] Y. Gao, Y. Feng, and X. Yu, "Feature extraction and classification of hyperspectral images using hierarchical network," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 287–291, 2019.

[115] J. Feng, X. Feng, J. Chen, X. Cao, X. Zhang, L. Jiao, and T. Yu, "Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 7, p. 1149, 2020.

[116] X. He, Y. Chen, and P. Ghamisi, "Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3246–3263, 2019.

[117] P. Dou and C. Zeng, "Hyperspectral image classification using feature relations map learning," *Remote Sensing*, vol. 12, no. 18, p. 2956, 2020.

[118] X. Jiang, Y. Zhang, W. Liu, J. Gao, J. Liu, Y. Zhang, and J. Lin, "Hyperspectral image classification with capsnet and markov random fields," *IEEE Access*, vol. 8, pp. 191 956–191 968, 2020.

[119] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.

[120] X. Jiang, Y. Zhang, Y. Li, S. Li, and Y. Zhang, "Hyperspectral image classification with transfer learning and markov random fields," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 3, pp. 544–548, 2019.

[121] X. Cao, M. Ren, J. Zhao, H. Li, and L. Jiao, "Hyperspectral imagery classification based on compressed convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 9, pp. 1583–1587, 2019.

[122] P. Wu, Z. Cui, Z. Gan, and F. Liu, "Residual group channel and space attention network for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 12, p. 2035, 2020.

[123] X. Zhang, Y. Wang, N. Zhang, D. Xu, H. Luo, B. Chen, and G. Ben, "Ssdanet: Spectral-spatial three-dimensional convolutional neural network for hyperspectral image classification," *Ieee Access*, vol. 8, pp. 127 167–127 180, 2020.

[124] X. Yang, X. Zhang, Y. Ye, R. Y. Lau, S. Lu, X. Li, and X. Huang, "Synergistic 2d/3d convolutional neural network for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 12, p. 2033, 2020.

[125] W. Wang, S. Dou, and S. Wang, "Alternately updated spectral–spatial convolution network for the classification of hyperspectral images," *Remote Sensing*, vol. 11, no. 15, p. 1794, 2019.

[126] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-d-cnn with transfer learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5813–5828, 2019.

[127] Y. Bai, Q. Zhang, Z. Lu, and Y. Zhang, "Ssdc-densenet: A cost-effective end-to-end spectral-spatial dual-channel dense network for hyperspectral image classification," *IEEE Access*, vol. 7, pp. 84 876–84 889, 2019.

[128] M. Khodadadzadeh, X. Ding, P. Chaurasia, and D. Coyle, "A hybrid capsule network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 11 824–11 839, 2021.

[129] C. Mu, Q. Zeng, Y. Liu, and Y. Qu, "A two-branch network combined with robust principal component analysis for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 12, pp. 2147–2151, 2020.

[130] L. Hong and M. Zhang, "Object-oriented multiscale deep features for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 41, no. 14, pp. 5549–5572, 2020.

[131] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 740–754, 2018.

[132] H. Gao, Y. Yang, D. Yao, and C. Li, "Hyperspectral image classification with preactivation residual attention network," *IEEE Access*, vol. 7, pp. 176 587–176 599, 2019.

[133] Y. Wang, J. Mei, L. Zhang, B. Zhang, P. Zhu, Y. Li, and X. Li, "Self-supervised feature learning with crf embedding for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 2628–2642, 2018.

[134] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE transactions on geoscience and remote sensing*, vol. 57, no. 10, pp. 8065–8080, 2019.

[135] B. Zhang, C. Qing, X. Xu, and J. Ren, "Spatial residual blocks combined parallel network for hyperspectral image classification," *IEEE Access*, vol. 8, pp. 74 513–74 524, 2020.

[136] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2017.

[137] B. Pan, Z. Shi, and X. Xu, "Mugnet: Deep learning for hyperspectral image classification using limited samples," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 108–119, 2018.

[138] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 120–147, 2018.

[139] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu, and X. Cao, "Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 10, pp. 5585–5599, 2017.

[140] S. Hou, H. Shi, X. Cao, X. Zhang, and L. Jiao, "Hyperspectral imagery classification based on contrastive learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[141] H. Madani and K. McIsaac, "Distance transform-based spectral-spatial feature vector for hyperspectral image classification with stacked autoencoder," *Remote Sensing*, vol. 13, no. 9, p. 1732, 2021.

[142] S. Li, X. Zhu, Y. Liu, and J. Bao, "Adaptive spatial-spectral feature learning for hyperspectral image classification," *IEEE access*, vol. 7, pp. 61 534–61 547, 2019.

[143] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4823–4833, 2019.

[144] S. Paul and D. N. Kumar, "Spectral-spatial classification of hyperspectral data with mutual information-based segmented stacked autoencoder approach," *ISPRS journal of photogrammetry and remote sensing*, vol. 138, pp. 265–280, 2018.

[145] R. Kemker and C. Kanan, "Self-taught feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2693–2705, 2017.

[146] B. Yang, F. Cao, and H. Ye, "A novel method for hyperspectral image classification: Deep network with adaptive graph structure integration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[147] J. Chen, L. Jiao, X. Liu, L. Li, F. Liu, and S. Yang, "Automatic graph learning convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[148] F. Luo, Z. Zou, J. Liu, and Z. Lin, "Dimensionality reduction and classification of hyperspectral image via multistructure unified discriminative embedding," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[149] J.-Y. Yang, H.-C. Li, W.-S. Hu, L. Pan, and Q. Du, "Adaptive cross-attention-driven spatial–spectral graph convolutional network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[150] X. He, Y. Chen, and P. Ghamisi, "Dual graph convolutional network for hyperspectral image classification with limited training samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2021.

[151] S. Wan, S. Pan, P. Zhong, X. Chang, J. Yang, and C. Gong, "Dual interactive graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[152] Y. Zhang, G. Cao, B. Wang, X. Li, P. Y. O. Amoako, and A. Shafique, "Dual sparse representation graph-based copropagation for semisupervised hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2021.

[153] S. Zhong, T. Zhou, S. Wan, J. Yang, and C. Gong, "Dynamic spectral–spatial poisson learning for hyperspectral image classification with extremely scarce labels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[154] H. Zhang, J. Zou, and L. Zhang, "Ems-gcn: An end-to-end mixhop superpixel-based graph convolutional network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[155] H. Hu, M. Yao, F. He, and F. Zhang, "Graph neural network via edge convolution for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[156] J. Bai, B. Ding, Z. Xiao, L. Jiao, H. Chen, and A. C. Regan, "Hyperspectral image classification based on deep attention graph convolutional network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[157] Z. Ma, Z. Jiang, and H. Zhang, "Hyperspectral image classification using feature fusion hypergraph convolution neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[158] S. Liu, Y. Cao, Y. Wang, J. Peng, P. T. Mathiopoulos, and Y. Li, "Dfl-lc: Deep feature learning with label consistencies for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3669–3681, 2021.

[159] Y. Li, B. Xi, J. Li, R. Song, Y. Xiao, and J. Chanussot, "Sgml: A symmetric graph metric learning framework for efficient hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 609–622, 2021.

[160] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2020.

[161] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 12, pp. 8246–8257, 2020.

[162] P. Sellars, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4180–4193, 2020.

[163] X. Zhang, S. Chen, P. Zhu, X. Tang, J. Feng, and L. Jiao, "Spatial pooling graph convolutional network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[164] J. Feng, J. Zhang, and Y. Zhang, "A multiview spectral–spatial feature extraction and fusion framework for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[165] B. Tu, W. He, W. He, X. Ou, and A. Plaza, "Hyperspectral classification via global-local hierarchical weighting fusion network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 184–200, 2021.

[166] Y. Ding, X. Zhao, Z. Zhang, W. Cai, N. Yang, and Y. Zhan, "Semi-supervised locality preserving dense graph neural network with arma filters and context-aware learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2021.

[167] X. He and Y. Chen, "Transferring cnn ensemble for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 5, pp. 876–880, 2020.

[168] Q. Lv, W. Feng, Y. Quan, G. Dauphin, L. Gao, and M. Xing, "Enhanced-random-feature-subspace-based ensemble cnn for the imbalanced hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3988–3999, 2021.

[169] W.-Y. Wang, H.-C. Li, Y.-J. Deng, L.-Y. Shao, X.-Q. Lu, and Q. Du, "Generative adversarial capsule network with convlstm for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 3, pp. 523–527, 2020.

[170] C. Zhang, J. Yue, and Q. Qin, "Deep quadruplet network for hyperspectral image classification with a small number of samples," *Remote Sensing*, vol. 12, no. 4, p. 647, 2020.

[171] W. Masarczyk, P. Głomb, B. Grabowski, and M. Ostaszewski, "Effective training of deep convolutional neural networks for hyperspectral image classification through artificial labeling," *Remote Sensing*, vol. 12, no. 16, p. 2653, 2020.

[172] Z. Zheng, Y. Zhong, A. Ma, and L. Zhang, "Fpga: Fast patch-free global learning framework for fully end-to-end hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5612–5626, 2020.

[173] J. Wang, F. Gao, J. Dong, and Q. Du, "Adaptive dropblock-enhanced generative adversarial networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5040–5053, 2020.

[174] Y. Quan, X. Zhong, W. Feng, J. C.-W. Chan, Q. Li, and M. Xing, "Smote-based weighted deep rotation forest for the imbalanced hyperspectral data classification," *Remote Sensing*, vol. 13, no. 3, p. 464, 2021.

[175] H. Liang, W. Bao, and X. Shen, "Adaptive weighting feature fusion approach based on generative adversarial network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 2, p. 198, 2021.

[176] X. Ma, X. Mou, J. Wang, X. Liu, J. Geng, and H. Wang, "Cross-dataset hyperspectral image classification based on adversarial domain adaptation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4179–4190, 2020.

[177] G. Wang and P. Ren, "Hyperspectral image classification with feature-oriented adversarial active learning," *Remote Sensing*, vol. 12, no. 23, p. 3879, 2020.

[178] J. Feng, H. Yu, L. Wang, X. Cao, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5329–5343, 2019.

[179] S. Singh and S. S. Kasana, "Hyperspectral image classification using spectral lstm networks," in *The 40th Asian conference on remote sensing*, 2019, pp. 1–7.

[180] J. Liu, X. Zhang, J. Zhang, J. An, C. Li, and L. Gao, "Hyperspectral image classification based on long short-term memory network," in *2018 Fifth International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*. IEEE, 2018, pp. 1–5.

[181] W.-S. Hu, H.-C. Li, T.-Y. Ma, Q. Du, A. Plaza, and W. J. Emery, "Hyperspectral image classification based on tensor-train convolutional long short-term memory," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 858–861.

[182] A. Ma, A. M. Filippi, Z. Wang, and Z. Yin, "Hyperspectral image classification using similarity measurements-based deep recurrent neural networks," *Remote Sensing*, vol. 11, no. 2, p. 194, 2019.

[183] B. Huang, Z. Wang, J. Shang, G. Chen, and M. Radenkovic, "A spectral sequence-based nonlocal long short-term memory network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 3041–3051, 2022.

[184] S. Mei, X. Li, X. Liu, H. Cai, and Q. Du, "Hyperspectral image classification using attention-based bidirectional long short-term memory network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2021.

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

[186] Z. Wang, C. Zou, and W. Cai, "Small sample classification of hyperspectral remote sensing images based on sequential joint deeping learning model," *Ieee Access*, vol. 8, pp. 71 353–71 363, 2020.

[187] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5384–5394, 2019.

[188] Z. Li, X. Zhao, Y. Xu, W. Li, L. Zhai, Z. Fang, and X. Shi, "Hyperspectral image classification with multiattention fusion network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[189] X. Wang and Y. Fan, "Multiscale densely connected attention network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 1617–1628, 2022.

[190] Z. Zhang, D. Liu, D. Gao, and G. Shi, "S^3net: Spectral–spatial–semantic network for hyperspectral image classification with the multiway attention mechanism," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2021.

[191] X. Li, M. Ding, and A. Pižurica, "Spectral feature fusion networks with dual attention for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[192] X. Zhang, S. Shang, X. Tang, J. Feng, and L. Jiao, "Spectral partitioning residual network with spatial attention mechanism for hyperspectral image classification," *IEEE transactions on geoscience and remote sensing*, vol. 60, pp. 1–14, 2021.

[193] X. Mei, E. Pan, Y. Ma, X. Dai, J. Huang, F. Fan, Q. Du, H. Zheng, and J. Ma, "Spectral-spatial attention networks for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 8, p. 963, 2019.

[194] S. Huang, Z. Liu, W. Jin, and Y. Mu, "A superpixel-correlation-based multiview approach for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[195] H. Fu, G. Sun, J. Ren, A. Zhang, and X. Jia, "Fusion of pca and segmented-pca domain multiscale 2-d-ssa for effective spectral-spatial feature extraction and data classification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2020.

[196] W. Wang, X. Liu, and X. Mou, "Data augmentation and spectral structure features for limited samples hyperspectral classification," *Remote Sensing*, vol. 13, no. 4, p. 547, 2021.

[197] D. Li, X. Wang, and Y. Cheng, "Spatial-spectral neighbour graph for dimensionality reduction of hyperspectral image classification," *International Journal of Remote Sensing*, vol. 40, no. 11, pp. 4361–4383, 2019.

[198] A. Challa, S. Danda, B. D. Sagar, and L. Najman, "Triplet-watershed for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[199] Y. Guo, H. Cao, S. Han, Y. Sun, and Y. Bai, "Spectral–spatial hyperspectralimage classification with k-nearest neighbor and guided filter," *IEEE Access*, vol. 6, pp. 18 582–18 591, 2018.

[200] C. Ding, M. Zheng, F. Chen, Y. Zhang, X. Zhuang, E. Fan, D. Wen, L. Zhang, W. Wei, and Y. Zhang, "Hyperspectral image classification promotion using clustering inspired active learning," *Remote Sensing*, vol. 14, no. 3, p. 596, 2022.

[201] M. Xu, Q. Zhao, and S. Jia, "Multiview spatial–spectral active learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[202] B. Liu, A. Yu, P. Zhang, L. Ding, W. Guo, K. Gao, and X. Zuo, "Active deep densely connected convolutional network for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 42, no. 15, pp. 5915–5934, 2021.

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

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

[205] Z. Zhong, Y. Li, L. Ma, J. Li, and W.-S. Zheng, "Spectral–spatial transformer network for hyperspectral image classification: A factorized architecture search framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

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

[207] X. Li, Z. Cao, L. Zhao, and J. Jiang, "Alpn: Active-learning-based prototypical network for few-shot hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[208] J. Bai, S. Huang, Z. Xiao, X. Li, Y. Zhu, A. C. Regan, and L. Jiao, "Few-shot hyperspectral image classification based on adaptive subspaces and feature transformation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[209] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2290–2304, 2018.

[210] Y. Dong, C. Yang, and Y. Zhang, "Deep metric learning with online hard mining for hyperspectral classification," *Remote Sensing*, vol. 13, no. 7, p. 1368, 2021.

[211] S. Pu, Y. Wu, X. Sun, and X. Sun, "Hyperspectral image classification with localized graph convolutional filtering," *Remote Sensing*, vol. 13, no. 3, p. 526, 2021.

[212] N. Li, D. Zhou, J. Shi, M. Zhang, T. Wu, and M. Gong, "Deep fully convolutional embedding networks for hyperspectral images dimensionality reduction," *Remote Sensing*, vol. 13, no. 4, p. 706, 2021.

[213] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[214] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource remote sensing data classification based on convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 937–949, 2017.

[215] X. Kang, C. Li, S. Li, and H. Lin, "Classification of hyperspectral images by gabor filtering based deep network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1166–1178, 2017.

[216] L. Chen, Z. Wei, and Y. Xu, "A lightweight spectral–spatial feature extraction and fusion network for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 9, p. 1395, 2020.

[217] J. Lin, R. Ward, and Z. J. Wang, "Deep transfer learning for hyperspectral image classification," in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2018, pp. 1–5.

[218] Y. Shi, D. Ma, J. Lv, and J. Li, "Actl: Asymmetric convolutional transfer learning for tree species identification based on deep neural network," *IEEE Access*, vol. 9, pp. 13 643–13 654, 2021.

[219] J. Nalepa, M. Myller, and M. Kawulok, "Transfer learning for segmenting dimensionally reduced hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 7, pp. 1228–1232, 2019.

[220] Y. Jiang, Y. Li, and H. Zhang, "Hyperspectral image classification based on 3-d separable resnet and transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1949–1953, 2019.

[221] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-learning-incorporated deep transfer learning for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4048–4062, 2018.

[222] X. Zhao, Y. Liang, A. J. Guo, and F. Zhu, "Classification of small-scale hyperspectral images with multi-source deep transfer learning," *Remote Sensing Letters*, vol. 11, no. 4, pp. 303–312, 2020.

[223] Q. Sun and S. Bourennane, "Hyperspectral image classification with unsupervised feature extraction," *Remote Sensing Letters*, vol. 11, no. 5, pp. 475–484, 2020.

[224] J. Feng and J. Zhang, "Unsupervised feature extraction in hyperspectral image based on improved neighborhood preserving embedding," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 1291–1294.

[225] B. Tu, X. Zhang, G. Zhang, J. Wang, and W. He, "Dual unsupervised features fusion for hyperspectral image classification," *International Journal of Remote Sensing*, vol. 41, no. 16, pp. 6135–6156, 2020.

[226] Y. Kong, X. Wang, Y. Cheng, and C. P. Chen, "Hyperspectral imagery classification based on semi-supervised broad learning system," *Remote sensing*, vol. 10, no. 5, p. 685, 2018.

[227] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[228] Z. He and D. He, "Spatial-adaptive siamese residual network for multi-/hyperspectral classification," *Remote Sensing*, vol. 12, no. 10, p. 1640, 2020.

[229] S. Chen, M. Jin, and J. Ding, "Hyperspectral remote sensing image classification based on dense residual three-dimensional convolutional neural network," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 1859–1882, 2021.

[230] S. Pande and B. Banerjee, "Dimensionality reduction using 3d residual autoencoder for hyperspectral image classification," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 2029–2032.

[231] X. Liu, Y. Meng, and M. Fu, "Classification research based on residual network for hyperspectral image," in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2019, pp. 911–915.

[232] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns," *IEEE transactions on geoscience and remote sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.

[233] E. Pan, Y. Ma, F. Fan, X. Mei, and J. Huang, "Hyperspectral image classification across different datasets: A generalization to unseen categories," *Remote Sensing*, vol. 13, no. 9, p. 1672, 2021.

[234] L. Wei, K. Wang, Q. Lu, Y. Liang, H. Li, Z. Wang, R. Wang, and L. Cao, "Crops fine classification in airborne hyperspectral imagery based on multi-feature fusion and deep learning," *Remote Sensing*, vol. 13, no. 15, p. 2917, 2021.

[235] J. Liu, Z. Yang, Y. Liu, and C. Mu, "Hyperspectral remote sensing images deep feature extraction based on mixed feature and convolutional neural networks," *Remote Sensing*, vol. 13, no. 13, p. 2599, 2021.

[236] H. Shi, G. Cao, Z. Ge, Y. Zhang, and P. Fu, "Double-branch network with pyramidal convolution and iterative attention for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 7, p. 1403, 2021.

[237] C. Zhang, J. Wang, and K. Yao, "Global random graph convolution network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 12, p. 2285, 2021.

[238] J. Xi, M. Cong, O. K. Ersoy, W. Zou, C. Zhao, Z. Li, J. Gu, and T. Wu, "Dynamic wide and deep neural network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 13, p. 2575, 2021.

[239] X. He, Y. Chen, and Z. Lin, "Spatial-spectral transformer for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 3, p. 498, 2021.

[240] C. Chen, Y. Ma, and G. Ren, "A convolutional neural network with fletcher–reeves algorithm for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 11, p. 1325, 2019.

[241] M. Pal and P. M. Mather, "Support vector machines for classification in remote sensing," *International journal of remote sensing*, vol. 26, no. 5, pp. 1007–1011, 2005.

[242] D. Akbari, S. Homayouni, A. Safari, and N. Mehrshad, "Mapping urban land cover based on spatial-spectral classification of hyperspectral remote-sensing data," *International Journal of Remote Sensing*, vol. 37, no. 2, pp. 440–454, 2016.

[243] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal processing magazine*, vol. 19, no. 1, pp. 17–28, 2002.

[244] F. Lacar, M. Lewis, and I. Grierson, "Use of hyperspectral imagery for mapping grape varieties in the barossa valley, south australia," in *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*, vol. 6. IEEE, 2001, pp. 2875–2877.

[245] T. J. Malthus and P. J. Mumby, "Remote sensing of the coastal zone: an overview and priorities for future research," *International JOurnal of Remote Sensing*, 2003.

[246] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[247] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[248] X. He and Y. Chen, "Optimized input for cnn-based hyperspectral image classification using spatial transformer network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1884–1888, 2019.

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

[250] M. Ahmad, M. Mazzara, and S. Distefano, "3d/2d regularized cnn feature hierarchy for hyperspectral image classification," *arXiv preprint arXiv:2104.12136*, 2021.

[251] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[252] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[253] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.

[254] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[255] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[256] Z. Xiong, Y. Yuan, and Q. Wang, "Ai-net: Attention inception neural networks for hyperspectral image classification," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 2647–2650.

[257] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018.

[258] J. Zhao, L. Hu, Y. Dong, and L. Huang, "Hybrid dense network with dual attention for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 23, p. 4921, 2021.

[259] W. Wang, S. Dou, Z. Jiang, and L. Sun, "A fast dense spectral–spatial convolution network framework for hyperspectral images classification," *Remote sensing*, vol. 10, no. 7, p. 1068, 2018.

[260] T. Zhang, C. Shi, D. Liao, and L. Wang, "Deep spectral spatial inverted residual network for hyperspectral image classification," *Remote Sensing*, vol. 13, no. 21, p. 4472, 2021.

[261] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 5893–5909, 2018.

[262] A. Diakite, G. Jiangsheng, and F. Xiaping, "Hyperspectral image classification using 3d 2d cnn," *IET Image Processing*, vol. 15, no. 5, pp. 1083–1092, 2021.

[263] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected topics in applied earth observations and remote sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[264] J. Zhang, F. Wei, F. Feng, and C. Wang, "Spatial–spectral feature refinement for hyperspectral image classification based on attention-dense 3d-2d-cnn," *Sensors*, vol. 20, no. 18, p. 5191, 2020.

[265] M. Wong and C.-C. Hung, "Dimensionality reduction with weighted k-means for hyperspectral image classification," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 44–47.

[266] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279–317, 2019.

[267] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[268] C. Chen, F. Jiang, C. Yang, S. Rho, W. Shen, S. Liu, and Z. Liu, "Hyperspectral classification based on spectral–spatial convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 165–171, 2018.

[269] M. Zhang, M. Gong, H. He, and S. Zhu, "Symmetric all convolutional neural-network-based unsupervised feature extraction for hyperspectral images classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2981–2993, 2020.

[270] F. Feng, S. Wang, C. Wang, and J. Zhang, "Learning deep hierarchical spatial–spectral features for hyperspectral image classification based on residual 3d-2d cnn," *Sensors*, vol. 19, no. 23, p. 5276, 2019.

[271] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sensing*, vol. 11, no. 11, p. 1307, 2019.

[272] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *Ieee Access*, vol. 6, pp. 14 118–14 129, 2018.

[273] X. Liao, K. Li, X. Zhu, and K. R. Liu, "Robust detection of image operator chain with two-stream convolutional neural network," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 955–968, 2020.

[274] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 897–911, 2020.

[275] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 3, pp. 685–696, 2019.

[276] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5408–5423, 2018.

[277] I. Dópido, J. Li, P. R. Marpu, A. Plaza, J. M. B. Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE transactions on geoscience and remote sensing*, vol. 51, no. 7, pp. 4032–4044, 2013.

[278] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sensing Letters*, vol. 8, no. 9, pp. 839–848, 2017.

[279] H. Gao, J. Zhang, X. Cao, Z. Chen, Y. Zhang, and C. Li, "Dynamic data augmentation method for hyperspectral image classification based on siamese structure,"

IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 8063–8076, 2021.

[280] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 862–873, 2009.

[281] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear pca for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 447–451, 2011.

[282] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE transactions on Geoscience and remote sensing*, vol. 49, no. 12, pp. 4865–4876, 2011.

[283] M. Dalla Mura, A. Villa, J. A. Benediktsson, J. Chanussot, and L. Bruzzone, "Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 542–546, 2010.

[284] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[285] H. Song, W. Yang, H. Yuan, and H. Bufford, "Deep 3d-multiscale densenet for hyperspectral image classification based on spatial-spectral information," *Intelligent Automation & Soft Computing*, vol. 26, no. 4, pp. 1441–1458, 2020.

[286] Y. Li, W. Xie, and H. Li, "Hyperspectral image reconstruction by deep convolutional neural network for classification," *Pattern Recognition*, vol. 63, pp. 371–383, 2017.

[287] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, no. 1. Lille, 2015.

[288] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[289] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3642–3649.

[290] H. Huang, M. Chen, and Y. Duan, "Dimensionality reduction of hyperspectral image using spatial-spectral regularized sparse hypergraph embedding," *Remote Sensing*, vol. 11, no. 9, p. 1039, 2019.

[291] X. Tong, J. Yin, B. Han, and H. Qv, "Few-shot learning with attention-weighted graph convolutional networks for hyperspectral image classification," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1686–1690.

[292] D. Pathak, S. Kalita, and D. Bhattacharya, "Hyperspectral image classification using support vector machine: a spectral-spatial feature-based approach," *Evolutionary Intelligence*, vol. 15, pp. 1809–1823, 2022.

[293] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, pp. 1–12, 2015.

[294] Q. Zhu, W. Deng, Z. Zheng, Y. Zhong, Q. Guan, W. Lin, L. Zhang, and D. Li, "A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 11, pp. 11 709–11 723, 2021.

[295] L. N. Eeti and K. M. Buddhiraju, "Two hidden layer neural network-based rotation forest ensemble for hyperspectral image classification," *Geocarto International*, vol. 36, no. 16, pp. 1820–1837, 2021.

[296] J. Lin, L. Mou, X. X. Zhu, X. Ji, and Z. J. Wang, "Attention-aware pseudo-3-d convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7790–7802, 2021.

[297] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 1909–1921, 2017.

[298] W. Cai and Z. Wei, "Remote sensing image classification based on a cross-attention mechanism and graph convolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[299] N. Javaid, N. Jan, and M. U. Javed, "An adaptive synthesis to handle imbalanced big data with deep siamese network for electricity theft detection in smart grids," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 44–52, 2021.

[300] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference*

on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, 2005, pp. 539–546 vol. 1.

[301] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2017.

[302] Grupo de Inteligencia Computacional (GIC). (n.d.) Hyperspectral remote sensing scenes. Accessed: January 31, 2022. [Online]. Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

[303] T.-M. Tu, C.-H. Chen, J.-L. Wu, and C.-I. Chang, "A fast two-stage classification method for high-dimensional remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 1, pp. 182–191, 1998.

[304] M. Graña, M. Veganzons, and B. Ayerdi, "Computational intelligence group (gic)," *Hyperspectral Remote Sensing Scenes*, 2021.

[305] L. Chapel, T. Burger, N. Courty, and S. Lefèvre, "Classwise hyperspectral image classification with perturbo method," in *2012 IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 6883–6886.

[306] P. Ranjan, R. Kumar, and A. Girdhar, "Recent cnn advancements for stratification of hyperspectral images," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*. IEEE, 2023, pp. 1–5.

[307] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[308] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3d convolutional autoencoder for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6808–6820, 2019.

[309] S. Liu, Q. Shi, and L. Zhang, "Few-shot hyperspectral image classification with unknown classes using multitask deep learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5085–5102, 2020.

[310] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE geoscience and remote sensing magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[311] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," *Journal of applied remote sensing*, vol. 11, no. 4, pp. 042 609–042 609, 2017.

[312] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, "3d deep learning on medical images: a review," *Sensors*, vol. 20, no. 18, p. 5097, 2020.

[313] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020.

[314] L. Samaniego, A. Bárdossy, and K. Schulz, "Supervised classification of remotely sensed imagery using a modified k-nn technique," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2112–2125, 2008.

[315] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4085–4098, 2010.

[316] J. Ediriwickrema and S. Khorram, "Hierarchical maximum-likelihood classification for improved accuracies," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 810–816, 1997.

[317] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[318] L. Wang, L. Wang, Q. Wang, and P. M. Atkinson, "Ssa-siamnet: Spectral–spatial-wise attention-based siamese network for hyperspectral image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2021.

[319] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[320] Z. Li, M. Liu, Y. Chen, Y. Xu, W. Li, and Q. Du, "Deep cross-domain few-shot learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2022.

[321] Z. Xue, T. Zhu, Y. Zhou, and M. Zhang, "Bag-of-features-driven spectral-spatial siamese neural network for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1085–1099, 2022.

[322] Z. Xue, Y. Zhou, and P. Du, "S3net: Spectral–spatial siamese network for few-shot hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–19, 2022.

[323] Z. Xue, Z. Liu, and M. Zhang, "Dsr-gcn: Differentiated-scale restricted graph convolutional network for few-shot hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

[324] S. Jiang and S. Jia, "A 3d lightweight siamese network for hyperspectral image classification with limited samples," in *Proceedings of the 2021 10th International Conference on Computing and Pattern Recognition*, 2021, pp. 142–148.

[325] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017.

[326] C. Pu, H. Huang, and L. Luo, "Classification of hyperspectral image with attention mechanism-based dual-path convolutional network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[327] X. Cao, C. Li, J. Feng, and L. Jiao, "Semi-supervised feature learning for disjoint hyperspectral imagery classification," *Neurocomputing*, vol. 526, pp. 9–18, 2023.

[328] A. Sellami, M. Farah, and D. M., "Shcnet: A semi-supervised hypergraph convolutional networks based on relevant feature selection for hyperspectral image classification," *Pattern Recognition Letters*, vol. 165, pp. 98–106, 2023.

[329] X. Wang, J. Liu, W. Wang, W. Chi, and R. Feng, "Weakly supervised hyperspectral image classification with few samples based on intradomain sample expansion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023.

[330] Z. Yu, Z. Zhang, W. Cao, C. Liu, C. P. Chen, and H.-S. Wong, "Gan-based enhanced deep subspace clustering networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3267–3281, 2020.

[331] Z. Cao, X. Li, Y. Feng, S. Chen, C. Xia, and L. Zhao, "Contrastnet: Unsupervised feature learning by autoencoder and prototypical contrastive learning for hyperspectral imagery classification," *Neurocomputing*, vol. 460, pp. 71–83, 2021.

[332] P. Ranjan, R. Kumar, and A. Girdhar, "A 3d-convolutional-autoencoder embedded siamese-attention-network for classification of hyperspectral images," *Neural Computing and Applications*, pp. 1–20, 2024.

[333] M. Ye, C. Ji, H. Chen, L. Lei, H. Lu, and Y. Qian, "Residual deep pca-based feature extraction for hyperspectral image classification," *Neural Computing and Applications*, vol. 32, pp. 14 287–14 300, 2020.

[334] Y. Li, M. Ye, Y. Qian, and Q. Qian, "Cross-domain hyperspectral image classification based on graph convolutional networks," in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2023, pp. 5974–5977.

[335] Z. Zhou, Y. Lei, B. Zhang, L. Liu, and Y. Liu, "Zegclip: Towards adapting clip for zero-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 175–11 185.

[336] Z. Wang, J. Li, T. Zhang, and S. Yuan, "Spectral–spatial discriminative broad graph convolution networks for hyperspectral image classification," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 3, pp. 1037–1051, 2023.

[337] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3d convolutional autoencoder for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6808–6820, 2019.

[338] Y. Ding, X. Zhao, Z. Zhang, W. Cai, and N. Yang, "Multiscale graph sample and aggregate network with context-aware learning for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4561–4572, 2021.

[339] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sensing Letters*, vol. 8, no. 9, pp. 839–848, 2017.