Hyperspectral Image Compression using Implicit Neural Representations

by

Shima Rezasoltani

A thesis submitted to the
School of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Ph.D. in **Computer Science**

Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada
July 2025

THESIS EXAMINATION INFORMATION

Submitted by: **Shima Rezasoltani**

PhD in **Computer Science**

Thesis title: HYPERSPECTRAL IMAGE COMPRESSION USING IMPLICIT NEURAL REPRESENTATIONS

An oral defense of this thesis took place on June 17, 2025 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Kenneth Wilson
Research Supervisor	Faisal Z. Qureshi
Research Co-supervisor	
Examining Committee Member	Kourosh Davoudi
Examining Committee Member	Jane Breen
Thesis Examiner	
University Examiner	Andrew Hogue
External Examiner	Boubakeur Boufama

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Hyperspectral images (HSI) capture the full electromagnetic spectrum for each pixel in a scene. They often hold hundreds of channels per pixel, providing significantly more information compared to a comparably sized RGB color image. As the cost of obtaining these images decreases, there is a need to create effective ways for storing, transferring, and interpreting hyperspectral data. In this thesis, we develop several neural compression-based methods for hyperspectral images. Our methodology relies on transforming hyperspectral images into implicit neural representations (INR), specifically neural functions that establish a correspondence between coordinates and features. We use a multilayer perception (MLP) network with sinusoidal activation functions that "learns" to map pixel locations to pixel spectrum for a given hyperspectral image. This representation thus acts as a compressed encoding of this image, and the original image is reconstructed by evaluating this network at each pixel location. In the other variation of using implicit neural representation to compress hyperspectral images, a sampling scheme is introduced to achieve better compression times while keeping decoding errors low. In our other method, instead of explicitly saving the weights of the implicit neural representation, the modulations that are applied to a base network that has been meta-learned are recorded. These modulations serve as a compressed coding for the hyperspectral image. An assessment of the proposed approach was conducted using four benchmarks: Indian Pines, Jasper Ridge, Pavia University, and Cuprite. The proposed method is evaluated against sixteen other schemes ((1) JPEG, (2) JPEG2000, (3) PCA-DCT, (4) PCA-JPEG2000, (5) MPEG, (6) X264, (7) X265, (8) PCA-X264, (9) PCA-X265, (10) FPCA-JPEG2000, (11) 3D-DCT, (12) 3D-DWT-SVR, (13) WSRC, (14) HEVC, (15) RPM, and (16) 3D-SPECK.) for hyperspectral image compression, and according to the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) metrics, the method developed in this study achieves state-of-the-art compression rates at low-bit rates. We also used a large hyperspectral image dataset, compressed it using our methods, and compared our results with JPEG and MPEG. Finally, we conducted task-aware hyperspectral image compression, in which regions are chosen according to a task, and hyperspectral images are compressed using our proposed method.

Keywords: hyperspectral image compression; implicit neural representations; task-aware compression; meta networks

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Shima Rezasoltani

STATEMENT OF CONTRIBUTIONS

Part of the work described in Chapter 4 has been published as:

S. Rezasoltani and F. Z. Qureshi, "Hyperspectral Image Compression Using Implicit Neural Representations, "2023 20th Conference on Robots and Vision (CRV), Montreal, QC, Canada, 2023, pp. 248-255, doi: 10.1109/CRV60082.2023.00039.

S. Rezasoltani and F. Z. Qureshi, "Hyperspectral Image Compression Using Sampling and Implicit Neural Representations," in IEEE Transactions on Geoscience and Remote Sensing, vol. 63, pp. 1-12, 2025, Art no. 5500712, doi: 10.1109/TGRS.2024.3509718.

Part of the work described in Chapter 5 has been published as:

Hyperspectral Image Compression Using Implicit Neural Representation and Meta-Learned-Based Network. Rezasoltani, S.; and Qureshi, F. Z. In Proc. 14th International Conference on Pattern Recognition Applications and Methods, pages 9pp, Porto, February 2025.

Part of the work described in Chapter 5 has been submitted as:

Meta-Learned Implicit Neural Representations for Scalable and Fast Hyperspectral Image Compression. Rezasoltani, S.; and Qureshi, F.Z. Lecture Notes in Computer Science, ICPRAM 2025 Selected Papers, Springer, pages 18pp (In review)

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Faisal Qureshi, for his guidance, encouragement, and support throughout my research journey. His invaluable insights, constructive feedback, and unwavering dedication have been instrumental in shaping this thesis.

I would also like to extend my appreciation to my lab colleagues for their invaluable support and collaboration. Their valuable input, discussions, and feedback have greatly enriched my research experience and enabled me to broaden my horizons.

Finally, I would like to thank my family for their support, and encouragement. Their constant encouragement and belief in me have been a constant source of motivation and inspiration, and I am forever grateful for their unwavering support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Hyperspectral images capture the electromagnetic spectrum per pixel, in contrast to color images, which typically record three values per pixel, or grayscale images, which record a single intensity value per pixel [Goetz et al., 1985]. As a result, each pixel in a hyperspectral image has tens or even hundreds of values that correspond to reflectance measurements made at different frequency bands. Consequently, compared to grayscale or color images, hyperspectral images present more opportunities for object detection, material identification, and scene analysis. It is not surprising that hyperspectral images have found widespread use in fields like remote sensing, biotechnology, crop analysis, environmental monitoring, food production, medical diagnosis, pharmaceutical industry, mining, and oil and gas exploration, among others.

The capture, storage, processing, and transmission costs of hyperspectral images continue to be high. Hyperspectral sensors are expensive. A typical hyperspectral image requires an order of magnitude more storage space than that needed to store a typical color image that has the same spatial resolution. Thus, there is significant interest in the community in developing efficient methods for capturing, storing, sending, and analyzing hyperspectral data.

1.1 Thesis Focus

This thesis examines the issue of hyperspectral image compression, demonstrating its crucial role in reducing storage and transmission costs by leveraging neural implicit representations. Specifically, the thesis introduces novel compression methods that optimize storage efficiency while preserving spectral fidelity, as validated through benchmark experiments.

1.2 Applications

Hyperspectral imaging (HSI) is a powerful technology that captures spatial and spectral information across hundreds of narrow, contiguous frequency bands. Unlike conventional imaging, which relies on broad spectral bands (e.g., RGB), HSI provides detailed spectral signatures that enable precise material identification, environmental monitoring, industrial inspection, and medical diagnostics [Kruse and Perry, 2009]. HSI allows us to combine spectral and spatial information, offering unparalleled capabilities in detecting, classifying, and mapping materials with high accuracy.

HSI's unique ability to capture rich spectral information across the electromagnetic spectrum—from the visible (VIS) and near-infrared (NIR) to the short-wave infrared (SWIR) and beyond—enables applications in agriculture, remote sensing, mineral exploration, healthcare, and security. This section explores the fundamental principles of hyperspectral imaging, highlighting its core features and real-world applications.

1.2.1 Agriculture & Farming

In precision agriculture, hyperspectral sensors on satellites or drones analyze crop health by detecting subtle spectral variations in nutrient levels, water stress, and disease presence [Lu et al., 2020]. Farmers can make data-driven decisions on fertilizer application, irrigation, and pest control, optimizing resources while increasing yield [Madani and McIsaac, 2021].

1.2.2 Mineral Exploration

Similarly, in mineral exploration, HSI enables the detection of mineral deposits and geological compositions by analyzing their unique spectral reflectance properties. Instead of relying on intrusive geological surveys and drilling, geologists can map mineral-rich areas remotely, optimizing resource extraction and reducing costs [Wolfe and Black, 2018].

1.2.3 Industrial Quality Control

In industrial applications, hyperspectral sensors enhance quality control in manufacturing by identifying contaminants, assessing product consistency, and detecting defects in pharmaceuticals, textiles, and food products. Even minor spectral deviations may indicate spoilage, chemical inconsistencies, or defects, ensuring high manufacturing standards [Kruse and Perry, 2009].

1.2.4 Environment Monitoring

Hyperspectral imaging plays a crucial role in environmental monitoring and land use classification. By analyzing spectral characteristics, HSI can distinguish between forests, water bodies, and urban areas, facilitating long-term studies on deforestation, urban expansion, and climate change [Sommer et al., 1998]. Additionally, hyperspectral sensors can detect oil spills, monitor water quality, and assess pollution levels by identifying spectral anomalies in water bodies [Bansod et al., 2018].

1.2.5 Security & Defense

In security and defense, hyperspectral imaging enables threat detection, surveillance, and hazardous material identification. Since explosives, concealed weapons, and toxic substances have unique spectral signatures, hyperspectral sensors can detect these materials even if disguised or hidden, improving border security and airport screening [Vishnu et al., 2013].

1.2.6 Vehicle Safety

In autonomous navigation, self-driving vehicles use hyperspectral imaging to classify road surfaces, lane markings, and obstacles. By distinguishing between pavement, ice, or water through spectral differences, hyperspectral imaging enhances safety in changing environmental conditions [Feng et al., 2019].

1.2.7 Biomedical & Healthcare

HSI has emerged as a valuable tool in medical diagnostics and surgical guidance. The ability to differentiate healthy and diseased tissues based on spectral properties enhances non-invasive disease detection and precision surgery [Madani and McIsaac, 2021].

In cancer diagnostics, hyperspectral imaging enables early-stage tumor detection by identifying subtle spectral differences between normal and malignant tissues [Fei, 2019]. Unlike traditional biopsy-based methods, hyperspectral imaging offers real-time, non-invasive analysis, aiding in early intervention and improved patient outcomes.

Furthermore, in wound assessment and infection monitoring, hyperspectral imaging provides insights into tissue oxygenation, blood flow, and healing progress. Studies have explored its feasibility in assessing burn injuries, diabetic ulcers, and skin disorders, offering objective and quantitative assessments for medical professionals [Madani and McIsaac, 2021].

1.2.8 Food Safety

Hyperspectral imaging is revolutionizing food quality assessment and forensic investigations. In food safety, HSI detects spoilage indicators, contaminants, and nutritional content in fruits, vegetables, and packaged products [Liu et al., 2017]. By analyzing spectral properties, manufacturers can ensure higher quality control and reduced food waste.

1.2.9 Crime Scene Forensics

In forensics, hyperspectral imaging aids in crime scene analysis, trace evidence detection, and fingerprint analysis. Since different materials (e.g., blood, drugs, fibers) reflect light uniquely, hyperspectral imaging provides detailed spectral differentiation that enhances forensic accuracy [Edelman et al., 2012]. This technology is increasingly used in law enforcement and counter-terrorism efforts to identify substances and reconstruct crime scenes with high precision.

Crane *et al*. [Crane et al., 2007] showcase the capability of infrared hyperspectral imaging (IR HSI) in detecting latent, untreated fingermarks on a variety of porous surfaces, including copier paper, cigarette butt paper, U.S. dollar bills, and postcards, as well as non-porous surfaces such as trash bags, soda cans, and tape [Crane et al., 2007]. Notably, fingermarks on the soda can, and black trash bag were distinctly visible when examining the intensity band image at 9842 nm, corresponding to the asymmetric O-C-C stretch in esters (see Fig. 1.1).

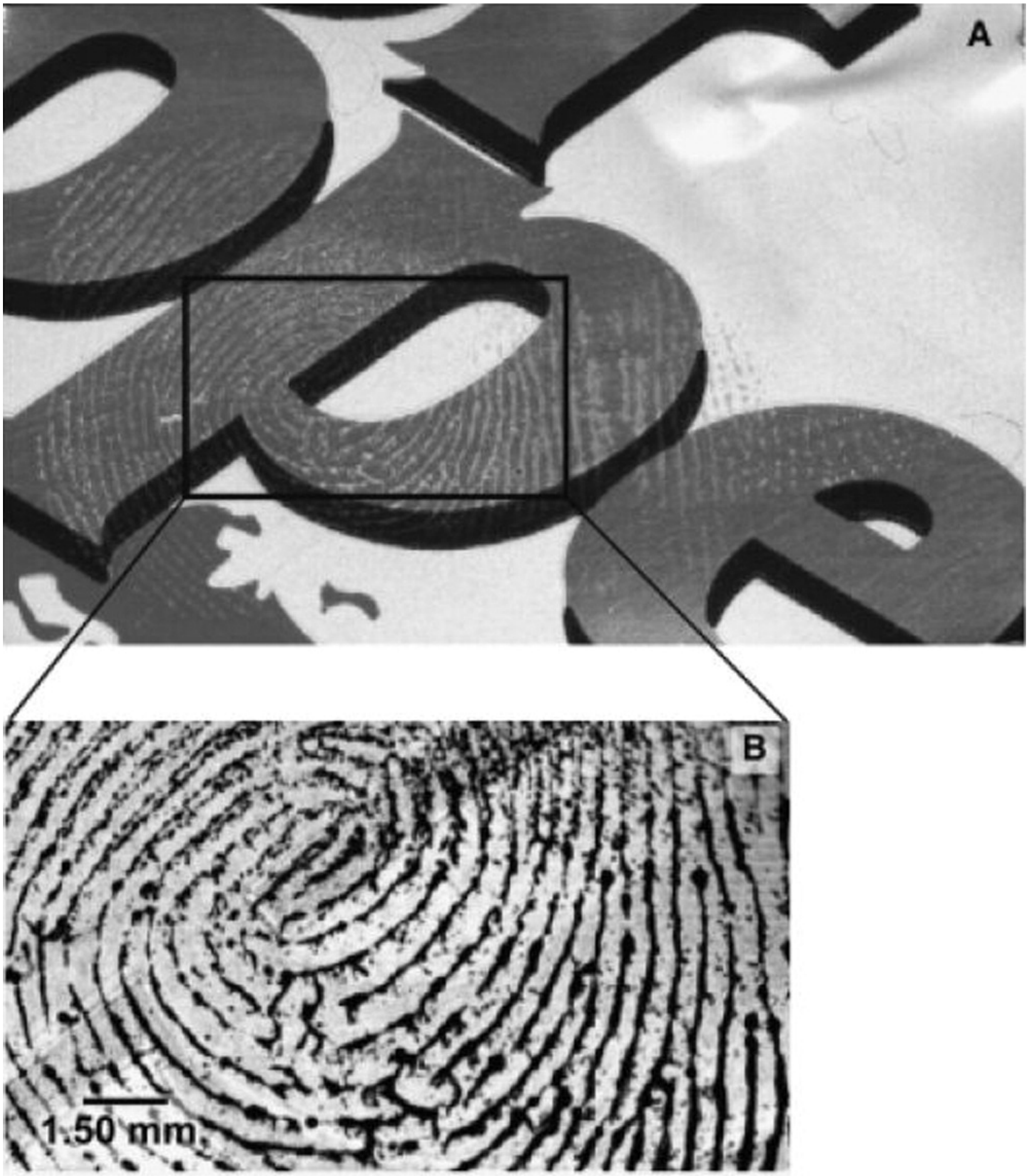


Figure 1.1: Cut and flattened Dr. Pepper soda can with a deposited fingermark. (A) The soda can be imaged using a document scanner. (B) An infrared image of the highlighted area was generated by mapping the band intensity at 9842 nm. Adapted from [Crane et al., 2007].

1.2.10 Future Applications

One of HSI's key strengths is its ability to capture data across a broad spectral range, from visible (400 to 700 nm) to near-infrared (700 to 1000 nm) and short-wave infrared (1000 to 2500 nm). Different spectral regions provide information on the biological, chemical, and physical properties of materials:

- Visible (400–700 nm): Captures data in the human-perceivable spectrum for easy interpretation;
- Near-infrared (700–1000 nm): Useful for detecting plant health, organic materials, and moisture content; and
- Short-wave infrared (1000–2500 nm): Provides information on mineral composition, industrial contaminants, and chemical compounds.

The ability to collect detailed spectral data across multiple wavelengths makes hyperspectral imaging highly adaptable for diverse industrial, scientific, and security applications. The area that investigates this makes hyperspectral capture, storage, and analysis offer exciting opportunities for knowledge generation.

1.3 Hyperspectral Image Compression

Figure 1.2 is a cube image taken by the AVIRIS satellite of the Jet Propulsion Laboratory (JPL) above Moffett Field in California. The false-color image on top of Figure 1.2 depicts a complex structure in the water and evaporation ponds to the right. The Moffett Field airport is also visible on the top of the cube.

Figure 1.3 illustrates the storage requirements for different image types: RGB, Multispectral, and Hyperspectral. As shown, hyperspectral images require substantially more storage space than RGB and multispectral images due to the vast number of

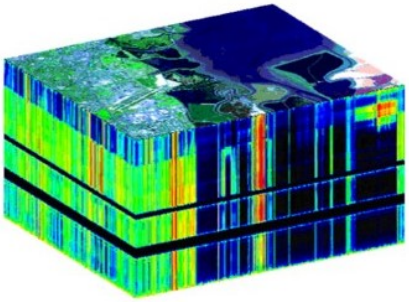


Figure 1.2: The JPL's AVIRIS hyperspectral data cube provided on a NASA ER-2 plane over Moffett Field. The top layer represents a pseudocolor visualization of the scene, while the stacked layers illustrate the multiple spectral bands that compose the hyperspectral image, capturing detailed spectral information across different wavelengths.

spectral bands captured in hyperspectral imaging. While an RGB image typically contains only three bands (red, green, and blue), hyperspectral images can have hundreds of bands, providing much richer spectral information but at the cost of increased data size. This makes efficient compression techniques particularly important for hyperspectral image storage and transmission. In comparison, multispectral images, which capture a moderate number of bands, require more storage than RGB images but significantly less than hyperspectral data. This Figure underscores the need for optimized compression methods for hyperspectral images to manage their large storage demands.

1.4 Thesis Contributions

Learning-based compression techniques have seen a recent rise in interest. To learn compact representations of the input signals, for instance, autoencoders [Hinton and Zemel, 1993] and rate-distortion autoencoders [Alemi et al., 2018, Balle' et al., 2017] have been employed. In this situation, the compressed representation of the input signal is provided by the network weights in conjunction with the signal signature, which is the latent representation in the case of autoencoders. Simultaneous studies

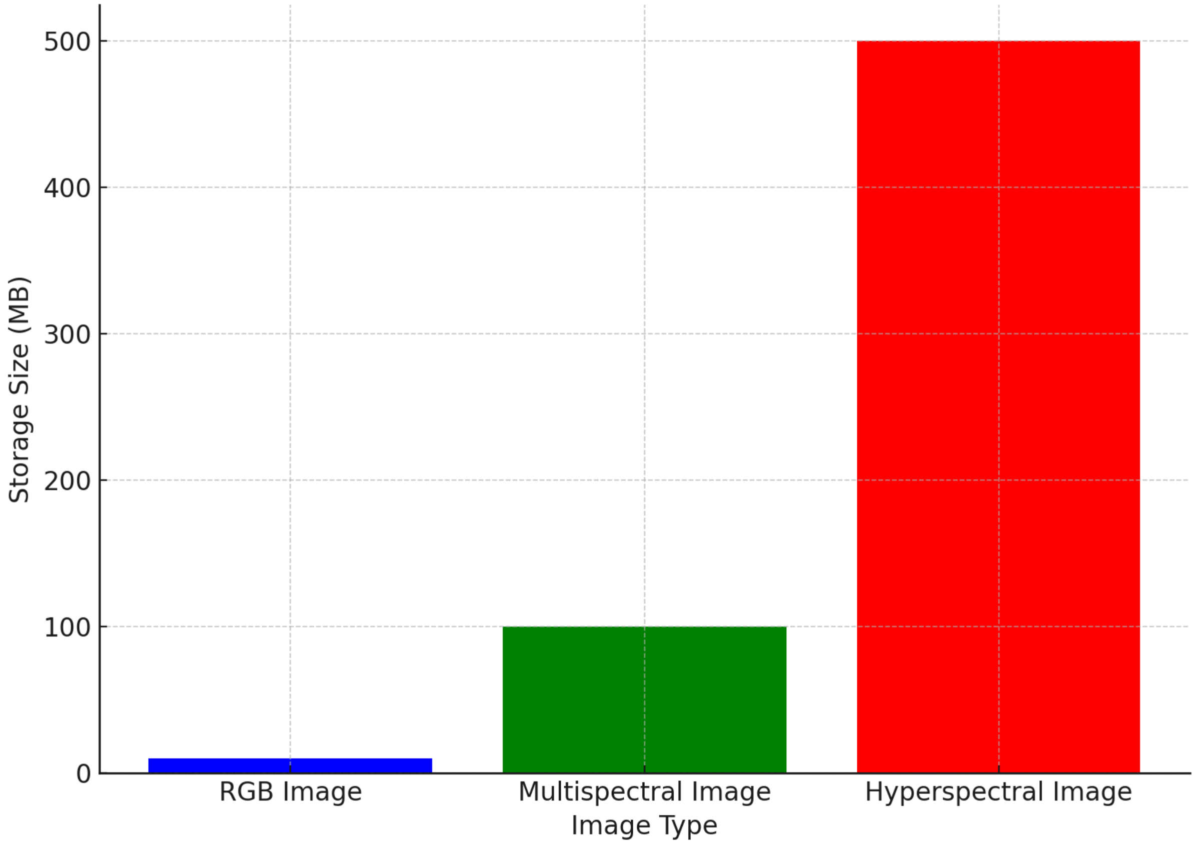


Figure 1.3: Comparison of storage requirements for different image types (RGB, Multispectral, and Hyperspectral). Hyperspectral images require significantly more storage space due to their high spectral resolution and large number of spectral bands, while RGB images use the least storage space as they contain only three bands.

are investigating the application of Implicit Neural Representations (INRs) in signal compression.

INRs are particularly well-suited to describe data that lives on an underlying grid. These are able to capture complex patterns and connections between the data without the need for explicit parameterization of the grid structure. Specifically, INRs learn a mapping between the grid coordinates and the related data values (e.g., a mapping between pixel location to its spectrum) [Strümpler et al., 2022]. Mathematically, consider a hyperspectral image $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$, where h, w, and c denote the height, width, and the number of channels, respectively. The goal is to learn a function $f_\Theta : (x, y) \mapsto \mathbf{I}[x, y]$, where (x, y) denote pixel coordinates and $\mathbf{I}[x, y] \in \mathbb{R}^c$ denotes the pixel spectrum. Θ, which serves as an encoding for the image $\mathbf{I}$, denotes the function parameters. The original image can be reconstructed given Θ by evaluating f_Θ at $[1, h] \times [1, w]$.

This thesis presents the first comprehensive study that explores the use of Implicit Neural Representations for hyperspectral image compression. While INRs have been previously applied to natural images, their potential in the context of high-dimensional spectral data has remained largely unexplored. This thesis not only demonstrates the feasibility of using INRs for encoding and reconstructing hyperspectral images, but also provides a systematic analysis of their performance under varying compression constraints. The proposed method introduces a novel perspective on hyperspectral image compression, positioning INRs as a viable and efficient alternative to both traditional and learning-based approaches. By leveraging the continuous nature of coordinate-based mappings, the method enables high-quality reconstruction from compact model parameters, significantly reducing the storage and transmission burden associated with hyperspectral data.

These contributions provide a comprehensive framework for hyperspectral image compression, addressing key challenges such as encoding efficiency, spectral fidelity, and task-aware optimization. By leveraging implicit neural representations, meta-learning, and adaptive compression strategies, this work offers practical and scalable solutions for reducing hyperspectral data volume while maintaining high reconstruction quality. The proposed approaches are evaluated across multiple benchmark datasets, demonstrating their effectiveness compared to conventional and learning-based methods. The following sections delve deeper into the theoretical foundations and practical benefits of using INRs for hyperspectral image compression.

1.4.1 Benefits of Using INRs

Using INRs to encode hyperspectral images offers the following advantages. (1) Flexibility: Implicit neural representations can accurately represent complex and irregular grids or surfaces. Contrary to explicit grid-based representations, which require the explicit definition of the grid structure, implicit representations can adjust to the data without imposing strict grid limitations. (2) Generalization ability: Implicit neural representations exhibit strong generalization capabilities to previously unknown data items. They have the ability to catch complex patterns in the data, enabling extrapolation beyond the recorded grid points. This is especially advantageous when working with data that is sparsely or irregularly collected. (3) Computational efficiency and scalability: Implicit representations can offer computational efficiency, particularly when dealing with high-dimensional data or huge grids. Instead of explicitly storing or computing values for each grid point, they can generate values on demand by evaluating neural networks. Additionally, implicit representations provide excellent scalability in high-dimensional spaces. The complexity of explicit grid-based approaches frequently increases dramatically as the dimensionality of the underlying grid increases. On the other hand, implicit representations are capable of properly managing data with a large number of dimensions. (4) Smoothness and continuity assumptions: Implicit neural representations have the ability to depict smooth and uninterrupted changes in the data accurately. Describing phenomena that demonstrate progressive variations throughout the grid is advantageous because neural networks naturally smooth out the representations they learn. Lastly, (5) learnable representations: Implicit representations can be learned directly from data. Neural networks can be trained using optimization approaches to accurately capture the underlying patterns in the grid-based data, enabling the identification of intricate relationships and features. In summary, implicit neural representations offer a powerful framework for encoding grid-based data, providing flexibility, strong generalization capabilities, computational efficiency, scalability, and the ability to capture smooth and continuous variations in the data.

1.4.2 Contribution 1: HSI Compression using INR

In Chapter 4, we investigate the use of INRs for hyperspectral image compression and show that it is possible to achieve high rates of compression while maintaining acceptable Peak Signal-to-Noise Ratio (PSNR) values. Figure 1.4 provides an overview of the proposed compression and decompression pipeline. We evaluate the proposed approach on four benchmarks (1) Indian Pines, (2) Jasper Ridge, (3) Pavia University, and (4) Cuprite—and show that at comparable bits-per-pixel-per-band values, our method achieves better PSNR values than those posted by three popular hyperspectral image compression schemes—(1) JPEG [Good et al., 1994, Qiao et al., 2014], (2) JPEG2000 [Du and Fowler, 2007], and (3) PCA-DCT [Nian et al., 2016]—at comparable bits-per-pixel-per-band values. The results confirm that our method achieves better PSNRs at low compression rates than those obtained by other methods.

Additionally, in the second part of this Chapter 4, we propose a sampling technique to speed up the encoding process and show that sampling achieves faster compression times while achieving PSNR values similar to those obtained when the image is encoded without sampling. In this part, We again evaluate the proposed approach on four benchmarks (1) Indian Pines, (2) Jasper Ridge, (3) Pavia University,

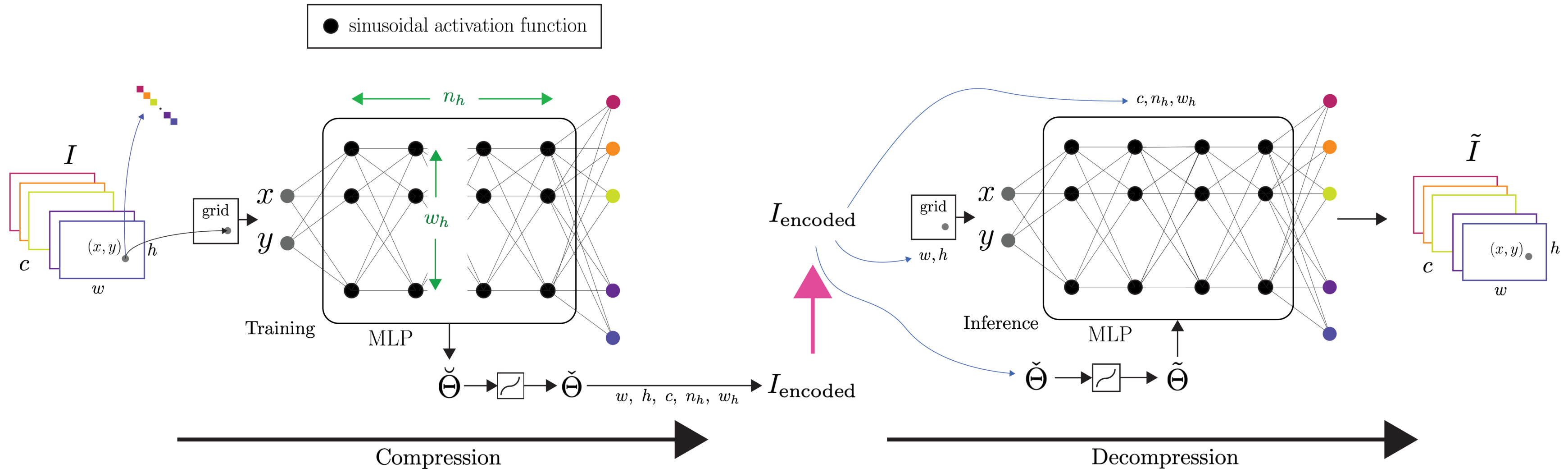


Figure 1.4: **Compression and Decompression Pipeline**. (left) An MLP with a periodic activation function is *trained* to map pixel locations to the pixel's spectral signature. (right) Once fitted, the MLP is used to reconstruct the hyperspectral image by performing *inference* at various pixel locations.

and (4) Cuprite. Specifically, we evaluate the proposed approach against (1) JPEG, (2) JPEG2000, (3) PCA-DCT, (4) PCA-JPEG2000 [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a] (5) MPEG [Le Gall, 1991], (6) X264 [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a], (7) X265 [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a], (8) PCA-X264 [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a], (9) PCA-X265 [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a], (10) FPCA-JPEG2000 [Mei et al., 2018], (11) 3D-DCT [Yadav and Nagmode, 2018], (12) 3D-DWT-SVR [Zikiou et al., 2020], (13) WSRC [Ouahioune et al., 2021], (14) HEVC [Sullivan et al., 2012], (15) RPM [Paul et al., 2016], and (16) 3D-SPECK [Ngadiran et al., 2010]. This list includes both the so-called classical approaches and the more recent learning-based methods.

1.4.3 Contribution 2: Meta-Learned Based Network

In another study in Chapter 5, Hyperspectral Image Compression using Implicit Neural Representation and Meta-Learned Based Network, we recognize and resolve a number of issues with our previous work [Rezasoltani and Qureshi, 2023a, Rezasoltani and Qureshi, 2023b] and develop and implement an approach for hyperspectral neural compression. More precisely, we address the following problems with the previous work: 1. Encoding is time-consuming: it can take an hour or more to compress a hyperspectral image, depending on the image size; 2. It does not exploit the shared structure: since each hyperspectral image is compressed separately, networks do not share any information between themselves. We deal with these problems by: 1. Reducing encoding time by more than two orders of magnitude through meta-learning, 2. Utilizing a base network to encode shared structures across images, applying modulations to adapt it to specific image data. A visual overview of the proposed method, showing how the parameters of the neural networks are stored as compressed images, is provided in Figure 1.5. Figure 1.6 also provides an overview of

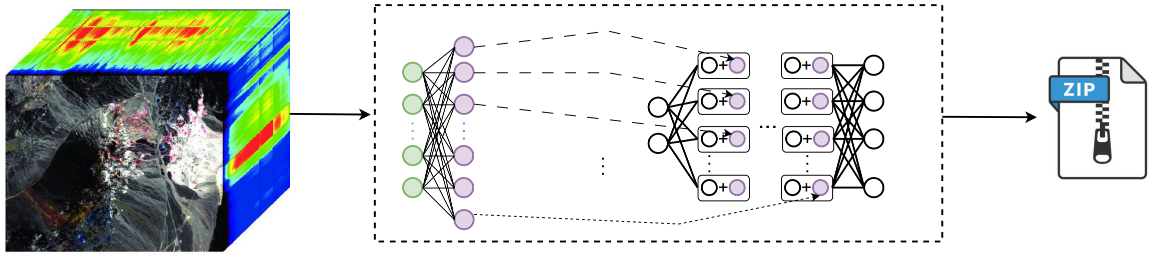


Figure 1.5: The proposed method represents hyperspectral images using implicit neural representations, where a neural network learns a function that maps image coordinates to pixel values. Instead of storing the full network weights, our approach compresses the image by recording a set of modulations applied to a meta-learned base network, resulting in a highly efficient and compact representation.

the proposed compression pipeline.

We assess the suggested methodology using four benchmarks. Our solution outperforms nine prominent hyperspectral image compression algorithms in terms of Peak Signal-to-Noise Ratio (PSNR) values for the Indian Pines, Jasper Ridge, Pavia University, and Cuprite datasets. Image compression techniques, namely JPEG, JPEG2000, PCA-DCT, MPEG, X264, X265, PCA-X264, PCA-X265, and PCA-JPEG2000 methods are compared in terms of their bits-per-pixel-per-band (bpppb) values.

In this study in Chapter 5, we also use a substantial dataset to evaluate the efficacy of our proposed meta-learned compression technique. This dataset, totaling 28.2 GB with dimensions of 4192 by 6708 pixels across 270 channels, allows us to rigorously test the scalability and effectiveness of our method in handling high-dimensional data. To effectively manage the complexity inherent in such large images, we implement a novel approach by partitioning each image into 7 by 7 grids, with dedicated networks to compress each row based on its unique properties. This design not only facilitates progressive compression but also ensures that image quality is preserved throughout the compression process. Our comparative analysis against existing methods, including JPEG and MPEG, highlights the competitive performance of our approach, as evidenced by the achieved Peak Signal-to-Noise Ratio (PSNR) and significant reductions in bits per pixel per band (bpppb). Ultimately, the flexibility and adaptability of

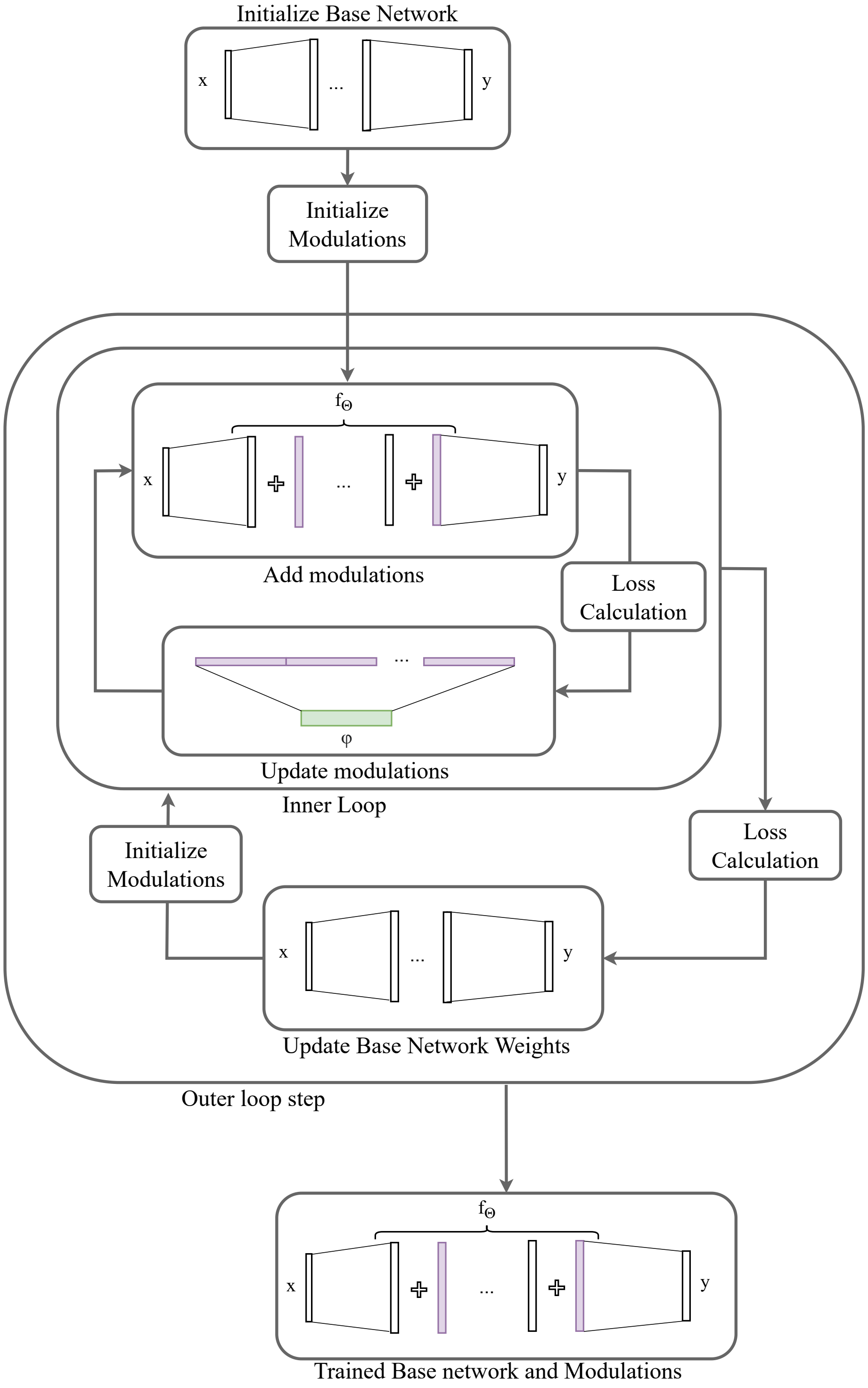


Figure 1.6: The compression pipeline used in the proposed method. The process begins by initializing the base network and modulations. Modulations are added and updated through inner loops, with loss calculation guiding the training process. The network weights are updated through outer loop iterations, resulting in a trained base network and modulations for efficient compression.

our meta-learned method make it a powerful tool for image compression, capable of accommodating various image sizes while delivering efficient compression outcomes.

1.4.4 Contribution 3: Differential Compression

In our final study, Chapter 6, we propose a task-aware hyperspectral image compression technique aimed at optimizing the balance between data reduction and the preservation of task-critical areas. The method utilizes the concept of region of interest (ROI) to prioritize compression, ensuring high fidelity in critical areas for tasks like classification or segmentation while applying more aggressive compression to non-essential regions to significantly reduce storage size.

Our method for task-aware hyperspectral image compression is illustrated in Figure 1.7. The method is divided into two main stages: object segmentation and task-aware compression. In the segmentation stage, K-means clustering is used in one part of the project, and a deep learning-based method is used in another part of it to identify regions of interest (ROI) based on the task requirements. Once the hyperspectral image is segmented, the compression stage is applied selectively. The segmented region undergoes a more detailed compression to preserve the critical spectral information, while the background regions are compressed using less strict criteria, focusing on minimizing the bitstream size.

The experimental results, showcased across five benchmark hyperspectral datasets, Indian Pines, Jasper Ridge, Pavia University, Cuprite, and a big dataset, exhibit the efficacy of the suggested methodology. The PSNR values in the area of interest indicate that the quality of task-sensitive regions is maintained, whilst the PSNR for the whole image demonstrates a satisfactory balance between global quality and compression efficiency. The bits per pixel per band (bpppb) and compressed image sizes exhibit substantial decreases in data volume, hence reinforcing the method's feasibility for practical applications when storage or transmission capacity is constrained.

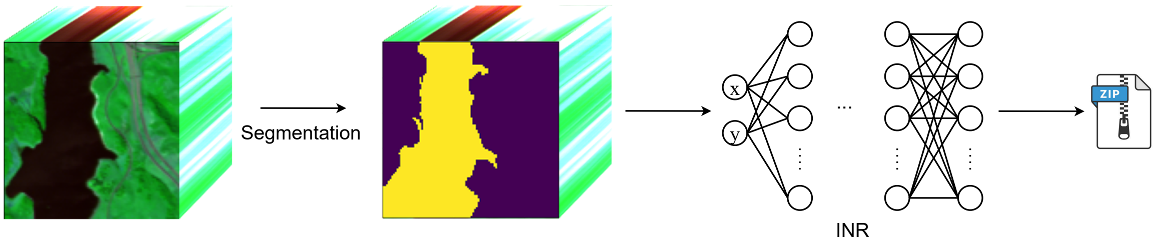


Figure 1.7: Overview of the proposed task-aware hyperspectral image compression method. The approach consists of two stages: segmentation and selective compression. In the segmentation stage, K-means clustering is used to identify regions of interest (ROI) based on the task requirements. During the compression stage, the segmented regions are compressed with higher fidelity to preserve critical spectral information, while background areas are compressed with less stringent criteria to optimize overall compression efficiency.

Task-aware hyperspectral image compression strategy is especially advantageous in situations where accurate task performance is prioritized over preserving overall image quality, as seen in remote sensing, video compression for streaming services, and hyperspectral analysis for object detection.

The findings demonstrate that the task-aware technique is an efficacious method for hyperspectral image compression, providing a compromise between elevated compression ratios and the retention of essential image information. This approach possesses significant potential for applications that necessitate quick storage and transfer while maintaining the requisite quality for subsequent activities.

1.5 Conclusion

In conclusion, this thesis makes the following key contributions to the field of hyperspectral image compression and analysis:

- Development of a novel hyperspectral image compression technique using Implicit Neural Representations (INRs): We propose an approach for hyperspectral image compression using implicit neural representations, where the data is compressed by learning a function that maps spatial coordinates to spectral values.

This method demonstrates a significant reduction in data size while maintaining competitive PSNR values.

- Proposal of a sampling-based hyperspectral image compression method: We propose a sampling technique that accelerates the encoding process while maintaining compression performance to address the challenge of long encoding times in hyperspectral image compression. This technique applies differential sampling rates to different regions of the image, allowing faster encoding with minimal impact on image quality.

- Meta-learning-based approach for hyperspectral image compression: We address the limitations of previous INR-based methods by introducing a meta-learned base network that reduces encoding time by two orders of magnitude. Our method uses a shared structure across images, which helps capture the common features of the dataset while adapting the model for image-specific information using modulation techniques. Furthermore, we validate the scalability of this approach by evaluating it on a large hyperspectral dataset, with a size of 28.2 GB and dimensions of 4192 by 6708 pixels across 270 channels, demonstrating its effectiveness across various hyperspectral benchmarks.

- Introduction of region-aware compression: This study introduces the concept of region-aware compression, where different parts of a hyperspectral image are compressed at different sampling rates depending on their significance. The results show that this method preserves critical regions with higher fidelity while reducing the overall data size, demonstrating its potential for real-world applications such as remote sensing and environmental monitoring.

- Introduction of task-aware hyperspectral image compression: We present a task-aware compression method that prioritizes regions of interest (ROI) based on the specific task at hand, such as classification or segmentation. This approach preserves the quality of task-critical areas while applying more aggressive compression to less relevant regions, resulting in significant storage savings without compromising task performance.

- Comprehensive evaluation and comparison against existing methods: The proposed methods are evaluated on four benchmark datasets (Indian Pines, Jasper Ridge, Pavia University, and Cuprite), and their performance is compared with a wide range of existing compression techniques, including JPEG, JPEG2000, PCA-DCT, PCA-JPEG2000, MPEG, X264, X265, PCA-X264, PCA-X265, FPCA-JPEG2000, 3D-DCT, 3D-DWT-SVR, WSRC, HEVC, RPM, and 3D-SPECK. Our methods consistently demonstrate superior performance in terms of compression efficiency and image quality.

Chapter 2

Research Landscape & Related Work

In this chapter, we discuss relevant literature to provide an overview of the hyperspectral image analysis research landscape. First, we discuss computational techniques developed in the community for the purposes of analyzing hyperspectral images. Next, we outline the challenges specific to capturing, processing, and managing hyperspectral data. Lastly, we review methods and techniques for hyperspectral image compression. Together, this set of topics sets the stage for the work presented in the following chapters.

2.1 Methods for Hyperspectral Image Analysis

Machine learning and deep learning techniques have become indispensable tools for processing and analyzing hyperspectral images, offering solutions for a wide range of applications. These methods are particularly effective in addressing the challenges posed by the high dimensionality and complex spectral-spatial information in hyperspectral data.

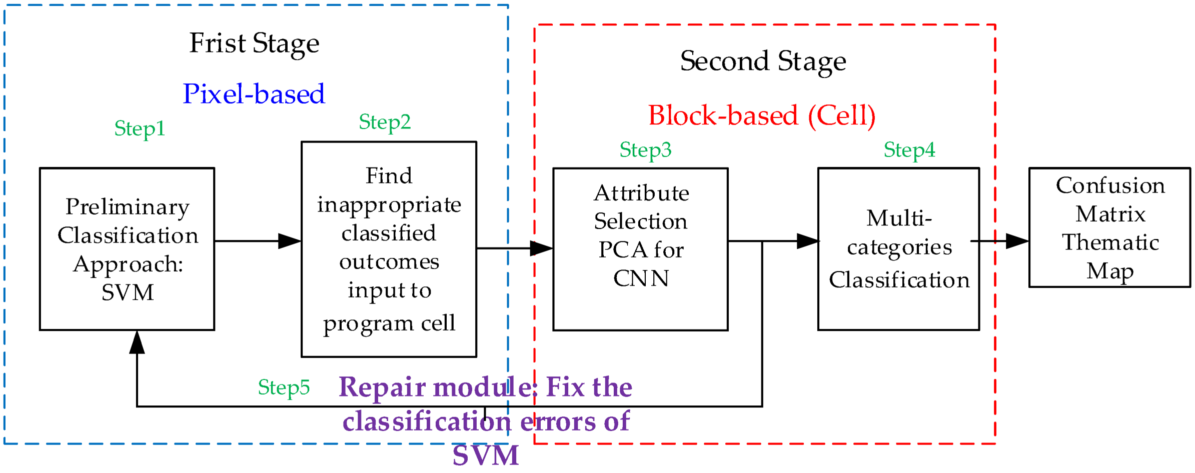

Frist Stage
Pixel-based
Step1
Step2
Preliminary Classification Approach: SVM
Find inappropriate classified outcomes input to program cell
Second Stage
Block-based (Cell)
Step3
Step4
Attribute Selection PCA for CNN
Multi-categories Classification
Confusion Matrix Thematic Map
Step5
Repair module: Fix the classification errors of SVM

Figure 2.1: Research steps from the study "An Innovative Intelligent System with Integrated CNN and SVM: Considering Various Crops through Hyperspectral Image Data." Adapted from [Wan et al., 2021].

2.1.1 Machine Learning Approaches

Traditional machine learning algorithms, such as Support Vector Machines (SVMs) and Random Forests, have been widely used for hyperspectral image classification, particularly in tasks like land cover mapping, agricultural monitoring, and urban area classification. These methods leverage hand-crafted spectral and spatial features but often struggle with high dimensional data and complex spectral relationships. For instance, Wan *et al.* [Wan et al., 2021] applied an SVM model to classify crop types in an agricultural region, achieving high accuracy by incorporating both spectral and spatial features extracted manually.

The research plan was structured into five distinct steps, as illustrated in Figure 2.1: (1) application of a support vector machine (SVM) prior to processing; (2) preparation of materials for principal component analysis (PCA) attribute selection; (3) refinement of detailed classification using a convolutional neural network (CNN); (4) establishment of multi-classification criteria and hierarchical rules; (5) implementation of a repair module to correct misclassification errors [Wan et al., 2021].

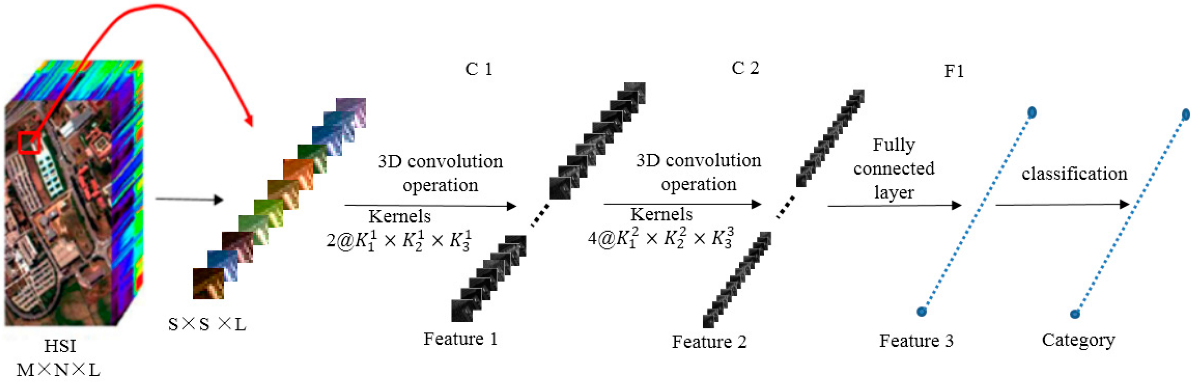


Figure 2.2: Diagram depicting the 3D-CNN-based framework for hyperspectral imagery classification. Adapted from [Li et al., 2017].

2.1.2 Deep Learning Approaches

However, with the rise of deep learning, methods such as Convolutional Neural Networks (CNNs) have demonstrated superior performance by automatically learning hierarchical features from hyperspectral images. Unlike traditional methods, CNNs can effectively capture spectral-spatial correlations without the need for extensive feature engineering. In a study by Li *et al*. [Li et al., 2017], a 3D-CNN model was used to analyze hyperspectral images for the classification of urban land cover. The model achieved state-of-the-art accuracy by simultaneously processing spectral and spatial information, outperforming traditional classifiers. To classify a pixel, the 3D-CNN model extracts relevant features from the pixel. Figure 2.2 illustrates the computational process [Li et al., 2017].

2.1.2.1 Object Detection

Object detection in hyperspectral images is another application where deep learning excels. For example, paper [Watanabe et al., 2019] implemented YOLO (You Only Look Once) [Redmon, 2016], a real-time object detection algorithm, to detect and classify objects in marine environments based on hyperspectral data. This approach allowed for the identification of different types of coral reefs and marine vegetation with high precision, even under challenging lighting conditions.

Figure 2.3 presents sample outcomes of underwater fish detection from video footage captured during a scuba diving session. The model successfully identified various fish species in different underwater conditions, including schools of mid-sized fish [Fig. 2.3(a)], small blue fish distinctly contrasted against rocky backgrounds [Fig. 2.3(b)], and fish swimming in low-visibility waters [Fig. 2.3(c)]. The detected fish are highlighted with pink bounding boxes. Notably, the detection model demonstrated high sensitivity by identifying small fish that might be challenging for human observers to notice, such as those positioned at the far left in Fig. 2.3(a) and those swimming above the rock formation in Fig. 2.3(c).

2.1.2.2 Anomaly Detection

Anomaly detection has also benefited from machine learning and deep learning techniques. In hyperspectral imaging, detecting anomalies often involves identifying materials or objects that differ significantly from the background. Researchers like [Fan et al., 2021] have employed Autoencoders for this purpose, where the network learns a compressed representation of normal data and detects anomalies based on reconstruction errors. This approach has been used in applications such as detecting chemical spills in environmental monitoring and identifying camouflaged objects in military reconnaissance.

2.1.2.3 Geological Surveys

In the field of target detection, deep learning techniques like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have shown promise in hyperspectral imaging, especially for tasks involving sequential spectral information. For instance, paper [Zhao et al., 2020] used an LSTM-based model for detecting specific mineral deposits in geological surveys, leveraging the sequential dependen-

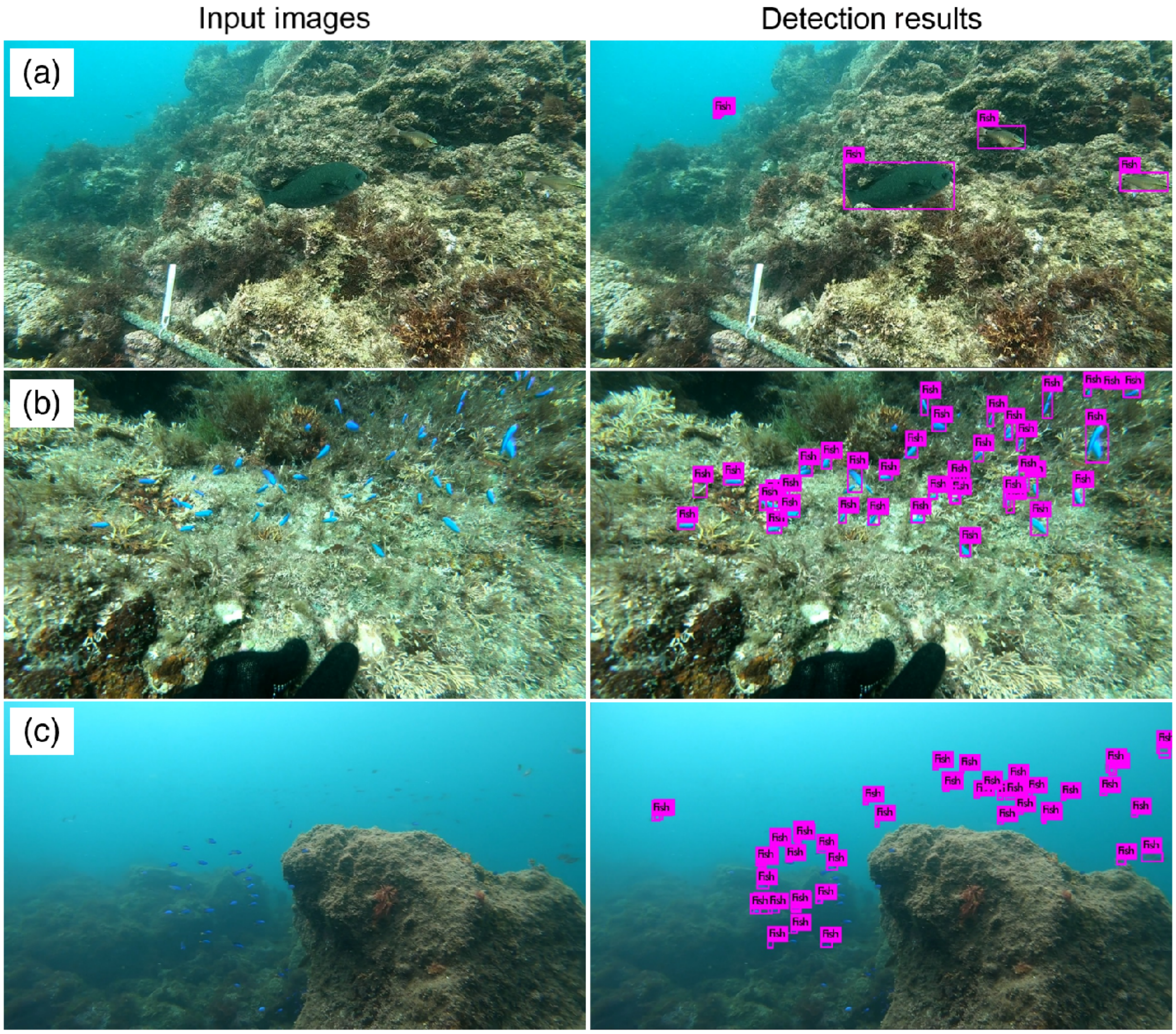


Figure 2.3: Example detection results of (a) schools of mid-sized fish, (b) small blue fish distinctly contrasted against the background, and (c) fish swimming in low-visibility water. The input video was captured during a scuba diving session. Adapted from [Watanabe et al., 2019].

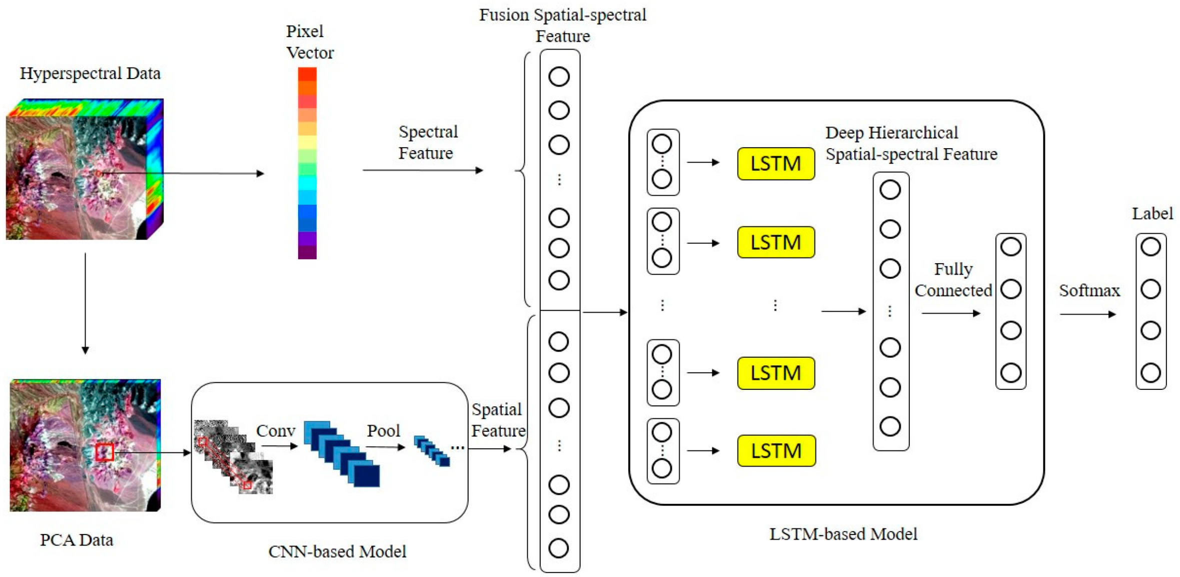


Figure 2.4: HSS-LSTM architecture for mineral identification using hyperspectral imagery. Adapted from [Zhao et al., 2020].

cies in hyperspectral data to improve detection accuracy.

2.1.2.4 Mineral Identification

Figure 2.4 illustrates the architecture of the HSS-LSTM model designed for mineral identification using hyperspectral imagery. The proposed model consists of three primary steps. First, local spatial features are extracted using a CNN-based model comprising multiple convolutional layers, max pooling layers, and a fully connected layer. Next, a fusion of spatial and spectral features is created by stacking the spectral information of a pixel with its corresponding local spatial features. Finally, an LSTM-based model is employed to extract deep hierarchical spatial-spectral features from the fused representation, followed by a softmax layer to perform mineral identification [Zhao et al., 2020].

2.1.2.5 Super-resolution

Deep learning techniques have made significant contributions to super-resolution in hyperspectral imaging. Super-resolution aims to enhance the spatial resolution of hyperspectral images by fusing information from different sources. For example, paper [Huang et al., 2019] utilized a Generative Adversarial Network (GAN) for hyperspectral image super-resolution, demonstrating improved spatial detail in remote sensing applications. GANs have also been used to fuse hyperspectral data with high-resolution multispectral images, resulting in sharper images with more detailed spectral information.

2.1.2.6 Change Detection

Change detection is another application where machine learning and deep learning techniques are commonly used. It involves identifying differences in a scene over time, which is particularly useful in environmental monitoring, disaster assessment, and urban development studies. Paper [Khan, 2022] employed a CNN-based Siamese network to detect changes in vegetation health using time-series hyperspectral data, enabling early detection of disease outbreaks in agricultural fields.

2.1.2.7 Spectral-Spatial Feature Extraction

Machine learning and deep learning have also found applications in spectral-spatial feature extraction, where the goal is to exploit spectral and spatial information for better analysis jointly. Techniques like Graph Convolutional Networks (GCNs) have been applied to hyperspectral data to model the relationships between different regions, improving classification performance by incorporating contextual information. In paper [Hong et al., 2020], GCNs were used for classifying various land cover types in a hyperspectral image of an urban area, achieving enhanced accuracy compared to traditional methods. GCNs [Hong et al., 2020] have gained significant attention as an

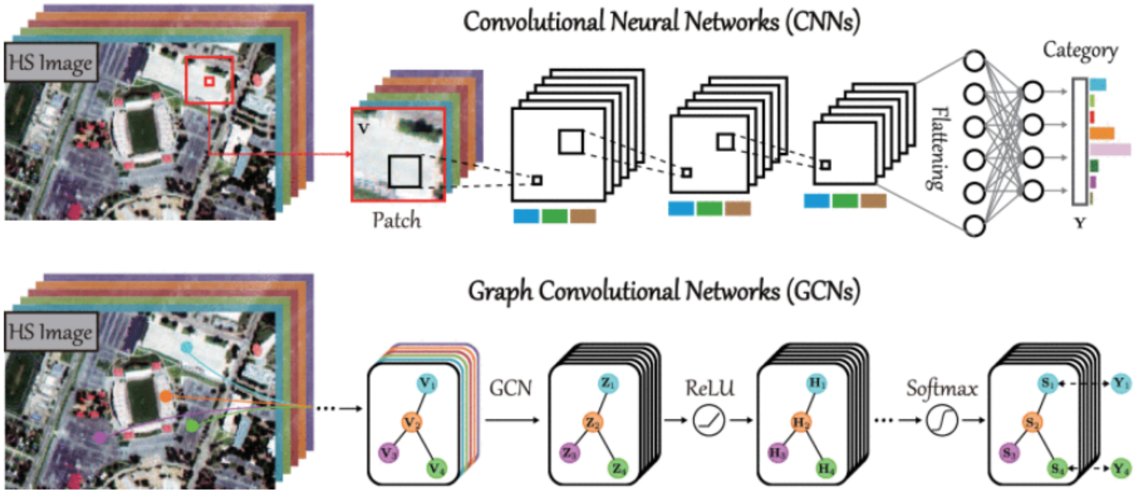

Convolutional Neural Networks (CNNs)
HS Image
Patch
V
Flattening
Category
Y
Graph Convolutional Networks (GCNs)
HS Image
GCN
ReLU
Softmax

Figure 2.5: Comparison of CNN and GCN architectures for hyperspectral image classification. In the GCN framework, V, Z, H, S, and Y represent vertices, hidden representations obtained from the GCN layer, hidden representations from the ReLU layer, hidden representations from the softmax layer, and labels, respectively. Adapted from [Hong et al., 2020].

emerging network architecture capable of effectively handling graph-structured data by capturing relationships between samples (or vertices). Unlike Convolutional Neural Networks (CNNs), which primarily focus on local spatial features, GCNs can naturally model long-range spatial dependencies in hyperspectral images, as illustrated in Figure 2.5 [Hong et al., 2020].

2.1.2.8 Data Augmentation

Data augmentation techniques powered by deep learning are increasingly being used to enhance hyperspectral imaging datasets. Techniques such as adversarial training and neural style transfer have been employed to artificially generate new training samples, thus improving the robustness of machine learning models. For example, paper [Nalepa et al., 2019] utilized adversarial training to generate realistic hyperspectral samples for training a deep learning model, leading to improved generalization on real-world test data.

2.1.2.9 Fractional Abundance Estimation

Fractional Abundance Estimation techniques extend traditional unmixing approaches by considering the uncertainties in abundance estimation. Methods like Bayesian Unmixing incorporate prior knowledge and probabilistic models to better estimate the abundance of materials, especially in environments with high noise levels. For example, paper [Duan et al., 2023] utilized Bayesian unmixing for monitoring oil spill detection, achieving robust performance even when the hyperspectral data were affected by significant background noise.

2.1.2.10 Spectral Unmixing

Spectral unmixing is a fundamental task in hyperspectral imaging that aims to decompose the mixed spectral signal recorded at each pixel into a set of pure spectral components (endmembers) and their corresponding proportions (abundances). Since real-world scenes often contain mixed pixels, especially in remote sensing applications, spectral unmixing is crucial for accurately identifying materials and understanding their distribution.

Traditional methods rely on either blind source separation or physics-based models, which often struggle with spectral variability and require labeled data. The work by Mantripragada and Qureshi [Mantripragada and Qureshi, 2024] introduces a data-driven approach leveraging a Latent Dirichlet Variational Autoencoder (LDVAE) to model abundance distributions. The proposed model assumes that abundance vectors follow a Dirichlet distribution, ensuring the necessary non-negativity and sum-to-one constraints, while endmembers are modeled as multivariate normal distributions. The LDVAE framework jointly performs endmember extraction and abundance estimation within a probabilistic deep learning paradigm. Notably, the model demonstrates strong generalization capabilities through a transfer learning paradigm, where it is pre-trained on synthetic data generated from the United States Geological Survey (USGS) Spectral Library and then applied to real-world datasets like Cuprite, Urban HYDICE, and Samson. Experimental results confirm that LDVAE achieves state-of-the-art performance in spectral reconstruction, abundance estimation, and endmember extraction. The study highlights the model's applicability to scenarios where labeled data is scarce and suggests future extensions incorporating spatial information for improved hyperspectral unmixing [Mantripragada and Qureshi, 2024].

One popular spectral unmixing method is Linear Spectral Unmixing (LSU), which assumes that the mixed spectral signature at a pixel is a linear combination of the end member spectra. LSU has been widely used in applications such as mineral mapping and soil analysis. For instance, paper [Kopackova and Hladikova, 2014] applied LSU to hyperspectral data collected over a mining site to estimate the abundance of various minerals, enabling geologists to assess ore quality and distribution effectively. The general processing framework is illustrated in Figure 2.6, where image endmembers are extracted from "water-only" pixels and subsequently utilized for further image analysis. To identify these pure spectral signatures, a methodology combining Minimum Noise Fraction (MNF) transformation [Joseph, 1994, Green et al., 1988] and Pixel Purity Index (PPI) [Boardman, 1995] was applied. This approach involves spectral and spatial dimensionality reduction to isolate spectrally distinct endmembers, assumed to represent the purest pixels within the dataset. Scatterplots of MNF-transformed bands were analyzed to examine data distribution within the feature space and investigate spectral mixing patterns. Spectral endmembers were then identified through scatterplot analysis across various MNF band combinations, following methodologies similar to those in [Adams, 1993, Rainey et al., 2003]. The identified endmembers were mapped back to their corresponding image pixels, facilitating further analysis. This processing technique is particularly useful for hyperspectral water analysis, en-

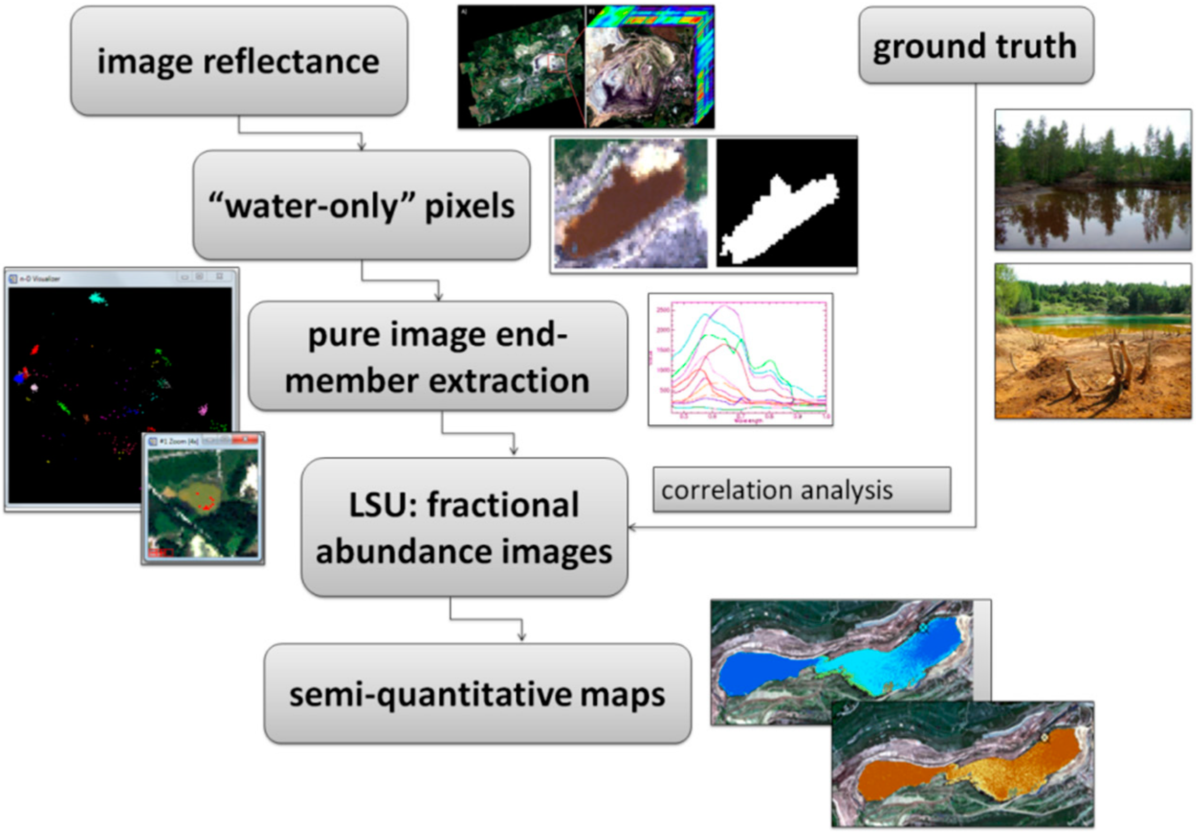


Figure 2.6: Processing workflow for HyMap image reflectance data and its resampled version at WV2 spectral resolution. Adapted from [Kopackova and Hladikova, 2014].

abling the characterization of different water types based on their unique chemical compositions and physical properties.

Paper [Ren et al., 2022] applied NMF to hyperspectral data from a mining site to unmix different minerals, providing insights into the composition of the ore deposits. Sparse Autoencoders have also been employed to improve the unmixing results by learning a compact representation of the spectral signatures.

Another widely used approach is Nonlinear Spectral Unmixing, which accounts for the complex interactions between materials, especially in cases where light scattering and other nonlinear effects are significant. Nonlinear unmixing models, such as Kernel-based methods and Artificial Neural Networks (ANNs), have been employed in fields like agricultural monitoring. For example, paper [Yang and Wang, 2018] utilized a kernel-based nonlinear unmixing method to detect the abundance of different crop types in agricultural fields, improving the estimation accuracy of mixed pixels compared to linear models. Sparse Unmixing techniques exploit the sparsity of hyperspectral data by assuming that only a few endmembers are active within a given pixel. This approach has been used effectively in applications such as urban land cover classification, where the number of materials in a single pixel is limited. For instance, paper [Zhang et al., 2016] applied Sparse Regression Unmixing to hyperspectral data of an urban area, identifying materials like concrete, asphalt, and vegetation with higher precision.

Another advanced spectral unmixing technique is Blind Source Separation (BSS), where both the endmembers and their abundances are unknown and need to be estimated directly from the data. Methods like Independent Component Analysis (ICA) and Non-negative Matrix Factorization (NMF) are commonly used for BSS. In a study by [Feretb et al., 2019], NMF was applied to hyperspectral images of a coral reef to separate different underwater species and substrates, providing insights into the reef's biodiversity and health. Incorporating spatial information can significantly improve unmixing performance, leading to the development of Spectral-Spatial Unmixing techniques. These methods combine spectral unmixing with spatial regularization or segmentation to exploit the spatial continuity of end members. For example, [Welikanna, 2008] proposed a Markov Random Field (MRF)-based approach that uses spatial constraints during the unmixing process to improve the abundance estimation of materials in an urban setting, reducing noise and enhancing the detection of small features.

Deep learning has also made significant inroads in spectral unmixing, with techniques like Autoencoders and Convolutional Neural Networks (CNNs) being applied for Deep Unmixing. For instance, paper [Sellami and Tabbone, 2022] used a Deep Autoencoder Network to learn a low-dimensional representation of hyperspectral data, which facilitated the extraction of endmembers and their corresponding abundances.

This approach was used in a study of forest canopy monitoring, where it provided more accurate estimates of different tree species than traditional unmixing methods.

Collaborative Unmixing is another advanced technique that uses information from multiple related hyperspectral images to improve unmixing accuracy. This approach is particularly useful in applications such as change detection and multi-temporal analysis. Paper [Rizkinia and Okuda, 2017] employed a Joint Sparse Unmixing technique to analyze hyperspectral images collected over a period of time in a mining area, effectively identifying changes in mineral composition caused by excavation activities.

2.2 Challenges

Despite its numerous advantages, hyperspectral imaging (HSI) has several challenges that limit its widespread adoption and practical implementation.

2.2.1 Data Volume & Computational Complexity

One of the most significant challenges in hyperspectral imaging (HSI) is the sheer volume of data generated. Unlike traditional imaging systems, which store only a few values (e.g., RGB) for each pixel, HSI collects hundreds of spectral bands per pixel, resulting in a hyperspectral data cube that is both high-dimensional and large in size. For instance, even a small hyperspectral image with modest spatial resolution can contain gigabytes of data, while large-scale remote sensing datasets captured by satellites can easily reach terabytes or petabytes. This data explosion not only imposes high demands on storage capacity but also presents unique challenges in terms of data management and processing [Nhaila et al., 2014].

The complexity of hyperspectral data lies in its multidimensional nature, where each pixel represents a point in both spatial and spectral space. This makes processing and analyzing the data computationally intensive. Traditional image processing techniques may face challenges when applied to hyperspectral datasets due to the high dimensionality and spectral dependencies. Specialized algorithms and tools—such as dimensionality reduction methods, machine learning, and deep learning models—are required to extract meaningful insights while preserving spectral integrity. However, these algorithms demand significant computational resources such as high-performance GPUs or cloud-based infrastructure, adding further cost and complexity [Li et al., 2022].

2.2.2 Real-Time Processing & Latency Issues

Real-time processing is especially challenging. In applications like precision agriculture, autonomous navigation, and environmental monitoring, timely insights are essential. However, processing the entire hyperspectral data cube in real time is difficult due to the need to analyze hundreds of bands for each pixel. Without optimized algorithms and efficient data pipelines, the latency in data analysis can hinder decision-making and reduce the utility of the system [Adão et al., 2017].

Moreover, storage solutions must accommodate not only the volume of raw data but also the processed outputs, intermediate results, and metadata. Efficient data compression techniques—both lossless and lossy—are essential to reduce the storage burden while retaining critical information. However, hyperspectral data compression itself is a research challenge, as preserving the spectral and spatial relationships during compression is non-trivial.

2.2.3 High-Cost of Hyperspectral Sensors

Another significant challenge in hyperspectral imaging lies in the high cost of the sensors and systems, which limits their accessibility for smaller industries, research labs, and individual users. Hyperspectral sensors are complex instruments that require precise components, including optical systems, diffraction gratings, filters, and detectors capable of capturing hundreds of narrow spectral bands with high sensitivity.

The cost of these specialized sensors can run into tens of thousands to hundreds of thousands of dollars, making them suitable primarily for large-scale research projects, government initiatives, or well-funded industries such as space exploration, defense, and precision agriculture. Additionally, the operational costs—including maintenance, calibration, data storage, and the expertise required to interpret the data—further increase the financial burden. This high price point hinders wider adoption of HSI technology, especially in small-scale industries, startups, and developing regions [Al-Hourani et al., 2023].

2.2.4 Sensor Calibration & Standardization

Precise calibration is essential for hyperspectral imaging systems to produce accurate, reliable, and reproducible results. Each sensor needs to be carefully calibrated to account for sensor-specific distortions, such as variations in sensitivity across wavelengths, pixel-to-pixel inconsistencies, and optical misalignments. Calibration involves using standardized light sources and reference materials to ensure that the spectral data accurately reflects real-world conditions. Any deviation in calibration can introduce artifacts or errors in the collected data, reducing the reliability of material identification or classification tasks. Maintaining calibration across multiple sensors or over extended periods also requires expertise and specialized equipment, which adds to the operational complexity and costs [Cruz-Guerrero et al., 2022].

2.2.5 Sensor Sensitivity to Environmental Conditions

The data acquisition process in HSI is highly sensitive to environmental conditions, which can degrade data quality if not properly managed. Lighting conditions, for example, play a crucial role in ensuring consistent and meaningful spectral measurements. Changes in illumination—whether from shadows, cloud cover, or artificial lighting variations—can alter the reflectance properties of objects, leading to inaccurate spectral signatures. This makes outdoor applications, such as remote sensing or agriculture, particularly challenging, as the imaging system must be carefully synchronized with natural light conditions or employ sophisticated algorithms to correct for illumination variability.

Atmospheric interference is another factor that can introduce noise in the data, especially for airborne and satellite-based HSI systems. Water vapor, dust, and other atmospheric particles can absorb or scatter specific wavelengths, affecting the integrity of the spectral data. Atmospheric correction algorithms are often required to mitigate these effects, but developing and applying these corrections can be computationally demanding and prone to errors [Katkovsky et al., 2018].

Movement during data acquisition is also a concern, especially when using hyperspectral sensors mounted on drones, satellites, or vehicles. Motion artifacts, misalignments, or vibrations can introduce spatial distortions or blurring in the hyperspectral data cube. In high-speed applications—such as drones surveying agricultural fields—specialized sensors with stabilization mechanisms or advanced image alignment algorithms are needed to compensate for motion-related errors.

These environmental challenges demand careful planning during data collection, including the use of standardized acquisition protocols, environmental sensors, and real-time corrections. However, this increases the complexity of operating HSI systems, making it more challenging for industries or researchers who lack the necessary technical expertise [Katkovsky et al., 2018]. The combination of high costs, calibration requirements, and sensitivity to environmental factors makes HSI systems less accessible for small businesses, startups, and individual researchers. This is a significant barrier to adoption in industries where profit margins are narrow or where the need for spectral imaging is occasional rather than continuous. Additionally, industries or academic institutions in developing regions may find it difficult to invest in hyperspectral technology due to budget constraints.

Efforts to overcome these challenges include the development of more affordable and portable hyperspectral sensors, such as miniaturized sensors for drones or smartphones. Advances in AI-driven calibration and correction algorithms are also helping to reduce the need for manual intervention, making the technology more user-friendly. Nevertheless, cost reduction, robust calibration solutions, and automated environmental correction tools remain key areas of research to enable broader adoption of HSI technology [Adão et al., 2017].

2.2.6 Visualization & Interpretation of Hyperspectral Data

Interpreting hyperspectral data is inherently challenging and labor-intensive due to the large volume of high-dimensional information it contains. Each pixel in a hyperspectral image carries a complete spectral signature, resulting in datasets that are significantly more complex than those from traditional imaging systems. As a result, extracting meaningful insights from hyperspectral data requires advanced techniques in data analysis, machine learning, signal processing, and domain-specific knowledge. Simply visualizing the data cube or identifying patterns using conventional tools is insufficient, making hyperspectral data analysis a specialized field with a steep learning curve [Vali et al., 2024].

2.2.6.1 Curse of Dimensionality

A typical hyperspectral data cube contains hundreds of spectral bands, resulting in a dataset with high dimensionality. While this abundance of information allows for precise material identification, it also introduces the curse of dimensionality, a phenomenon where the increased number of features (spectral bands) complicates data analysis. As the dimensionality grows, it becomes more difficult to identify meaningful patterns, increasing the risk of overfitting in machine learning models and making it challenging to design effective algorithms for classification, segmentation, or anomaly detection.

2.2.6.2 Dimensionality Reduction

One common approach to dimensionality reduction in hyperspectral imaging is Principal Component Analysis (PCA). PCA has been widely used for dimensionality reduction in hyperspectral image processing due to its ability to capture the most significant spectral variations while reducing data redundancy.

In paper [Mantripragada et al., 2022], Mantripragada *et al*. explore PCA's effectiveness in hyperspectral image compression, leveraging its strong energy compaction properties to reduce spectral dimensionality before applying advanced implicit neural representation (INR) models. The study highlights how PCA can serve as a preprocessing step to enhance the efficiency of INR-based compression by reducing input dimensionality while preserving essential spectral features. Additionally, the work demonstrates that by combining PCA with neural representations, hyperspectral image compression can achieve improved performance in terms of reconstruction accuracy and storage efficiency. The findings underscore the relevance of hybrid approaches that integrate traditional statistical techniques like PCA with modern deep learning-based methods for optimizing hyperspectral data compression and analysis.

In another paper [Tsai et al., 2007], authors applied PCA to hyperspectral data for vegetation classification, achieving significant data compression while maintaining the accuracy of the classification. However, PCA's linear nature may not capture complex nonlinear relationships in the data, prompting the use of alternative techniques. The process of spectrally segmented PCA for plant target detection is divided into multiple phases, as depicted in Figure 2.7 [Tsai et al., 2007].

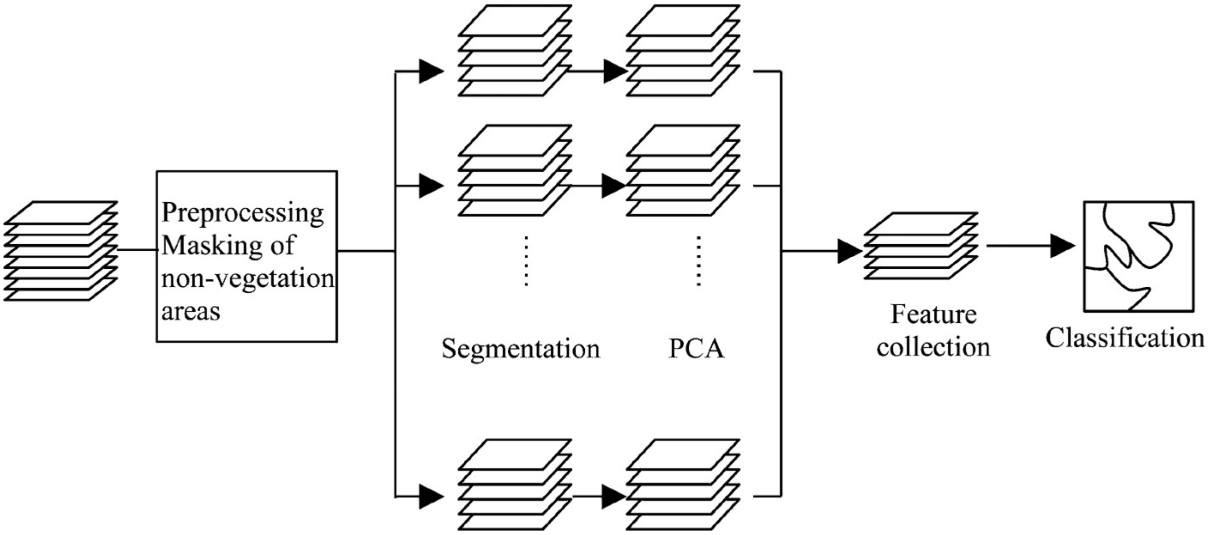


Figure 2.7: Overall workflow of spectrally segmented PCA. Adapted from [Tsai et al., 2007].

2.2.6.3 Independent Component Analysis (ICA)

Another effective method is Independent Component Analysis (ICA), which separates mixed signals into statistically independent components. Researchers like [Zhao et al., 2012] have utilized ICA for anomaly detection in hyperspectral images, such as identifying hidden objects in military applications or detecting geological features in remote sensing. ICA's ability to isolate independent sources from complex data allows for enhanced interpretation of hyperspectral images.

2.2.6.4 Nonlinear Dimensionality Reduction Techniques

Nonlinear dimensionality reduction techniques, such as t-distributed Stochastic Neighbor Embedding (t-SNE) and Isometric Mapping (Isomap), have also been explored. For instance, [Zhang et al., 2018] applied t-SNE to reduce the dimensionality of hyperspectral data from airborne sensors, facilitating the visualization of high-dimensional data clusters associated with different land cover types. Similarly, Isomap has been used to enhance feature extraction for mineral mapping in geological studies by capturing the manifold structure of hyperspectral data.

2.2.6.5 Autoencoders for Dimensionality Reduction

Autoencoders, a type of neural network, are increasingly popular for dimensionality reduction in hyperspectral imaging. These networks learn a compressed representation of the data through a bottleneck layer. For example, paper [Guerri et al., 2024] implemented a deep autoencoder to reduce the dimensionality of hyperspectral images in agricultural monitoring, allowing for the detection of plant stress with fewer spectral bands. The reconstruction quality of autoencoders ensures that important features are retained even after significant data compression.

A recent approach involves Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). Researchers such as [Chamain et al., 2022] have employed VAEs to compress hyperspectral data while simultaneously performing image restoration, showing that this technique can reduce noise and improve image quality. GANs have been explored for hyperspectral image super-resolution and denoising, with dimensionality reduction being a natural by-product of the generative modeling process.

Traditional classification approaches typically involve reconstructing images at the decoder before performing inference, as illustrated in Fig. 2.8(a) [Chamain et al., 2022]. However, this reconstruction step can introduce inefficiencies, particularly in cloud-based AI applications. Two key observations highlight these limitations. First, classification accuracy tends to degrade at high compression ratios when images are compressed using codecs optimized for rate-distortion performance, which is primarily designed for visualization rather than classification tasks [Chamain et al., 2022]. Second, performing inference on images compressed with standard codecs, such as JPEG2000, often reduces inference speed and accuracy compared to end-to-end joint classification and compression frameworks that bypass explicit image reconstruction [Chamain et al., 2022]. To address these challenges, the proposed VAE-based classification approach eliminates the reconstruction step, enabling direct classification from

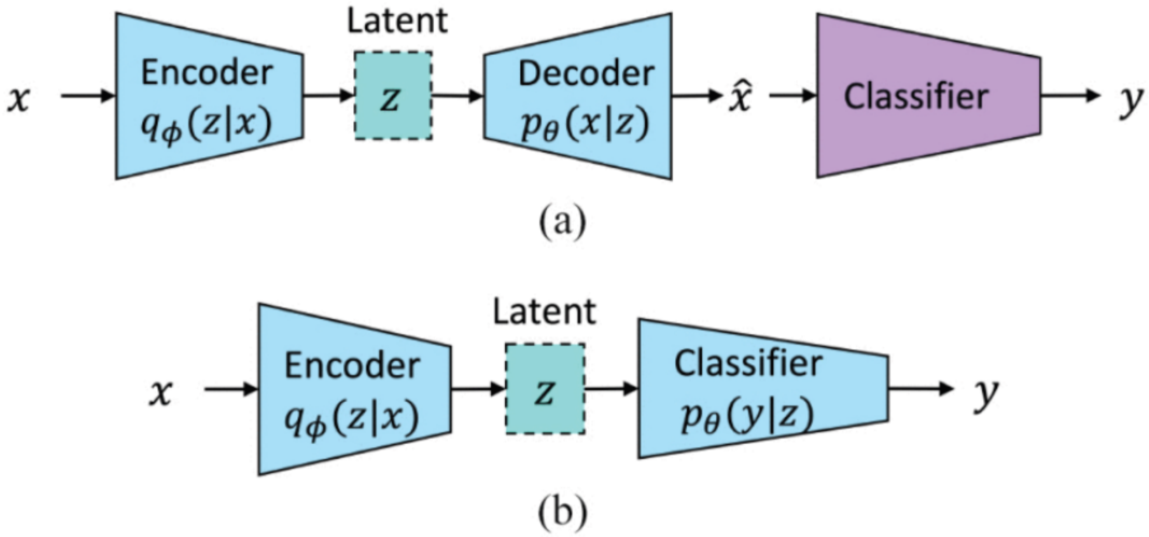


Figure 2.8: (a) VAE-based reconstruction before classification: The encoder maps the input image x to a low-dimensional latent representation z, from which the decoder reconstructs the image $\hat{x}$. (b) Proposed joint compression and classification: Instead of reconstructing the image, the classifier directly predicts the class label y from the latent representation z. Adapted from [Chamain et al., 2022].

the latent representation. This method naturally supports end-to-end optimization for improved rate-accuracy performance, as shown in Fig. 2.8(b).

2.2.6.6 Sparse Coding for Hyperspectral Analysis

Sparse coding is another effective strategy, where hyperspectral images are represented as a linear combination of a few basis functions. This technique has been applied to target detection in hyperspectral data. For example, paper [Yousefi et al., 2018] utilized sparse coding to detect specific minerals in airborne hyperspectral surveys, achieving accurate identification with a reduced number of spectral features.

2.2.6.7 Manifold Learning for Dimensionality Reduction

Furthermore, dimensionality reduction can be achieved using Manifold Learning, where the focus is on finding a lower-dimensional representation that preserves the intrinsic geometry of the data. In hyperspectral imaging, manifold learning techniques such as Locally Linear Embedding (LLE) have been applied to tasks like urban area classification and anomaly detection. Studies by [Iliopoulos et al., 2015] demonstrated that LLE effectively uncovers the underlying structure in high-dimensional hyperspectral data, leading to improved classification performance. In summary, while dimensionality reduction techniques address the challenges associated with processing hyperspectral images, the choice of method often depends on the specific application and the nature of the hyperspectral data.

2.3 Future Directions & Emerging Challenges in Hyperspectral Imaging

The body of work surrounding hyperspectral imaging (HSI) underscores the transformative potential of this technology across a variety of disciplines, from remote sensing, agriculture, and mineral exploration to medical diagnostics and industrial quality control. HSI's ability to capture detailed spectral information from every pixel in a scene offers unparalleled insights that conventional imaging technologies cannot achieve. This potential is being realized through significant advancements in sensor design, data processing algorithms, and machine learning techniques, which are enabling more efficient collection, analysis, and interpretation of hyperspectral data. As the technology evolves, new applications are emerging, demonstrating the versatility and power of HSI in addressing both scientific challenges and real-world problems.

One of the primary drivers of progress in the field is the development of more advanced sensors that offer improved spatial, spectral, and temporal resolution. Innovations such as miniaturized and low-cost sensors have expanded the accessibility of HSI, making it possible to deploy hyperspectral cameras on drones, satellites, and mobile platforms for applications such as precision agriculture, environmental monitoring, and autonomous navigation. Additionally, the integration of AI-driven data processing techniques is helping to overcome the challenges associated with the high-dimensional nature of hyperspectral datasets, enabling efficient classification, anomaly detection, and feature extraction.

However, the rapid growth of hyperspectral imaging is accompanied by several technical and operational challenges. The large volumes of data generated by HSI systems present storage and computational bottlenecks, requiring high-performance computing infrastructure and advanced algorithms for real-time processing. Moreover, the interpretation of hyperspectral data remains complex and labor-intensive, often requiring domain-specific expertise in fields like remote sensing, medical imaging, and geoscience. Addressing these challenges is critical to expanding the utility of HSI and ensuring that it can be effectively applied in diverse operational environments.

2.4 Hyperspectral Image Compression

Hyperspectral imaging has gained significant attention due to its ability to capture rich spectral information across multiple bands, making it valuable for applications in remote sensing, medical imaging, and industrial inspection. However, the high dimensionality and large data volumes associated with HSI present considerable challenges in terms of storage, transmission, and processing. To address these challenges, various compression techniques have been explored, ranging from traditional coding-based methods to modern machine learning-driven approaches.

Compression algorithms are used to compress digital image data, and they can also be applied to text and audio files. These algorithms generate an encoded representation using the minimum number of bits, allowing for transmission and reconstruction at the destination with minimal data loss.

Compression is a fundamental technique used to reduce the size of digital data while preserving its essential information. As digital data grows exponentially in size and complexity, efficient compression methods have become critical for various applications, including storage, transmission, and real-time processing. Whether in cloud storage, streaming services, or embedded systems with limited memory, compression helps optimize resource utilization by eliminating redundant or non-essential data. The effectiveness of compression is measured by the trade-off between the compression ratio (how much the data size is reduced) and the quality of reconstruction after decompression. In many cases, compression improves data efficiency and enhances transmission speed, making it essential for bandwidth-limited environments such as satellite communication and mobile networks.

Compression techniques can be broadly classified based on whether they retain all the original information after decompression. Some methods aim to provide exact reconstruction, ensuring that no information is lost, while others prioritize higher compression ratios at the cost of some data fidelity. The choice of compression technique depends on the specific requirements of the application—whether accuracy, speed, or compact storage is the primary concern. Beyond conventional coding-based approaches, modern compression strategies leverage advanced signal processing and machine learning techniques to optimize data representation. For instance, sparse coding, transform-based methods, and deep learning models have introduced new paradigms for compressing complex datasets, including images and videos. Given these advancements, understanding the core principles of compression is essential for effectively handling large-scale data in various domains.

Compression algorithms can be classified as lossless or lossy depending on the quality of reconstruction sought. The image created at the end of the compression process in lossless compression algorithms should be an exact duplicate of the original image. Lossless algorithms aim to send visual data without losing any information. Only a small amount of compression is achievable due to this restriction. Character data, numerical data, icons, and executable components are all compressed using non-lossy compression. Lossy algorithms, on the other hand, are based on the assumption that it is not required to transmit the complete image file because not all sections of the image are important. Extraneous or inconsequential image elements, such as subtle background noise, minor color variations, or imperceptible details in shadowed regions, could be deleted without compromising the image's visual effect. Furthermore, because redundant data does not need to be communicated, it can be discarded entirely. As a result, lossy compression methods offer a significant level of compression, although at the expense of some information loss and image quality degradation. As a result, lossy compression has been widely used to compress multimedia files.

2.4.1 History and Evolution of Image Compression Techniques

The early methods of image compression, such as Run-Length Encoding (RLE), Huffman coding, and Lempel-Ziv-Welch (LZW), laid the foundation for more complex compression schemes [Birajdar et al., 2019, Moffat, 2019, Mishra et al., 2012]. RLE, for instance, is a lossless compression technique that represents sequences of the same data values as a single value and count, which works well for simple images like icons or drawings but is not effective for complex photographic images. Similarly, Huffman coding, introduced in the 1950s, uses variable-length codes based on the frequency of occurrence of image symbols. While Huffman coding offers better efficiency than RLE, its practical application is still limited to simplistic image structures.

With the advent of Discrete Cosine Transform (DCT) in the late 20th century, more sophisticated compression algorithms such as JPEG were developed, enabling efficient compression of photographic images [Khayam, 2003]. DCT transforms the image into the frequency domain, allowing high-frequency components (which correspond to less visible details) to be discarded, resulting in lossy compression. JPEG, based on DCT, is still one of the most widely used lossy compression standards today due to its simplicity and efficiency. However, JPEG's shortcomings, such as blocking artifacts, loss of fine details, and poor performance in preserving spectral integrity for hyperspectral images, prompted the development of newer methods.

In the mid-1990s, wavelet-based compression emerged as a more advanced technique for image compression, leading to standards such as JPEG2000 [Walker and Nguyen, 2001]. Unlike DCT, which divides images into blocks, wavelets allow for the decomposition of the entire image in a hierarchical manner, capturing both spatial and frequency information simultaneously. This method offers improved image quality, especially at lower bit rates, and is particularly useful for high-resolution images.

JPEG2000, the most popular wavelet-based image compression algorithm, introduced progressive decoding, which allows images to be viewed with increasing quality as more data is transmitted. This method also supports both lossy and lossless compression, making it flexible for various applications. However, despite its technical superiority, JPEG2000 has not been widely adopted in the consumer space due to its computational complexity.

2.4.2 Compression Techniques for Hyperspectral Images

Compression is a technique that reduces the size of a hyperspectral image while ensuring that its quality remains within an acceptable threshold. Hyperspectral images have spectral and spatial redundancy, which must be exploited for compression. Spectral redundancy occurs when consecutive spectral channels in a hyperspectral image capture highly similar spatial information, leading to redundancy in the data, while spatial redundancy is a redundancy relating to statistical relationships between pixel locations. Hyperspectral images are known for their spectral redundancy [Dusselaar and Paul, 2017]. The minimization of spatial redundancy has been the focus of studies in the past two decades. JPEG [Good et al., 1994], and JPEG2000 [Skodras et al., 2001] are two of the most widely used image compression standards. Compression approaches targeted at removing spectral redundancy are continuously being investigated. A desirable hyperspectral image compression solution must eliminate spatial redundancy while also taking into account spectrum redundancy. The goal is to decrease the spatial and spectral correlations in order to produce the best compression result possible. Hyperspectral image compression is a crucial step in hyperspectral image processing, widely employed in space missions to optimize bandwidth usage and reduce storage costs.

Hyperspectral data are equivalent to video data in that spectral band equates to frame, but they have distinctive statistical properties: there is no movement among hyperspectral planes, only color changes. To preserve the scientific value of data, compression should preferably be lossless. On the other hand, lossless compression algorithms yield compression ratios of roughly two or three, a constraint imposed in minimizing data by the noise inherent in high-resolution sensors that create such data [Christophe et al., 2008]. During sensor selection, near-lossless compression, where the highest error between the source and unencrypted image is constrained to a user-defined threshold, becomes an increasingly acceptable option.

Hyperspectral image compression algorithms are classified into several groups based on their application. Lossy or lossless compression methods, inter-band or intra-band compression methods, and compression methods based on various coding algorithms are among the subjects covered.

Various techniques for hyperspectral image compression research are currently accessible in the literature. As previously stated, a three-dimensional data structure known as a hyperspectral image captures combined spatial and spectral data. The hyperspectral image compression algorithms contain two key phases. Spectral decorrelation is usually the first stage in hyperspectral image compression, and Inter-band compression is what we call it. This step decreases the number of dimensions. The elimination of spectral redundancy is a crucial milestone. The second phase is mostly concerned with different types of compression encoders. A 2D or 3D compression encoder can be used. This is referred to as intra-band compression. In fact, both inter-band and intra-band compression should be addressed. These two techniques can sometimes be integrated into a single phase.

Some compression techniques vary in their coding algorithms; for example, some employ Principal Component Analysis (PCA) for image compression, others use Tensor-Based Compression, and others use the Differential Pulse Code Modulation Technique. Many compression methods use deep learning techniques, while others employ the Vector Quantization Transform or the Discrete Cosine Transform.

Hyperspectral image compression techniques can be broadly categorized into traditional transform-based methods, vector quantization techniques, predictive coding methods, dictionary learning approaches, and deep learning-based strategies. Below, we explore these categories in more detail.

2.4.2.1 Vector Quantization-Based Techniques

Vector Quantization (VQ) is a traditional approach used for compressing hyperspectral images. It is effective in minimizing color distribution during compression by representing the data using a finite set of codewords. Several methods have been developed to enhance the performance of VQ in hyperspectral image compression.

A multidimensional vector quantization method has been proposed to improve the compression of hyperspectral images, where the Fuzzy C-Means (FCM) clustering algorithm is employed to determine the optimal number of data clusters and codewords for the codebook [Li et al., 2014]. However, constructing an appropriate codebook remains challenging; for example, if the codebook is poorly designed, pixel intensities may be inaccurately mapped, leading to quality degradation. Additionally, the complexity involved in searching the codebook can slow down processing speed.

The Multidimensional Vector Quantization (MVQ) approach described by Li *et al.*

offers greater flexibility by using fuzzy C-means clustering to optimize both the number of clusters and codewords in the codebook. This adaptive clustering helps capture the inherent variability in hyperspectral data, ensuring more accurate compression across diverse spectral bands. However, balancing codebook complexity with computational efficiency remains a challenge. Larger or poorly optimized codebooks can increase processing time and degrade image quality if pixel intensities are not mapped accurately, emphasizing the need for careful codebook design and management.

To further optimize VQ-based methods, an adaptive VQ approach was introduced to adjust the size of the codebook based on the local statistics of the image [Li et al., 2023]. This technique dynamically updates the codebook as compression progresses, which helps improve compression efficiency and reduces the distortion of reconstructed images. However, the adaptive updating process increases the computational complexity, making it less suitable for real-time applications.

In addition to enhancing compression efficiency, the adaptive Vector Quantization (VQ) approach by Li *et al.* offers fine-grained control over image quality by adjusting the codebook dynamically. This adaptive strategy ensures that high-detail regions retain more information, improving the visual fidelity of the reconstructed image. However, the frequent updates to the codebook can introduce latency, posing a challenge for systems with strict real-time processing needs. Future research could explore ways to balance adaptability with computational efficiency, making it viable for time-sensitive applications.

Another enhancement of VQ techniques involves the use of Classified Vector Quantization (CVQ), where image data is divided into several classes based on spectral or spatial properties before applying VQ to each class individually [Karayiannis and Pai, 1995]. This method helps preserve the spectral information by ensuring that similar spectral features are compressed together, which can lead to better quality in the reconstructed images. CVQ further refines compression by organizing the data into clusters or classes based on spectral or spatial properties, allowing the quantization process to cater specifically to each class's characteristics. This targeted approach not only enhances compression efficiency but also maintains critical patterns within each group, such as material-specific spectral signatures, thereby minimizing artifacts. However, the classification step adds overhead, and the method's effectiveness depends on the accuracy of class assignments, making it sensitive to misclassifications or noisy input data [Karayiannis and Pai, 1995].

Moreover, Hierarchical Vector Quantization (HVQ) has been explored for hyperspectral data compression, where the image is divided into hierarchical levels, and a separate codebook is constructed for each level [Bao et al., 2020]. The advantage of HVQ lies in its ability to capture the multi-resolution characteristics of hyperspectral images, which can result in higher compression ratios without significant loss of image quality. However, managing multiple codebooks across hierarchical levels adds complexity to the encoding and decoding processes.

HVQ enhances hyperspectral image compression by leveraging multiple resolution levels, with each level utilizing a distinct codebook to capture finer details progressively. This hierarchical structure allows HVQ to efficiently manage both spectral and spatial redundancies, leading to improved compression ratios without significant loss in visual or spectral fidelity. However, the need to maintain and synchronize multiple codebooks at different levels introduces additional overhead, complicating the encoding and decoding processes. Optimizing these processes is essential to fully realize the benefits of HVQ in real-world applications [Bao et al., 2020].

Paper [Bascones et al., 2018] utilizes spectral decorrelation through vector quantization combined with PCA to reduce the dimensionality of hyperspectral images while maintaining essential spectral information. This paper combines spectral decorrelation with vector quantization and PCA to efficiently compress hyperspectral images. PCA reduces the data's dimensionality, focusing on preserving essential spectral features, while vector quantization further compresses the transformed data. This dual approach enhances compression ratios and ensures that key spectral information remains intact, supporting applications like material identification or classification. However, the computational burden of applying PCA and vector quantization together can increase processing times, posing challenges for real-time scenarios.

Similarly, Li *et al.* [Li et al., 2019] proposed a method that combines spectral clustering, linear prediction, and vector quantization, improving compression efficiency by reducing redundancy across spectral bands. This approach exploits inter-band correlations more effectively, clustering spectrally similar components and applying linear prediction to reduce redundancy. Vector quantization is then used to compress the residuals, ensuring minimal data loss. This method achieves a balance between compression efficiency and preservation of spectral integrity, making it suitable for applications that require both reduced storage and high data fidelity. However, the interplay between clustering and prediction increases the algorithm's computational complexity.

These techniques illustrate the ongoing evolution of VQ-based methods for hyperspectral image compression, with research efforts focusing on overcoming limitations such as codebook generation, computational complexity, and adaptability to different hyperspectral datasets. Table 2.1 highlights the use of VQ-based techniques in hyperspectral image compression.

2.4.2.2 Transform-Based Techniques

These techniques aim to balance compression ratios and image quality by exploiting the spatial, spectral, or both redundancies in hyperspectral images.

Wavelet-based compression methods are widely recognized for their effectiveness in image compression. Techniques such as Embedded Zerotrees of Wavelet (EZW) Transforms [Shingate et al., 2010, George and Manimekalai, 2014], Set Partitioning in

Year	Paper	Details	Datasets
2014	Novel multivariate vector quantization for effective compression of hyperspectral imagery [Li et al., 2014]	Proposed a multidimensional VQ method to improve the compression of HSI, where the FCM clustering algorithm determines the optimal number of clusters and codewords.	AVIRIS
2018	Hyperspectral image compression using vector quantization, PCA, and JPEG2000 [Bascones et al., 2018]	Spectrally decorrelates the image using Vector Quantization and Principal Component Analysis (PCA).	SUW, Cuprite
2019	The linear prediction vector quantization for hyperspectral image compression [Li et al., 2019]	Proposed a method based on spectral clustering, linear prediction, and vector quantization.	AVIRIS
2019	Multispectral transforms using convolution neural networks for remote sensing multispectral image compression [Li and Liu, 2019]	Uses CNNs to compress the HSI.	Not Mentioned

Table 2.1: Related Work on Vector Quantization-Based Techniques

Hierarchical Trees (SPIHT) [Zala and Parmar, 2013], and Set Partitioned Embedded Block (SPECK) [Tang and Pearlman, 2006] have been explored extensively for hyperspectral image compression. Additionally, regression wavelet analysis has been proposed as a predictive approach, further enhancing compression performance [Amrani et al., 2016]. These techniques illustrate the versatility and effectiveness of wavelet-based methods in compressing hyperspectral images by exploiting the multi-resolution properties inherent to wavelet transforms.

The Embedded Zerotrees of Wavelet (EZW) technique is a well-known wavelet-based compression method for images, including hyperspectral data. It works by encoding significant wavelet coefficients based on their hierarchical structure, allowing for efficient compression. The EZW algorithm utilizes the concept of zero trees, which exploits the hierarchical structure of wavelet coefficients across multiple resolutions. The algorithm works by identifying areas of an image where coefficients are likely to remain insignificant across scales, marking these as zero tree roots. This approach ensures that redundant coefficients are efficiently represented, reducing the size of the encoded data stream [Shapiro, 1993b].

One of EZW's key advantages is its embedded bitstream, which allows the encoder to terminate the compression process at any point and enables scalability and rate control. This makes EZW particularly valuable for applications where the available bandwidth is limited, such as satellite imaging. Additionally, its progressive transmission capability allows partial reconstruction of an image, improving quality incrementally as more data becomes available. Unlike other compression algorithms, EZW does not require pre-trained models or codebooks, further enhancing its versatility. Despite these strengths, the algorithm can be computationally demanding due to its two-pass process: a significance pass, which detects and encodes significant coefficients, and a refinement pass, which improves the precision of already-encoded data. This dual-pass structure enables high compression ratios while maintaining image quality, making EZW suitable for both lossy and lossless compression scenarios.

Shingate *et al.* [Shingate et al., 2010] explored EZW for still image compression, showing its potential for reducing data size without compromising quality. EZW achieves this by efficiently encoding wavelet coefficients and exploiting the inherent correlations within the image data. The algorithm progressively transmits the most important information first, enabling scalable compression where decoding can be stopped at any point to yield a reasonable approximation of the image. This makes it particularly useful for applications with variable storage or bandwidth constraints.
George *et al.* [George and Manimekalai, 2014] further extended this approach to hyperspectral images, demonstrating its capability to exploit the intrinsic multiscale nature of wavelets for better compression. George *et al.* extended the Embedded Zerotree Wavelet (EZW) algorithm to hyperspectral images, leveraging wavelets' multiscale properties to enhance compression efficiency. This approach preserves essential spectral and spatial information by representing the data at multiple resolutions, allowing critical features to be retained even at higher compression ratios. The use of wavelet decomposition ensures that redundancies across both spectral bands and

spatial dimensions are minimized, making the method effective for large-scale hyperspectral datasets.

Cheng *et al.* [Cheng et al., 2014] connected a study specifically tailored for hyperspectral image compression. The research focuses on addressing some of the limitations of the original EZW algorithm, such as handling large-scale hyperspectral datasets with varying spectral bands. The authors improve the algorithm by adapting the encoding process to better capture the correlations within both spatial and spectral dimensions of hyperspectral data. One key improvement discussed in this study is integrating an adaptive strategy that optimizes the encoding of wavelet coefficients. This modification ensures the most critical spectral information is preserved, resulting in higher reconstruction quality and better compression performance. The study demonstrates the effectiveness of the enhanced EZW algorithm by comparing its results to traditional wavelet-based methods, showing higher compression ratios and reduced MSE. The authors also highlight the challenges of balancing compression efficiency with computational complexity, especially for hyperspectral data, where both spatial and spectral redundancies must be exploited effectively. The improved EZW algorithm addresses these challenges by refining the encoding process while maintaining the progressive transmission feature of the original EZW, making it suitable for applications like remote sensing.

Set Partitioning in Hierarchical Trees (SPIHT) is another popular wavelet-based technique that improves upon EZW by refining the process of identifying significant coefficients. SPIHT offers enhanced performance by refining how coefficients are partitioned into significant and insignificant sets [Singh et al., 2017, Dudhagara and Patel, 2017]. It uses a hierarchical tree structure to group related coefficients, allowing it to prioritize the transmission of the most important data first. This results in higher compression ratios and better quality at lower bit rates than EZW. SPIHT also supports progressive transmission, meaning that an image's quality can improve incrementally as more bits are received. This makes it particularly effective for applications where transmission bandwidth is limited, such as satellite imaging or medical data processing, as the image can be reconstructed progressively without needing the full dataset [Dodla et al., 2013].

This research [Dodla et al., 2013] compares the effectiveness of two wavelet-based image compression algorithms: EZW and SPIHT. The study focuses on the strengths and limitations of each method in terms of compression efficiency and image quality. The authors highlight that both EZW and SPIHT leverage wavelet transforms to encode images in a scalable manner, transmitting important coefficients first. However, SPIHT is shown to outperform EZW by refining the way coefficients are partitioned and transmitted. It introduces hierarchical structures to better capture significant coefficients, resulting in higher Peak Signal-to-Noise Ratio (PSNR) and lower Mean Squared Error (MSE) compared to EZW. Additionally, SPIHT supports progressive transmission, allowing image quality to improve incrementally as more data is received—a feature especially valuable in scenarios with bandwidth constraints. This research concludes that SPIHT is better suited for applications requiring high-quality compression at low bit rates, such as medical imaging and remote sensing, due to its improved coding efficiency and progressive transmission capabilities.

Raja *et al.* [Raja and Suruliandi, 2011] provides a detailed comparison of wavelet-based compression techniques, focusing on SPIHT and (Adaptively Scanned Wavelet Difference Reduction) ASWDR. SPIHT's hierarchical tree structure allows for efficient sorting and encoding of significant coefficients, resulting in superior compression performance with minimal loss of image quality. On the other hand, ASWDR enhances compression by dynamically adjusting its scanning order to predict the location of significant coefficients better, leading to higher resolution in compressed images. The research emphasizes that while ASWDR improves edge detail preservation, SPIHT remains a preferred choice due to its simplicity, adaptability, and efficient handling of spatial orientation trees.

SreeVidya *et al.* [SreeVidya et al., 2016] provides a comprehensive overview of the application of the SPIHT algorithm for wavelet-based image compression. The study focuses on the efficiency of SPIHT in encoding images by using hierarchical partitioning, which separates significant and insignificant coefficients. This approach enables progressive transmission, meaning that the image's quality improves as more data is received, making it suitable for applications where bandwidth is limited. The paper discusses the steps involved in SPIHT, including sorting and refinement passes. The algorithm's structure relies on lists, such as the List of Significant Pixels (LSP) and the List of Insignificant Sets (LIS), to ensure that only relevant coefficients are encoded first, minimizing redundancy. The authors also explore the use of the Discrete Wavelet Transform (DWT) to decompose the image into subbands before SPIHT encoding. This process helps capture both high- and low-frequency components, improving the compression ratio without sacrificing reconstruction quality. The authors highlight the advantages of SPIHT over traditional methods like JPEG, emphasizing its adaptability and high PSNR. However, they also note that SPIHT's effectiveness depends on appropriate threshold selection and efficient management of computational resources during encoding and decoding.

Jaiswal *et al.* [Jaiswal and Sedamkar, 2014] explores the comparative performance of EZW and SPIHT for image compression. The focus of the study is on evaluating how both algorithms leverage wavelet transforms to encode images efficiently while preserving quality. The authors highlight that while both EZW and SPIHT employ wavelet decomposition to compress images, SPIHT offers enhanced performance by introducing more refined partitioning and progressive transmission of wavelet coefficients. SPIHT enhances EZW by employing spatial orientation trees to hierarchically structure coefficients for more efficient compression. This structure ensures that significant coefficients are encoded first, which enhances compression ratios and reduces

MSE in the reconstructed image. This study emphasizes that SPIHT's progressive transmission allows partial image reconstruction, improving quality incrementally as more data becomes available, a feature valuable for bandwidth-constrained environments such as remote sensing.

The study provides quantitative performance metrics, comparing PSNR and MSE for the two algorithms. SPIHT consistently outperforms EZW in terms of both compression efficiency and visual quality, achieving higher PSNR values. However, the study notes that SPIHT's improved performance comes at the cost of increased computational complexity, which could be a limiting factor for certain real-time applications. This study demonstrates that SPIHT is a more effective algorithm than EZW for applications requiring high-quality compression at low bit rates, such as satellite imaging and medical diagnostics. However, the trade-off between computational complexity and performance needs to be carefully considered based on the application requirements.

Zala *et al.* [Zala and Parmar, 2013] applied SPIHT to hyperspectral images, where both spatial and spectral dimensions are utilized in the wavelet decomposition. This adaptation enhances compression by taking advantage of the correlations within and across spectral bands, achieving higher compression ratios than traditional implementations. However, the added dimensionality also increases the computational complexity, requiring more memory and processing power during encoding and decoding.

The Set Partitioned Embedded Block (SPECK) algorithm is designed for efficient wavelet-based image compression by focusing on block-based encoding. This block-based method allows SPECK to capitalize on the spatial correlations within small sections of an image, improving both compression and decompression times. By partitioning the wavelet-transformed image into smaller sub-blocks, SPECK effectively manages memory and processing resources, making it well-suited for real-time and hardware-constrained environments, such as medical imaging or remote sensing applications [Cheng et al., 2014]. One of SPECK's strengths lies in its rate-distortion performance, where it achieves near-optimal compression by adaptively adjusting the encoding precision for different blocks based on their importance. This precision is determined by a recursive partitioning process that divides the image into significant and insignificant sets. Only the most important blocks are encoded with high fidelity, while less critical blocks are compressed more aggressively, leading to high compression ratios with minimal impact on image quality [Shapiro, 1993a].

Additionally, SPECK is recognized for its ability to handle progressive transmission, similar to SPIHT, where more critical image information is transmitted first. This is particularly advantageous in applications where image transmission must be halted due to bandwidth constraints, as the partially received data will still produce a visually coherent image. Wavelet decomposition enhances this process by transforming the image into multiple resolution levels, with SPECK further refining the data into progressively smaller blocks based on their significance [Shapiro, 1993a].

SPECK's block-based approach has also been adapted for hyperspectral image compression, where spectral and spatial correlations need to be exploited for effective data reduction. In this adaptation, SPECK compresses both spatial and spectral data simultaneously, improving the efficiency of hyperspectral image compression by focusing on localized features within each block [Raja and Suruliandi, 2011, Cheng et al., 2014]. This versatility, combined with SPECK's efficient use of computational resources, makes it a robust choice for a variety of applications where efficient wavelet-based compression is critical.

Tang *et al.* [Tang and Pearlman, 2006] investigated SPECK for three-dimensional hyperspectral images, organizing the data into blocks before compression. This approach allows for a more localized treatment of the wavelet coefficients, leading to better management of spatial and spectral correlations and yielding improved compression performance. Tang *et al.* demonstrated that applying the SPECK algorithm to hyperspectral images offers several advantages by encoding wavelet coefficients block by block. This localized block-based encoding ensures more precise handling of spatial and spectral correlations. The approach improves compression efficiency and allows flexibility in adapting compression rates across different blocks, which is useful for heterogeneous datasets. However, managing block partitions and optimizing encoding parameters for both spatial and spectral dimensions introduces additional computational overhead.

Regression Wavelet Analysis (RWA) is an approach used in hyperspectral image compression, particularly for lossless encoding. RWA combines wavelet decomposition with regression models to efficiently compress data by predicting wavelet coefficients across multiple scales. After performing a wavelet transform on spectral bands, linear regression is applied to estimate detail coefficients based on approximation coefficients, capturing intricate relationships within the data. This technique minimizes prediction residuals, ensuring high reconstruction accuracy [Ahanonu et al., 2019].

Clustering-based Regression Wavelet Analysis (RWA-C) integrates k-means clustering into wavelet analysis, grouping similar spectral components to enhance compression. This approach assigns feature vectors to each cluster, which are used during regression to improve prediction accuracy and reduce entropy in the residuals. The clustering step ensures that similar regions are processed together, minimizing redundancy and boosting efficiency in hyperspectral data compression. However, balancing the computational cost of clustering with compression gains remains a challenge. For more information, see the detailed discussion [Ahanonu et al., 2019].

Amrani *et al.* [Amrani et al., 2016] proposed a method that combines wavelet-based techniques with regression analysis for hyperspectral image compression. This predictive approach uses wavelet coefficients to forecast pixel values based on local image features, further enhancing the accuracy of compression. Regression wavelet analysis offers the benefit of reducing prediction errors while retaining important spectral information.

2.4.2.3 Karhunen-Loeve Transform (KLT)

The Karhunen-Loeve Transform has also been used to reduce spectral redundancy in hyperspectral images before applying other compression techniques like JPEG2000. KLT-based methods can compress the most relevant spectral information into fewer components by focusing on principal components, optimizing compression ratios while preserving significant features [González-Conejero et al., 2009].

By applying KLT as a pre-processing step before methods like JPEG2000, the transformed data can be compressed more efficiently, with the reduced number of principal components minimizing the amount of data to be encoded. This synergy between KLT and other compression techniques optimizes the compression ratio while retaining the essential features required for tasks such as classification or object detection [Ahanonu et al., 2019]. Furthermore, KLT helps mitigate artifacts and losses associated with lossy compression by focusing the compression efforts on the most informative parts of the data. However, the computational cost of calculating eigenvectors and eigenvalues for large datasets can be a challenge, especially in real-time applications. To address this, approximate or incremental KLT algorithms have been explored, providing a trade-off between performance and computational efficiency. Despite these challenges, KLT remains a powerful tool for hyperspectral image processing, particularly in remote sensing, environmental monitoring, and defense applications, where preserving spectral integrity is crucial for further analysis and interpretation [Cheng et al., 2014].

SA-JPEG2000 (Spatially Adaptive JPEG2000) extends the standard JPEG2000 framework by incorporating the Karhunen-Loeve Transform (KLT) to address the challenges of spectral redundancy in hyperspectral images. This combined approach allows the compression system to apply KLT to reduce inter-band correlations, compacting the most essential spectral information into fewer components. After spectral decorrelation, the JPEG2000 framework processes the spatial components efficiently, ensuring a high compression ratio without compromising the quality of critical data [González-Conejero et al., 2009, Ahanonu et al., 2019].

A unique feature of SA-JPEG2000 is its handling of "no-data" regions—areas in the hyperspectral image that contain no meaningful information. These regions can arise due to sensor limitations, cloud cover in remote sensing images, or occlusions. The SA-JPEG2000 algorithm optimizes the compression process by bypassing or encoding no-data regions minimally, reducing the computational and storage burden. This targeted compression ensures that only relevant portions of the data consume bandwidth and memory, further enhancing the method's efficiency [Cheng et al., 2014, Ahanonu et al., 2019]. This approach is particularly valuable for applications in satellite imaging, where data from certain regions may not be useful or fully captured. By selectively compressing active regions and minimizing the impact of no-data areas, SA-JPEG2000 achieves a better balance between compression performance and data quality. However, the computational overhead of performing both KLT and JPEG2000 encoding may require hardware acceleration or parallel processing to meet real-time processing demands. As a result, SA-JPEG2000 is a suitable choice for scenarios where high spectral and spatial fidelity must be preserved, such as environmental monitoring, agricultural analysis, or military surveillance [González-Conejero et al., 2009, Ahanonu et al., 2019].

2.4.2.4 Hybrid Transform Techniques

Combining multiple transforms, such as wavelet transforms and DCT, offers a powerful strategy for enhancing hyperspectral image compression. Each transform focuses on different aspects of the data, with wavelets efficiently capturing local variations and high-frequency components and DCT excelling at encoding global, low-frequency features. Together, these complementary transforms provide a more flexible and adaptive compression framework, enabling finer adjustments based on the characteristics of different spectral bands [Amrani et al., 2016, Ahanonu et al., 2019]. The wavelet transform decomposes an image into subbands, isolating high-frequency details such as edges and textures. In contrast, DCT aggregates the low-frequency information, condensing it into fewer coefficients, which reduces redundancy in smoother regions of the image. This hybrid approach ensures that both local anomalies (captured by wavelets) and global trends (captured by DCT) are effectively managed, yielding better compression ratios with minimal loss of data quality [Cheng et al., 2014].

This multi-transform strategy is especially beneficial for heterogeneous hyperspectral data, where certain spectral bands contain more noise or less relevant information. By applying the appropriate transform to different components, the compression algorithm can prioritize high-value regions while reducing the size of redundant or less significant data. Additionally, hybrid transform-based methods can adapt to varying bitrate constraints, ensuring optimal performance for real-time applications such as satellite imaging or environmental monitoring. However, implementing a hybrid wavelet-DCT framework introduces computational complexity, as each transform requires separate processing. To address this, some methods employ optimization techniques or hardware accelerators to achieve real-time performance. Nonetheless, the combination of multiple transforms continues to be a promising avenue for advancing hyperspectral image compression, balancing efficiency with high data fidelity [Amrani et al., 2016, Ahanonu et al., 2019].

The shared source encoding approach using embedded zero tree DCT (EZT-DCT) proposed by Pan *et al.* simplifies the compression process by encoding only the significant DCT coefficients, similar to the concept used in EZW for wavelets. This method leverages zero trees, which efficiently represent zero-valued coefficients and their descendants, reducing redundancy and bitstream size. It provides an effective balance between compression efficiency and computational simplicity, making it an attractive option for applications requiring fast encoding and decoding with minimal resources [Pan et al., 2011]. In their work, Pan *et al.* proposed a low-complexity image compression technique using EZT-DCT. The key innovation of this method lies in integrating the DCT with zero-tree encoding to enhance image compression efficiency. By leveraging DCT's energy compaction properties alongside the hierarchical representation of zero-tree encoding, this approach reduces computational overhead compared to traditional wavelet-based methods. This makes it particularly suitable for resource-constrained environments such as mobile or embedded systems [Pan et al., 2011].

To extend the approach proposed by Pan *et al.*, 3D-DCT was introduced by Qiao *et al.* and Rasti *et al.*. These works address the challenges posed by the large dimensionality of hyperspectral images by dividing the data into $8 \times 8 \times 8$ subcubes. This grouping ensures that spatial and spectral redundancies are exploited efficiently within small, manageable blocks. The 3D-DCT approach not only improves compression ratios but also allows for band-specific access during decompression, a critical feature for hyperspectral imaging, where users often need to analyze only a subset of bands rather than the entire dataset [Qiao et al., 2014, Rasti et al., 2012]. Qiao *et al.* extended the concept to 3D-DCT compression for hyperspectral data. Their method processes $8 \times 8 \times 8$ subcubes, taking advantage of the spatial and spectral correlations within these blocks. This extension ensures high compression efficiency and enables the selective decoding of individual bands, optimizing performance for tasks like remote sensing or agricultural monitoring, where only specific bands are analyzed [Qiao et al., 2014]. Rasti *et al.* further refined the 3D-DCT-based compression framework by optimizing the block size and improving the quantization process to enhance both compression ratio and reconstruction quality. They demonstrated that their method outperforms other transform-based techniques on various hyperspectral datasets, offering a practical solution for large-scale data management. This work highlights the importance of balancing compression efficiency, quality, and access speed for effective hyperspectral image processing [Rasti et al., 2012].

The adaptive band selection technique introduced alongside 3D-DCT addresses the challenge of high redundancy across spectral bands in hyperspectral images. In hyperspectral datasets, many bands contain similar information, which leads to unnecessary data duplication during compression. The adaptive band selection strategy works by grouping highly correlated bands based on their spectral similarity, ensuring that only essential information is retained for compression. This dimensionality reduction not only improves compression efficiency but also accelerates the encoding and decoding processes by reducing the number of bands to process [Rasti et al., 2012]. Once the correlated bands are grouped, they are compressed using JPEG-LS, a lossless compression standard optimized for images with smooth gradients. JPEG-LS enhances the performance of 3D-DCT by efficiently encoding the reduced spectral data, leading to better compression ratios and faster decompression speeds. This combination ensures that while redundant data is minimized, the spectral fidelity of the retained information remains intact, making it suitable for applications like remote sensing and agriculture, where accurate spectral information is critical for analysis [Cheng et al., 2014]. The adaptive band selection technique also enables flexibility in data access, as grouped bands can be selectively decoded based on the user's requirements. This is particularly advantageous for large-scale hyperspectral datasets, as users often need only a specific subset of bands. Additionally, by minimizing redundant spectral data, the overall storage and transmission requirements are significantly reduced, making this method well-suited for real-time applications with limited bandwidth or storage constraints [Ahanonu et al., 2019].

Karami *et al.* [Karami et al., 2010] introduced a Hybrid 3D-DCT and Tucker Decomposition method for hyperspectral image compression, combining two powerful techniques to leverage both spectral and spatial correlations. The 3D-DCT efficiently reduces redundancy within localized blocks of the hyperspectral cube, while the Tucker decomposition captures global data structures by factorizing the data into a core tensor and factor matrices. This hybrid approach balances local and global data representation, resulting in high compression ratios while preserving spectral fidelity. Karami *et al.* demonstrated that their method is particularly effective for large datasets, offering superior performance compared to individual use of either DCT or Tucker decomposition, making it suitable for real-world applications like remote sensing and environmental monitoring.

Toreyin *et al.* [Töreyın et al., 2015] explored wavelet transform-based spectral decorrelation techniques for lossless hyperspectral image compression. Their method applies wavelet transforms to decompose the hyperspectral data into multiple frequency subbands, separating high-frequency noise from meaningful low-frequency components. This decomposition allows the algorithm to decorrelate spectral bands efficiently, ensuring that redundant information is minimized across the dataset. The wavelet-based approach not only preserves essential spectral features but also achieves high compression efficiency without compromising the reconstruction quality. Toreyin *et al.*'s technique is particularly valuable for applications requiring lossless compression, such as scientific imaging and medical diagnostics, where data integrity is paramount.

Furthermore, Nagendran *et al.* [Nagendran et al., 2024] proposed a lossless hyperspectral image compression technique that combines spectral decorrelation methods with Binary Embedded Zero Tree Wavelet (BEZW) coding. This approach leverages the wavelet transform's ability to separate image data into different frequency components, minimizing both spatial and spectral redundancy. By applying BEZW coding to the wavelet coefficients, the algorithm efficiently encodes regions with minimal information, resulting in highly compressed bitstreams with minimal loss of quality. The use of spectral decorrelation prior to BEZW coding ensures that redundant information across the many spectral bands of hyperspectral data is reduced. This combination is particularly effective for remote sensing applications, where high compression ratios are essential due to bandwidth and storage limitations. However, one challenge lies in maintaining high-quality reconstruction across varying spectral distributions, as BEZW coding assumes certain properties in the underlying data that may not always hold. The study demonstrates that BEZW-based compression offers a promising balance between computational complexity and compression efficiency, making it ideal for onboard systems in satellites or aerial platforms.

Transform-based compression techniques have played a critical role in hyperspectral image compression, as outlined in Table 2.2. Methods such as the Hybrid 3D-DCT and Tucker Decomposition proposed by Karami *et al.* [Karami et al., 2010], and wavelet transform-based spectral decorrelation techniques [Töreyın et al., 2015], provide efficient means of compressing hyperspectral images by leveraging spectral and spatial redundancy. These transform-based techniques continue to be foundational in reducing the complexity and storage requirements of hyperspectral data.

Table 2.2: Related Work on Transform-Based Techniques

Year	**Paper**	**Details**	**Datasets**
2010	Hyperspectral image compression based on Tucker decomposition and discrete cosine transform [Karami et al., 2010]	Uses Hybrid 3D-DCT and TD	AVIRIS
2014	An Improved EZW Hyperspectral Image Compression [Cheng et al., 2014]	Adopts the spatial-spectral hybrid transform and the proposed transform-based coder	AVIRIS

Continued on next page

Table 2.2 – continued from previous page

Year	Paper	Details	Datasets
2015	Lossless hyperspectral image compression using wavelet transform based spectral decorrelation [Töreyın et al., 2015]	Uses JPEG-LS	AVIRIS
2016	Lossy compression of Landsat multispectral images [Kozhemiakin et al., 2016]	Combines channels into a group and uses DCT	Landsat
2016	Regression Wavelet Analysis for Lossless Coding of Remote-Sensing Data	Introduces a wavelet-based scheme to increase coefficient independence in hyperspectral images	AVIRIS
2017	ROI-based on-board compression for hyperspectral remote sensing images on GPU [Giordano and Guccione, 2017]	Uses a clustering algorithm that automatically segments the image	AVIRIS
2018	A new algorithm for the on-board compression of hyperspectral images [Guerra et al., 2018]	Uses a transform-based lossy compression algorithm, namely Lossy Compression Algorithm for Hyperspectral Image Systems (HyperLCA)	Indian Pines, Pavia

Continued on next page

Table 2.2 – continued from previous page

Year	Paper	Details	Datasets
2019	Clustering Regression Wavelet Analysis for Lossless Compression of Hyperspectral Imagery [Ahanonu et al., 2019]	Proposes Clustering Regression Wavelet Analysis (RWA-C), which improves compression performance while maintaining component scalability	AVIRIS
2019	Multispectral transforms using convolution neural networks for remote sensing multispectral image compression [Li and Liu, 2019]	Uses CNNs to compress HSIs	Not mentioned
2020	Fuzzy transform for high-resolution satellite images compression [Monica and Widipaminto, 2020]	Uses fuzzy transform to compress images	Pleiades constellation satellites
2023	Curvelet Transform Based Compression Algorithm for Low Resource Hyperspectral Image Sensors [Bajpai et al., 2023]	Proposes a curvelet transform-based method for hyperspectral image compression	Pavia

Continued on next page

Table 2.2 – continued from previous page

Year	Paper	Details	Datasets
2024	Lossless hyperspectral image compression by combining the spectral decorrelation techniques with transform coding methods [Nagendran et al., 2024]	Explores the utilization of Binary Embedded Zero Tree Wavelet Algorithms (BEZW) to compress hyperspectral images	Pavia

2.4.3 PCA and Tensor Decomposition-Based Techniques

Principal Component Analysis (PCA) and Tensor Decomposition techniques have gained application in hyperspectral image compression due to their ability to reduce the dimensionality of large datasets while preserving essential spectral information. These techniques are particularly useful for hyperspectral images, where the number of spectral bands is significantly higher than that of typical RGB or multispectral images.

PCA-based methods have been extensively used for compressing hyperspectral images. For instance, Uddin *et al.* [Uddin et al., 2021] applied both non-linear and linear variants of PCA to reduce the dimensionality of hyperspectral images. Uddin *et al.* demonstrated the effectiveness of both linear and non-linear PCA for hyperspectral image compression by reducing high-dimensional data into fewer components while retaining essential spectral features. The linear variant offers simplicity and computational efficiency, making it suitable for real-time processing, whereas the non-linear PCA variant captures complex, non-linear patterns in the data, improving compression quality for applications requiring higher precision. This dual approach allows for better adaptability across various datasets, balancing between speed and accuracy depending on the task requirements.

Similarly, Gowtham *et al.* [Gowtham et al., 2021] incorporated PCA with a Siamese network, effectively using the combination to compress hyperspectral data and improve classification performance. The Siamese network, known for its ability to learn similarity metrics, helps capture intricate relationships between spectral bands. When combined with PCA, this method not only reduces data dimensionality but also preserves discriminative features crucial for accurate classification. This synergy between PCA's feature reduction and the network's similarity learning offers a robust framework for hyperspectral image analytics.

Jayaprakash *et al.* [Jayaprakash et al., 2020] extended the PCA framework by applying Randomized Independent Component Analysis (RFICA) and Randomized Linear Discriminant Analysis (RFLDA) to achieve more efficient compression. These methods leverage the ability of PCA and similar techniques to identify and reduce redundancy among the spectral bands, making them highly effective for high-dimensional data like hyperspectral images. This framework ensures efficient compression with minimal loss of spectral information, offering a valuable balance between accuracy and performance.

Tensor Decomposition techniques, such as the Correlation-Based Tucker Decomposition (CBTD) proposed by Li *et al.* [Li et al., 2021], aim to decompose the multi-dimensional data into smaller, more manageable components while retaining the core information. Aidini *et al.* [Aidini et al., 2019] applied low-rank tensor completion to compress hyperspectral images while also enhancing their super-resolution capabilities.

Similarly, Xue *et al.* [Xue et al., 2019] proposed a nonlocal tensor sparse and low-rank regularization (NTSRLR) approach, which efficiently captures the structured sparsity of hyperspectral images for compressive sensing reconstruction. Tensor decomposition methods further extend compression capabilities by addressing multi-

Year	Paper	Details	Datasets
2019	Nonlocal tensor sparse representation and low-rank regularization for hyperspectral image compressive sensing reconstruction [Xue et al., 2019]	Proposes a nonlocal tensor sparse and low-rank regularization (NTSRLR) approach, which can encode essential structured sparsity of an HSI.	Pavia, Indian Pines
2020	PCA-based feature reduction for hyperspectral remote sensing image classification [Uddin et al., 2021]	Uses non-linear and linear variants of PCA to compress HSIs.	Indian Pines
2020	Randomized independent component analysis and linear discriminant analysis dimensionality reduction methods for hyperspectral image classification [Jayaprakash et al., 2020]	Applies Randomised independent component analysis (RFICA) and randomised linear discriminant analysis (RFLDA) to compress HSIs.	Salinas, Pavia
2021	Hyperspectral image analysis using principal component analysis and siamese network [Gowtham et al., 2021]	Uses PCA and siamese network.	Pavia, Indian Pines
2021	The correlation-based Tucker decomposition for hyperspectral image compression [Li et al., 2021]	Proposes a CBTD method that can be directly used in most tensor-based compression methods to obtain a highly improved compression performance.	Indian Pines, Salinas

Table 2.3: Related Work on PCA and Tensor Decomposition-Based Techniques

dimensional redundancies. Techniques such as Correlation-Based Tucker Decomposition (CBTD) efficiently break down hyperspectral data into smaller components without losing essential spectral information. Similarly, nonlocal tensor regularization approaches like NTSRLR leverage both sparse and low-rank structures to ensure more efficient data reconstruction for hyperspectral image datasets [Licciardi et al., 2014]. Table 2.3 summarizes the recent advancements in PCA and tensor decomposition-based compression methods, highlighting their effectiveness across several benchmark datasets such as Indian Pines, Salinas, and Pavia.

2.4.4 Compressive Sensing-Based Techniques

Compressive Sensing (CS) techniques have recently emerged as powerful tools for hyperspectral image compression, offering the ability to reconstruct high-dimensional signals from fewer measurements than conventional methods. These techniques exploit the sparsity inherent in hyperspectral images, allowing for a significant reduction in data while preserving critical information for tasks such as classification and reconstruction.

One such technique is the SHSIR (Sparsification of Hyperspectral Image and Reconstruction) algorithm proposed by Gunasheela *et al.* [Gunasheela and Prasantha, 2018], which compresses hyperspectral images using a sparse reconstruction approach. This method is particularly well-suited for satellite imagery, as demonstrated on the URBAN dataset. SHSIR focuses on identifying the most relevant spectral components while discarding redundant information, ensuring that critical features are preserved. This sparse representation is particularly effective for satellite-based imagery, as shown using the URBAN dataset, where limited bandwidth and storage are constraints. However, the algorithm's reliance on sparse reconstruction may introduce challenges in real-time scenarios, given the computational resources required for high-dimensional data processing.

One such technique is the Sparsification of Hyperspectral Image and Reconstruction (SHSIR) algorithm proposed by Gunasheela *et al.* [Gunasheela and Prasantha, 2018], which compresses hyperspectral images using a sparse reconstruction approach. This method is particularly well-suited for satellite imagery, where bandwidth and storage constraints require efficient data representation. SHSIR identifies the most relevant spectral components while discarding redundant information, preserving critical features essential for analysis. However, its reliance on sparse reconstruction may introduce computational challenges in real-time applications, particularly due to the high-dimensional nature of hyperspectral data processing.

Similarly, Kumar *et al.* [Kumar et al., 2018] proposed a real-time compressive sensing-based scheme on board aimed at satellite applications, where data storage and transmission capacities are restricted. Their method combines deep learning with

Year	Paper	Details	Datasets
2018	Onboard hyperspectral image compression using compressed sensing and deep learning [Kumar et al., 2018]	Proposes a real-time onboard compression scheme.	Pavia
2019	Compressive sensing approach to satellite hyperspectral image compression [Gunasheela and Prasantha, 2018]	Proposes a new algorithm called SHSIR (sparsification of hyperspectral image and reconstruction) for the compression of HSI using the compressive sensing approach.	URBAN
2019	Nonlocal tensor sparse representation and low-rank regularization for hyperspectral image compressive sensing reconstruction [Xue et al., 2019]	Proposes a nonlocal tensor sparse and low-rank regularization (NTSRLR) approach, which can encode essential structured sparsity of an HSI.	Pavia, Indian Pines

Table 2.4: Related Work on Compressive Sensing-Based Techniques

compressed sensing, achieving high compression ratios while maintaining fidelity in the reconstructed data. By combining the strengths of compressed sensing and neural networks, the method effectively balances data fidelity with storage and transmission constraints, making it ideal for remote sensing applications. However, optimizing deep learning models for onboard use presents challenges in terms of computational efficiency and hardware limitations.

Xue *et al.* [Xue et al., 2019] further advanced compressive sensing techniques by incorporating a Nonlocal Tensor Sparse and Low-rank Regularization (NTSRLR) approach. This method effectively reduces the dimensionality of hyperspectral data while retaining the spatial and spectral coherence necessary for high-quality reconstruction. NTSRLR is particularly well-suited for large hyperspectral datasets, balancing dimensionality reduction with high fidelity. However, the nonlocal nature of the approach introduces computational complexity, requiring optimization for real-time applications and large-scale processing. Table 2.4 details the key contributions of compressive sensing techniques, emphasizing their utility in reducing hyperspectral data volumes without compromising the overall accuracy of reconstruction tasks.

2.4.5 Dictionary Learning and Sparse Coding

Recent advancements in machine learning have significantly influenced hyperspectral image compression, particularly through dictionary-based learning (DL) and sparse coding techniques. These methods aim to capture the intrinsic properties of the data by using only a subset of dictionary elements, unlike traditional fixed dictionary approaches [Qian et al., 2012, UlkU and TOreyin, 2014]. For example, dictionary learning algorithms applied to hyperspectral data can achieve high compression efficiency through sparse representation [Charles et al., 2011]. This approach adapts the dictionary to the unique properties of the data, ensuring that only the most relevant components are retained. By focusing on sparse coding, the method reduces storage requirements while maintaining essential spectral information for accurate reconstruction. However, constructing optimal dictionaries can be computationally intensive, especially for large datasets, requiring advanced optimization techniques for practical implementation.

In this study, the authors explore the application of dictionary learning algorithms for hyperspectral image compression, emphasizing the effectiveness of sparse representation techniques in achieving high compression efficiency. The core idea of sparse representation is to express data using a small number of significant elements from a larger dictionary. In the context of hyperspectral images, this means that the high-dimensional data can be approximated well using a few dictionary elements.

The authors demonstrate that the compression process can retain essential information while reducing the overall data size by selecting the most relevant features from hyperspectral data. The paper discusses various dictionary learning methods designed to adaptively learn a dictionary from the hyperspectral data itself, rather than relying on pre-defined or fixed dictionaries. This adaptability allows the algorithm to tailor the dictionary to the specific characteristics of the dataset, enhancing the representation capability for the unique spectral signatures found in hyperspectral images. The findings from this research highlight the potential for dictionary learning algorithms to be integrated into practical hyperspectral imaging systems.

An online training sparse encoding technique was recently introduced for hyperspectral image compression, representing a pioneering application of online dictionary learning in this domain [UlkU and Toreyin, 2015, Castrodad et al., 2011]. The study [UlkU and Toreyin, 2015] introduces an innovative online training sparse encoding technique for hyperspectral image compression, representing a significant advancement in the application of online dictionary learning. This approach aims to optimize the representation of hyperspectral data by leveraging sparse coding, which focuses on using a minimal number of dictionary elements to reconstruct the data accurately. The authors detail a two-stage algorithm that consists of a dictionary learning phase followed by a dictionary update phase. The initial phase is the dictionary learning phase. In this initial phase, the algorithm randomly selects data samples from the hyperspectral image cube. Sparse coding is then performed using the Least Angle Regression (LARS) algorithm to find sparse coefficients that minimize the reconstruction error. Constraints are enforced to ensure the coefficients and dictionary elements remain non-negative, which is essential for hyperspectral data representation.

The second phase is the dictionary update phase. After obtaining the sparse representations, the dictionary is updated iteratively. This update is based on auxiliary matrices that help refine the dictionary elements to fit the training data better. The algorithm continues to adjust the dictionary until the difference between the current and updated dictionary elements falls below a specified threshold. The proposed method has shown promising results, achieving competitive compression performance compared to state-of-the-art techniques, such as JPEG2000 and other predictive lossy compression methods.

Extensive experiments conducted using datasets like AVIRIS demonstrate the effectiveness of this online learning approach in enhancing both compression efficiency and data reconstruction quality [UlkU and Toreyin, 2015]. A recent method integrates spatial and spectral dictionaries for hyperspectral image denoising and compression. This approach extracts cubic patches from hyperspectral images and applies PCA-based clustering in both spatial and spectral domains, resulting in adaptive dictionaries tailored for specific data characteristics. This combined strategy improves compression efficiency and data reconstruction accuracy, particularly for complex datasets like the Columbia Multispectral Image Dataset [Fu et al., 2015].

Additionally, a dictionary learning method incorporating spectral grouping has been suggested, combining online dictionary learning with classification characteristics of spectral curves. The paper [Jifara et al., 2017] presents a novel approach to hyperspectral image compression that integrates dictionary learning with spectral grouping techniques. This method leverages the unique classification characteristics of spectral curves to enhance the performance of dictionary learning in capturing the inherent structure of hyperspectral data. The authors propose that hyperspectral images often exhibit inherent similarities within certain spectral bands. By grouping these similar bands, the algorithm can reduce redundancy and improve the efficiency of the compression process. The grouping is based on the spectral signatures of the pixels, which allows the method to focus on the most representative features of the data.

The research employs an online dictionary learning framework that updates the dictionary in real-time as new data becomes available. This dynamic adaptation is particularly beneficial for hyperspectral imaging applications, where the acquisition of data is often sequential. By continually refining the dictionary based on the grouped spectral characteristics, the algorithm ensures that it maintains an accurate representation of the data throughout the compression process. Through this combined approach, the authors demonstrate that the proposed method significantly enhances data reconstruction quality while maintaining high compression ratios. The algorithm's ability to effectively capture and represent the essential features of the hyperspectral data leads to improved performance compared to traditional fixed dictionary approaches. The results presented in this study indicate that the integration of spectral grouping with online dictionary learning is a promising direction for advancing hyperspectral image compression techniques. This approach not only improves compression efficiency but also facilitates better preservation of spectral information during the encoding process [Jifara et al., 2017].

Table 2.5 showcases the contribution of Dictionary Learning and Sparse Coding-based techniques to hyperspectral image compression. These methods, such as those by Jifara *et al.* [Jifara et al., 2017], apply a spectral dictionary learned through sparse coding to compress hyperspectral images effectively. Ulku *et al.* [UlkU and Kizgut, 2018] extend this by using proximity-based optimization algorithms to achieve large-scale hyperspectral compression. These techniques exploit the sparsity inherent in hyperspectral images to minimize data redundancy, resulting in high compression efficiency without significant loss of critical information.

2.4.6 Predictive Coding Techniques

Predictive coding strategies leverage statistical models to predict pixel values based on surrounding information. A two-stage prediction technique employing Wiener filtering has been proposed, with the first stage performing initial filtering and the second stage refining the prediction using a reverse pixel location search algorithm [Lin and Hwang, 2010]. The first stage involves initial filtering of the image using Wiener filtering, a method known for its effectiveness in minimizing the MSE between the original and reconstructed pixel values. This filtering process works by taking into account the statistical characteristics of the image noise, leading to a more accurate prediction of pixel values. The second stage refines this initial prediction through a

Year	Paper	Details	Datasets
2014	Lossy compression of hyperspectral images using online learning based sparse coding [UlkU and TOreyin, 2014]	Proposes a lossy hyperspectral image compression method using online learning based sparse coding	AVIRIS
2015	Sparse coding of hyperspectral imagery using online learning [UlkU and Toreyin, 2015]	Proposed a hyperspectral image compression method using a discriminative online learning-based sparse coding algorithm	AVIRIS
2017	Hyperspectral image compression based on online learning spectral features dictionary [Jifara et al., 2017]	Uses a spectral dictionary that is learned in sparse coding to compress HSIs.	AVIRIS
2018	Large-scale hyperspectral image compression via sparse representations based on online learning [UlkU and Kizgut, 2018]	Uses proximity-based optimization algorithms for HSI compression.	AVIRIS
2019	Efficient lossless compression of multitemporal hyperspectral image data [Shen et al., 2018]	Proposes an information-theoretic analysis to estimate potential compression performance for multitemporal HSI data.	Natural Scenes

Table 2.5: Related Work on Dictionary Learning and Sparse Coding Techniques

reverse pixel location search algorithm. This step improves the accuracy of the predictions by assessing the spatial relationships among pixels and correcting any discrepancies from the first stage. By refining the predictions based on a more localized context, this approach aims to enhance the overall reconstruction quality. Wiener filtering is well-regarded for its ability to reduce noise and enhance signal quality. However, while it performs well in many scenarios, its effectiveness can be compromised when dealing with hyperspectral images that exhibit diverse characteristics across different spectral bands. These variations in spectral signatures and noise characteristics can challenge the assumptions underlying Wiener filtering, potentially leading to suboptimal performance in certain cases [Lin and Hwang, 2010].

The authors note that the applicability of Wiener filtering in the context of hyperspectral image data may be limited due to the inherent complexity and variability of the data. This limitation underscores the need for further research into more robust predictive coding strategies that can adapt to the diverse nature of hyperspectral images, ensuring that compression methods remain effective across a wide range of scenarios [Lin and Hwang, 2010]. Overall, the paper highlights the potential of predictive coding strategies, particularly with the two-stage Wiener filtering approach, to improve hyperspectral image compression while also acknowledging the challenges posed by the variability of hyperspectral data. This research contributes to ongoing efforts to develop more efficient and effective compression techniques in the field of hyperspectral imaging. While Wiener filtering is effective in reducing the mean squared error between the original and reconstructed images, its applicability may be limited due to the diverse characteristics of hyperspectral image data.

Paper [Zheng et al., 2022] introduces a technique that plays a key role in near-lossless hyperspectral image compression. It removes spatial and spectral redundancies by iteratively updating prediction coefficients to minimize prediction errors, significantly reducing data size while maintaining high reconstruction quality. RLS filtering adapts to new data dynamically, which is particularly useful for hyperspectral images acquired over time [Zheng et al., 2022]. A recent approach combines predictive coding with convolutional neural networks (CNNs) to enhance image reconstruction after lossy compression is introduced in [Valsesia and Magli, 2019]. This method leverages the spatial and spectral correlations within hyperspectral data and offers competitive performance compared to traditional predictive coding algorithms. CNNs are used on the ground segment to correct the residual errors left by onboard predictive coding, ensuring improved accuracy during decompression [Valsesia and Magli, 2019].

Paper [Lin and Hwang, 2011] introduces advanced predictive coding schemes that utilize look-up tables and multi-band prediction models to handle the inter-band correlations in hyperspectral images. These techniques calculate prediction coefficients by analyzing pixels across multiple spectral bands, refining predictions using linear models optimized for minimal mean squared error. This strategy enhances the compression ratio, making it effective for datasets with highly correlated spectral information [Lin and Hwang, 2011].

The table on Predictive Coding Techniques (Table 2.6) illustrates approaches that predict pixel values based on previously known values to achieve hyperspectral image compression. Jiang *et al.* [Jiang et al., 2018] utilized Long Short-Term Memory (LSTM)-based recurrent neural networks to reduce prediction errors in hyperspectral images, enhancing compression performance. In hardware-based solutions, Barrios *et al.* [Barrios et al., 2018] implemented the CCSDS 123.0-B-1 compression standard on FPGA-based systems for real-time compression of hyperspectral data. These predictive coding methods leverage advanced algorithms to reduce data redundancies, particularly in sequential spectral data, resulting in efficient compression schemes.

Predictive coding techniques aim to efficiently compress data by predicting pixel values based on neighboring or previously encoded pixels. In hyperspectral imaging, these methods become particularly useful for handling no-data regions, where portions of the image contain missing or unimportant information. One effective strategy involves using GR (Golomb-Rice) codes to encode the prediction residuals and boundary pixels adjacent to these no-data areas. GR codes are well-suited for compressing residuals with geometric or exponential distributions, as they provide a compact representation of small-valued prediction errors, minimizing the overall bitstream size [Shen et al., 2016, Ahanonu et al., 2019].

However, the efficiency of GR codes is contingent upon the statistical behavior of the residuals. If the prediction errors deviate from an exponential distribution, the performance of GR-based encoding can degrade, leading to suboptimal compression ratios. In such cases, alternative entropy coding techniques, such as arithmetic coding or Huffman coding, may be required to better match the distribution of residuals and improve compression performance [Cheng et al., 2014, Ahanonu et al., 2019]. To address these limitations, some methods incorporate adaptive predictive coding

Year	Paper	Details	Datasets
2018	LSTM based adaptive filtering for reduced prediction errors of hyperspectral images [Jiang et al., 2018]	Uses LST-RNN.	Indian Pines, Pavia
2018	Hardware implementation of the CCSDS 123.0-B-1 lossless multispectral and hyperspectral image compression standard by means of high-level synthesis tools [Barrios et al., 2018]	Hyperspectral Image Compression algorithm is implemented over an FPGA-based MPSoC.	AVIRIS, CRISM
2018	An efficient real-time FPGA implementation of the CCSDS-123 compression standard for hyperspectral images [Fjeldtvedt et al., 2018]	Presents a low-complexity high-throughput FPGA implementation of CCSDS-123 compression algorithm with band interleaved by pixel ordering.	MODIS, Hyperion, AVIRIS, HICO
2019	High-Throughput On-board Hyperspectral Image Compression With Ground-Based CNN Reconstruction [Valsesia and Magli, 2019]	Studies the performance of a faster method based on the quantization of the image, followed by a lossless predictive compressor.	Not provided
2019	Superpixel based recursive least-squares method for lossless compression of hyperspectral images [Karaca and Güllü, 2019]	Uses super RLS.	AVIRIS
2022	Recursive Least Squares for Near-Lossless Hyperspectral Data Compression [Zheng et al., 2022]	Proposes a near-lossless prediction-based compression scheme that removes spatial and spectral redundant information, thereby significantly reducing the size of hyperspectral images.	AVIRIS

Table 2.6: Related Work on Predictive Coding Techniques

strategies, dynamically switching between different coding schemes depending on the local statistics of the prediction errors. This ensures that no-data regions and their surrounding areas are encoded efficiently, even when the underlying data distribution varies. The combination of predictive coding with GR codes offers a lightweight solution for hyperspectral data compression, particularly in resource-constrained environments, such as satellite imaging or onboard sensor systems, where minimizing transmission size and computation is critical [Cheng et al., 2014, Ahanonu et al., 2019].

2.4.7 Deep Learning-Based Approaches

Deep learning is increasingly adopted for hyperspectral image processing, leveraging complex network architectures to enhance compression. Recent advancements in hyperspectral image classification [Marmanis et al., 2015, Li et al., 2016, Kussul et al., 2017] and feature extraction [Romero et al., 2015, Zhao and Du, 2016, Chen et al., 2016] have demonstrated the potential of deep learning techniques. Paper [Marmanis et al., 2015] explored the use of deep learning for hyperspectral image classification by leveraging convolutional neural networks (CNNs). Their approach utilized pre-trained networks to extract hierarchical spatial features, improving classification performance for remote sensing applications. The method demonstrated the effectiveness of CNNs in learning discriminative features, even with limited labeled hyperspectral data.

Deep learning has been widely explored for hyperspectral image classification, with various methods leveraging both spectral and spatial information for feature extraction. The following studies highlight different approaches that demonstrate the effectiveness of deep learning in hyperspectral analysis. Li and colleagues [Li et al., 2016] proposed a stacked autoencoder framework for hyperspectral image classification. By learning deep feature representations, their method captured both spectral and spatial information, leading to enhanced classification accuracy. The study highlighted how unsupervised pre-training can address the lack of labeled hyperspectral data.

The research by Kussul *et al.* [Kussul et al., 2017] integrated deep learning models with ensemble techniques for hyperspectral image classification. They focused on combining the outputs of multiple neural networks to boost robustness and accuracy. Their results showed that deep ensemble methods outperform traditional classifiers on complex hyperspectral datasets.

The research by Romero *et al.* [Romero et al., 2015] introduced an unsupervised feature extraction method for hyperspectral data using deep learning. The method involved training deep architectures to discover meaningful feature representations, even without labeled samples, enabling more effective clustering and classification. Zhao and colleagues [Zhao and Du, 2016] proposed a spectral-spatial deep learning model that incorporates both spectral and spatial information for feature extraction. Their framework outperformed existing methods by capturing spatial dependencies and local spectral patterns in hyperspectral images, leading to superior classification results.

Chen *et al.* [Chen et al., 2016] developed a deep CNN model tailored for feature extraction in hyperspectral image processing. Their approach emphasized spatial feature learning through convolutional layers, which improved classification performance on benchmark datasets by effectively handling the high-dimensional nature of hyperspectral data. These works provide strong evidence that deep learning-based methods are extensively used for hyperspectral image classification and feature extraction. They illustrate how various deep architectures, including autoencoders, CNNs, and ensemble techniques, have improved classification accuracy by capturing spectral-spatial dependencies and handling high-dimensional data.

In particular, a Google research team has proposed a method for compressing image data using a deep neural network [Toderici et al., 2017]. The authors propose a model that employs a deep convolutional neural network (CNN) to analyze and compress images. The architecture is designed to learn hierarchical representations of the data, which allows for more effective encoding of image features compared to conventional methods. One of the standout aspects of this approach is its end-to-end learning capability. The model can be trained directly on raw image data, enabling it to learn optimal compression strategies based on the specific characteristics of the images being processed. This results in a more tailored compression mechanism that can adapt to various image types and contents.

The research emphasizes the importance of maintaining reconstruction quality while achieving high compression rates. By utilizing the learned representations, the model can generate high-fidelity reconstructions of the original images, even at lower bit rates. This is particularly beneficial for applications where preserving visual quality is crucial. The method incorporates an iterative process that progressively refines the compressed representation. This approach not only enhances compression efficiency but also allows the model to focus on the most significant features of the image, ensuring that essential details are preserved during the compression process.

The authors conduct extensive experiments to evaluate the performance of their proposed method against existing compression techniques. The results demonstrate that their deep learning-based approach outperforms traditional methods in terms of both compression efficiency and visual quality of the reconstructed images. The research conducted in this paper offers a significant contribution to the field of image compression by employing deep neural networks to enhance compression efficiency while maintaining high reconstruction quality. This approach marks a shift towards adaptive methods of handling image data, which may lead to new developments in various imaging technologies. Although deep learning-based compression techniques show promise, their computational complexity and larger model sizes can limit practical implementation in hyperspectral image compression.

A study proposed the SSCNet, a fully convolutional autoencoder, for spectral signal compression [La Grassa et al., 2022]. SSCNet demonstrated superior performance over traditional methods like JPEG2000 in both PSNR and SSIM metrics, highlighting its ability to preserve high reconstruction quality across hyperspectral datasets such as the VIRTIS-Rosetta dataset. By replacing the traditional fully connected layers with convolutional layers, SSCNet reduces model complexity, improving training efficiency and making it more suitable for high-dimensional data like hyperspectral images (MDPI).

The HyCoT model introduces a transformer-based architecture for hyperspectral image compression [Fuchs et al., 2024]. It employs multi-head self-attention to capture long-range dependencies within spectral data and adapts efficiently to a variety of compression ratios. The model's lightweight decoder also makes it practical for real-time applications, offering competitive rate-distortion performance compared to other state-of-the-art methods like convolutional autoencoders [Fuchs et al., 2024].

Real-time hyperspectral image compression has become critical for remote sensing applications using embedded GPUs, especially in satellite systems [Fuchs and Demir, 2023]. Efficient compression algorithms aim to balance speed and accuracy by leveraging the parallel processing capabilities of GPUs, enabling fast compression without sacrificing reconstruction quality. This approach is ideal for handling large-scale data streams from airborne and spaceborne sensors where transmission speed and storage space are limited [Fuchs and Demir, 2023].

A recent study proposed a novel hybrid approach that integrates convolutional neural networks (CNNs) with transformers to harness both local and non-local feature extraction capabilities [Liu et al., 2023]. These models can efficiently compress hyperspectral data while maintaining high fidelity, offering a more effective solution for complex tasks like anomaly detection and classification. This hybrid method represents a promising direction for task-aware hyperspectral image compression by combining the strengths of both architectures [Liu et al., 2023]. The HySpecNet-11k dataset has been used to benchmark various learning-based hyperspectral image compression models. The results indicate that modern deep-learning techniques can achieve high compression ratios with minimal loss in reconstruction quality. These techniques use task-specific compression strategies that adjust dynamically to the spatial or spectral properties relevant to the task at hand, showcasing how adaptive methods can outperform traditional compression approaches [Fuchs and Demir, 2023, Liu et al., 2023].

Table 2.7 summarizes recent developments in Deep Learning-based techniques for hyperspectral image compression. These approaches employ advanced neural network architectures to automatically learn compact representations of hyperspectral images. For instance, Kumar *et al.* [Kumar et al., 2018] propose an onboard hyperspectral image compression technique that utilizes compressed sensing and deep learning to meet the limited hardware constraints of space missions. In this study, the onboard technique refers to real-time hyperspectral image compression and processing performed directly on satellites or airborne sensors before data transmission, reducing bandwidth and storage requirements. This method involves a lightweight encoder onboard satellites and a deep learning-based decoder on the ground, enabling efficient data reconstruction with reduced transmission requirements.

Table 2.7: Related Work on Deep Learning-Based Approaches

Year	**Paper**	**Details**	**Datasets**
2018	Onboard hyperspectral image compression using compressed sensing and deep learning [Kumar et al., 2018]	Proposes a real-time onboard compression scheme.	Pavia

Continued on next page

Table 2.7 – continued from previous page

Year	Paper	Details	Datasets
2019	Hyperspectral image compression and super-resolution using tensor decomposition learning [Aidini et al., 2019]	Proposes a low-rank tensor completion scheme for image compression.	EUROSAT
2020	Autoencoder-based dimensionality reduction and classification using convolutional neural networks for hyperspectral images [Ramamurthy et al., 2020]	Uses convolutional neural networks for HSI compression.	Pavia
2021	Convolution Neural Network based lossy compression of hyperspectral images [Dua et al., 2021]	Proposes a lossy hyperspectral image compression algorithm based on the concept of autoencoders.	Indian Pines, Pavia University
2021	Learned Hyperspectral Compression Using a Student's T Hyperprior [Guo et al., 2021]	Proposes a trainable network architecture for hyperspectral compression.	Pavia, Cave

Continued on next page

Table 2.7 – continued from previous page

Year	Paper	Details	Datasets
2022	Hyperspectral Data Compression Using Fully Convolutional Autoencoder [La Grassa et al., 2022]	Proposes a spectral signals compressor network based on deep convolutional autoencoder (SSCNet).	VIRTIS-Rosetta
2022	Edge-Guided Hyperspectral Image Compression With Interactive Dual Attention [Guo et al., 2022]	Proposes an HSI compression network embedded with an edge extraction network, which can be jointly optimized.	Pavia, Cave
2022	Hyperspectral image compression using convolutional neural networks with local spectral transforms and non-uniform sample normalization [Verdú et al., 2022]	Utilizes transposed deconvolution in convolutional neural networks to produce significant artifacts in decompressing low-variance images.	AVIRIS
2023	Reconstruction of Compressed Hyperspectral Image Using SqueezeNet Coupled Dense Attentional Net [Mohan et al., 2023]	Addresses image denoising alongside the compression and reconstruction of hyperspectral images using deep learning techniques.	AVIRIS

Continued on next page

Table 2.7 – continued from previous page

Year	Paper	Details	Datasets
2023	Hyperspectral Image Compression via Cross-Channel Contrastive Learning [Guo et al., 2023]	Proposes a hyperspectral compression network via contrastive learning (HCCNet).	Pavia, Cave
2023	HySpecNet-11k: a Large-Scale Hyperspectral Dataset for Benchmarking Learning-Based Hyperspectral Image Compression Methods [Fuchs and Demir, 2023]	Presents HySpecNet-11k, a large-scale hyperspectral benchmark dataset.	HySpecNet-11k
2024	HyCoT: A Transformer-Based Autoencoder for Hyperspectral Image Compression [Fuchs et al., 2024]	Proposes Hyperspectral Compression Transformer (HyCoT), a transformer-based autoencoder for pixelwise HSI compression.	HySpecNet-11k

Ramamurthy *et al.* [Ramamurthy et al., 2020] developed a CNN-based approach for dimensionality reduction in hyperspectral images, enhancing both compression and classification performance. Their model extracts relevant spectral and spatial features, improving downstream analytics tasks.

Recent work by Guo *et al.* [Guo et al., 2021] introduces a Student-t hyperprior model for hyperspectral image compression, combining probabilistic modeling with trainable neural networks. Their method achieves efficient compression by capturing the latent distribution of hyperspectral data, providing superior performance compared to conventional approaches. These deep learning techniques provide powerful tools for managing the high-dimensional nature of hyperspectral data by automatically capturing key features and reducing redundant information.

Our work falls under the category of deep learning-based hyperspectral image compression methods. Unlike conventional approaches that rely on handcrafted compression techniques, we leverage deep neural networks to learn compact and task-relevant representations of hyperspectral data. Our method integrates implicit neural representations (INRs), which enable adaptive encoding by focusing on spectral and spatial features that are most critical for downstream tasks such as classification, target detection, and anomaly detection. By utilizing these learning-based techniques, our approach ensures that essential information is preserved while achieving high compression efficiency. This aligns with recent advancements in deep learning-based hyperspectral compression, demonstrating the potential of neural networks in optimizing storage and transmission while maintaining reconstruction quality.

2.4.8 Task-aware Hyperspectral Image Compression

Task-aware compression has emerged as an innovative approach to address the limitations of traditional hyperspectral image (HSI) compression methods, which often fail to prioritize specific information essential for certain tasks, such as classification or target detection. Traditional compression methods, whether lossless or lossy, typically apply uniform compression across all parts of an image without considering the end task, potentially leading to unnecessary loss of critical information. Task-aware compression, on the other hand, introduces an adaptive methodology where regions or spectral bands of the image that are more critical to the task at hand are preserved with higher fidelity, while less relevant regions undergo more aggressive compression.

One key aspect of task-aware compression is region-aware processing, where different regions of the image are compressed at different rates based on their importance for the task. For example, in target detection tasks, regions containing the target objects are compressed with minimal loss, while other regions are subjected to greater compression. This region-based strategy can be extended to spectral bands in hyperspectral data. Hyperspectral images contain hundreds of contiguous spectral bands, not all of which contribute equally to every task. For instance, certain spectral bands may be more crucial for identifying specific materials, while others can be compressed more heavily without affecting the task performance.

Deep learning methods have further advanced task-aware compression. Neural networks, particularly convolutional neural networks (CNNs) and transformers, can be trained to learn which spatial or spectral features are most important for a given task and to apply adaptive compression accordingly. These methods use supervised or unsupervised learning techniques to minimize the impact of compression on task-specific metrics such as classification accuracy or detection precision. Implicit neural representations (INRs) have also been explored, encoding hyperspectral data as continuous functions that can be adaptively sampled and reconstructed based on task-specific requirements [Yu, 2024, Ye et al., 2023].

Ye *et al.* [Ye et al., 2023] demonstrated the effectiveness of task-aware compression for image restoration tasks by optimizing compression parameters for super-resolution and denoising, ensuring that critical regions maintain high fidelity.

2.4.8.1 Challenges and Future Directions

While task-aware compression has shown significant promise, several challenges remain to fully realize its potential. One of the main challenges is the computational complexity involved in adapting compression parameters based on task-specific requirements. While effective deep learning-based methods require extensive computational resources for both training and inference, in hyperspectral image compression, the number of spectral bands and the spatial resolution contribute to enormous data size, making real-time or near-real-time processing difficult, especially for applications with constrained hardware resources, such as satellite imaging [Rezasoltani and Qureshi, 2023b].

Another challenge is generalization across different tasks. Most task-aware methods are designed with a specific task in mind, such as classification or anomaly detection, but these methods may not perform well when applied to other tasks. For example, a compression model optimized for classification may not be as effective for target detection or spectral unmixing, and vice versa. Developing generalized models that can adapt to multiple tasks without sacrificing performance remains an area of active research [Fuchs et al., 2024].

There is also the issue of data availability. Many deep learning-based methods require large amounts of labeled data for training, which is often not available for hyperspectral images. Acquiring ground-truth labels for hyperspectral data is both time-consuming and costly, limiting the scalability of task-aware methods. Techniques such as transfer learning and active learning have been explored as potential solutions to mitigate the data bottleneck, but these approaches also come with their own challenges [Fuchs et al., 2024, Yu, 2024].

Several research directions could enhance the effectiveness and applicability of task-aware hyperspectral image compression. Hybrid compression techniques, combining traditional and learning-based methods, could offer a balance between computational efficiency and task-specific adaptability. Additionally, hardware-accelerated compression approaches that leverage GPUs or specialized hardware like FPGAs could enable real-time processing for spaceborne or airborne sensors, where rapid data transmission and storage are critical [Altamimi and Ben Youssef, 2021].

2.4.8.2 Afterthought

Task-aware hyperspectral image compression represents a significant advancement in the field of remote sensing and data processing. By prioritizing the retention of information most relevant to the task at hand, these methods offer a more efficient solution to the challenges posed by the large size and complexity of hyperspectral data. The integration of deep learning techniques, particularly CNNs, transformers, and INRs, has further enhanced the potential of task-aware compression, enabling adaptive, task-specific data handling that improves performance in applications such as classification, detection, and spectral unmixing.

However, the path forward involves overcoming several key challenges, including computational complexity, the need for task generalization, and the limited availability of labeled data. As research in this area continues to evolve, we can expect to see the development of more efficient, scalable, and task-adaptive compression techniques. These advancements will be crucial for unlocking the full potential of hyperspectral imaging in a wide range of applications, from environmental monitoring and agriculture to military surveillance and space exploration.

By refining and expanding these methods, researchers can contribute to the growing demand for hyperspectral data processing solutions that are both efficient and adaptable to the specific needs of various tasks. Task-aware hyperspectral image compression stands poised to play a central role in the future of remote sensing and beyond.

2.5 Summary

In this chapter, we reviewed various approaches to hyperspectral image compression, spanning from classical transform-based and predictive coding techniques to modern deep learning-driven methods. Among these, deep learning-based methods have shown significant promise in efficiently compressing hyperspectral images while maintaining high reconstruction quality. These methods leverage advanced neural architectures to learn compact, task-relevant representations, which can be highly beneficial in preserving critical spectral and spatial information. However, computational complexity remains a major challenge for real-time applications.

Furthermore, we discussed the emerging paradigm of task-aware hyperspectral image compression, which tailors compression strategies to the specific needs of downstream tasks such as classification and target detection. Unlike traditional methods that apply uniform compression across all spectral bands and spatial regions, task-aware approaches selectively allocate higher fidelity to important regions while aggressively compressing less relevant areas. This technique enhances the efficiency of hyperspectral imaging applications by preserving the most informative aspects of the data.

Our proposed method integrates deep learning techniques and extends them to task-aware compression. By investigating the central problem of deep learning-based compression, specifically the challenge of compression time, we aim to develop a method that balances compression efficiency and computational cost. The use of implicit neural representations (INRs) in our approach allows for adaptive encoding and improved flexibility in handling hyperspectral data. This method addresses key challenges in hyperspectral image compression while contributing to advancements in task-aware compression strategies.

In summary, deep learning has revolutionized hyperspectral image compression by enabling high-performance encoding and reconstruction techniques. Task-aware compression further refines this process by ensuring that essential spectral and spatial information is preserved for specific applications. Future research directions should focus on optimizing computational efficiency, improving real-time processing capabilities, and developing more adaptive compression frameworks that are generalized to different tasks.

Chapter 3

Benchmarks & Evaluation Metrics

Hyperspectral image compression is a crucial step in efficiently handling hyperspectral datasets while preserving essential spectral and spatial information. Given the high dimensionality of hyperspectral images, effective compression techniques must balance storage efficiency and image fidelity, ensuring that the compressed data remains useful for various downstream applications such as remote sensing, environmental monitoring, and medical imaging. In this chapter, we evaluate our proposed compression approach using widely studied hyperspectral datasets. We present the evaluation metrics used to quantify the compression performance, including both distortion-based metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), as well as compression efficiency metrics like bits-per-pixel-per-band (bpppb). The subsequent sections detail the datasets utilized for evaluation and describe the quantitative measures employed to assess the quality of compression.

3.1 Datasets

We have used four datasets to evaluate our approach: Indian Pines, Jasper Ridge, Pavia University, and Cuprite. We have chosen these datasets since they have been utilized in prior research on hyperspectral image compression. The Indian Pines, Cuprite, and Jasper Ridge datasets were acquired using NASA's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. The AVIRIS sensor collects geometrically coherent spectroradiometric data for the characterization of the Earth's surface. The Pavia University dataset was acquired via the ROSIS (Reflective Optics System Imaging Spectrometer).

The Indian Pines dataset, which was collected in 1992, is one of the most widely used hyperspectral images for remote sensing and classification research. It consists of 145 × 145 pixels with 220 spectral bands covering the 0.4˘2.5 μm wavelength range in the visible and infrared spectrum. The dataset was collected in 1992 using NASA's AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) sensor over Northwest Indiana, USA. This region is primarily agricultural land, with fields of corn, soybeans, wheat, alfalfa, and pastures interspersed with forested areas, roads, and urban structures. The dataset contains 16 ground-truth classes representing different types of crops and land covers, such as corn-no-till, corn-minimum till, corn-clean, soybean-no-till, soybean-minimum till, soybean-clean, wheat, woods, buildings, and grass pastures. Due to its diverse land cover, the Indian Pines dataset is widely used in hyperspectral image classification, anomaly detection, and compression research.

The Jasper Ridge dataset, which was obtained in 1998, is another well-known hyperspectral benchmark with dimensions 100 × 100 pixels and 224 spectral bands covering the 0.4˘2.5 μm wavelength range. Like Indian Pines, it was also acquired using the AVIRIS sensor, but it covers an area in Jasper Ridge Biological Preserve in California, USA. This region is known for its diverse natural vegetation, including grasslands, chaparral, deciduous forests, and mixed coniferous forests. Additionally, the dataset contains geological formations and soil types, making it valuable for both ecological monitoring and geological analysis. Jasper Ridge is commonly used in research related to land cover classification, species mapping, and ecosystem studies.

The Pavia University dataset, which was recorded more recently in 2001, is a hy-

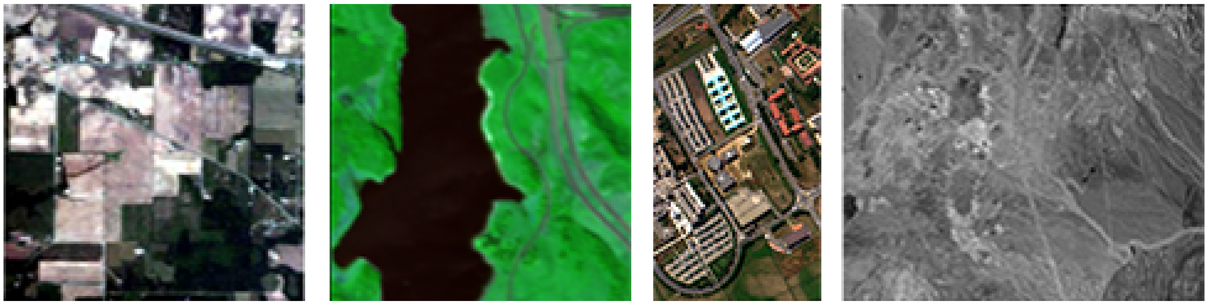


Figure 3.1: The datasets used in this thesis are represented using pseudo-colors. The following datasets are referred to as L2R: Indian Pines, Jasper Ridge, Pavia University, and Cuprite.

perspectral image captured by the ROSIS-03 (Reflective Optics System Imaging Spectrometer) sensor during an airborne campaign conducted by the German Aerospace Centre (DLR) over the University of Pavia, Italy. The dataset has dimensions 610 × 340 pixels with 103 spectral bands, covering the 0.43˘0.86 μm spectral range in the visible and near-infrared (VNIR) regions. The image primarily consists of urban landscapes, including buildings, asphalt roads, meadows, gravel surfaces, trees, metallic objects (e.g., rooftops), shadows, and water bodies. It is often used for urban land cover classification, spectral unmixing, and target detection.

The Cuprite dataset, which was captured in 1997, is a well-known hyperspectral image in geological mapping and mineral exploration studies. It was captured using the AVIRIS sensor over the Cuprite mining district in Nevada, USA, and has a spatial resolution of 614 × 512 pixels with 224 spectral bands. The Cuprite region is famous for its rich mineralogical diversity, and the dataset contains abundant hydrothermally altered minerals, including kaolinite, alunite, buddingtonite, hematite, calcite, muscovite, montmorillonite, and quartz. This dataset is widely used for mineral classification, geological surveys, and remote sensing applications in Earth sciences.

Figure 3.1 provides a visual representation of the four datasets used in this study: Indian Pines, Jasper Ridge, Pavia University, and Cuprite. Each dataset is depicted using pseudo-color images to highlight their unique spectral and spatial characteristics.

The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) is one of the most widely used instruments for acquiring high-quality hyperspectral data from airborne platforms. Developed by NASA's Jet Propulsion Laboratory, AVIRIS employs a "whisk broom" scanning mechanism and captures reflectance data across 224 contiguous spectral bands, ranging from the visible to shortwave infrared (400–2500 nm). The instrument uses Silicon (Si), Indium Gallium Arsenide (InGaAs), and Indium Antimonide (InSb) detectors to cover the visible, near-infrared, and shortwave infrared regions, respectively. With a 34-degree field of view and 1 milliradian instantaneous field of view (IFOV), AVIRIS provides fine spatial and spectral resolution critical for remote sensing applications. It features liquid nitrogen-cooled detectors to reduce thermal noise and maintains a channel bandwidth calibrated to within 1 nm.

Figure 3.2 illustrates the internal structure and configuration of the AVIRIS sensor, highlighting its key components, including the spectrometer, scan drive, tape recorder, and optical fibers. This level of instrumentation allows AVIRIS to collect detailed hyperspectral data suitable for environmental monitoring, mineral mapping, vegetation analysis, and more.

In Chapter 5, in addition to standard benchmark datasets such as Indian Pines, Pavia University, Jasper Ridge, and Cuprite, we evaluate our method on a large-scale hyperspectral dataset acquired by a high-resolution airborne imaging spectrometer. This dataset, referred to as Strip 4, contains 6708×4192 spatial pixels and 270 spectral bands, and was acquired as part of an internal aerial hyperspectral imaging project. The data was collected using a high-resolution pushbroom sensor during a high-altitude flight.

This large dataset is not publicly hosted, but it was made available to our lab through a collaborative agreement. Upon request, and with proper authorization, researchers may access the dataset. A visual representation of this dataset is provided in Figure 3.3, where a single representative band is shown.

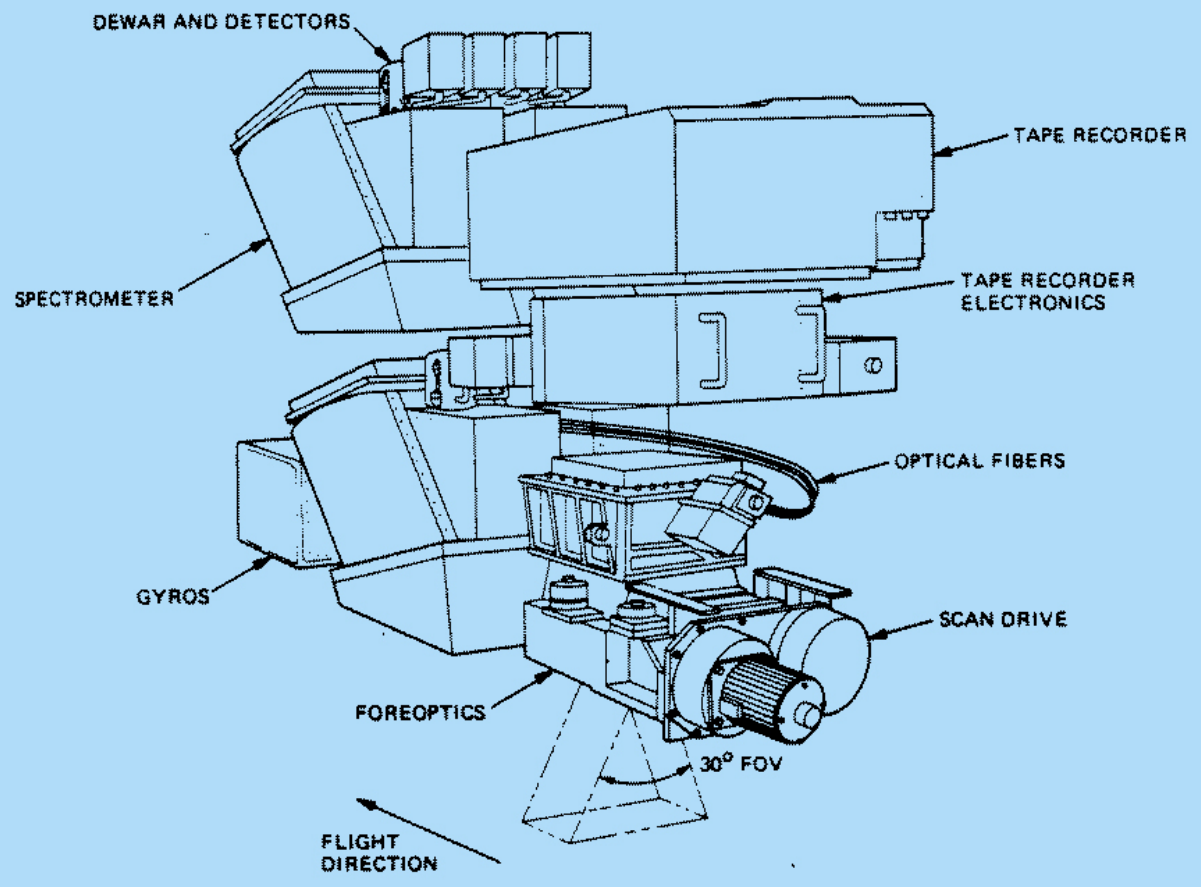


Figure 3.2: Schematic diagram of the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) instrument. Key components include the spectrometer, scan drive, optical fibers, gyros, and the tape recorder system. The "whisk broom" scanning mechanism enables the acquisition of hyperspectral data across a wide field of view with high spectral resolution. (Image source: NASA/JPL AVIRIS instrument documentation.)

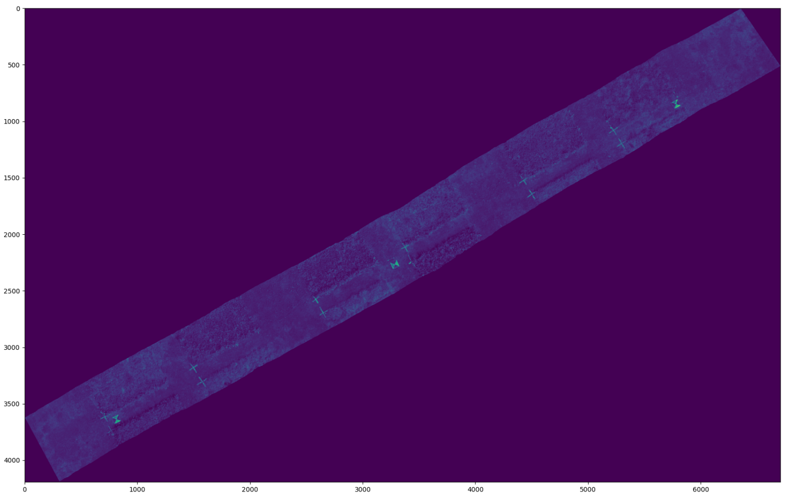


Figure 3.3: Visualization of the dataset used in this study, consisting of images with dimensions 4192 by 6708 pixels and 270 channels. The dataset has a total size of 28.2 GB.

3.2 Metrics for Measuring Compression Quality

To evaluate the performance of the proposed hyperspectral image compression method, we employ three widely used metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Bits-Per-Pixel-Per-Band (bpppb). These metrics allow for a quantitative comparison between the original and compressed images, ensuring that compression achieves a balance between high quality and efficient storage.

3.2.1 Peak Signal-to-Noise Ratio (PSNR)

Similar to previous studies, we use the Peak Signal-to-Noise Ratio (PSNR) to compare the proposed method with other methods. PSNR, measured in decibels, is a frequently used metric in image compression. It measures the difference in “quality” between the original image and its compressed version. Higher PSNR values suggest that the compressed image is more similar to the original image, i.e., the compressed image preserves more of the information present in the original image and that it has higher quality.

PSNR is computed using the Mean Squared Error (MSE), which measures the cumulative pixel-wise error between the original and compressed images. Lower MSE values indicate better reconstruction quality.

MSE is computed as follows:

$$MSE = \sum_{i} \frac{|I[i] - \tilde{I}[i]|^2}{i}, \tag{3.1}$$

where I and $\tilde{I}$ denotes the original and compressed images, respectively, and i indices over the pixels. MSE is used to calculate PSNR as follows:

$$PSNR = 10 \ \log_{10} \left(\frac{R^2}{MSE} \right), \tag{3.2}$$

where R is the largest variation in the input image in Equation 3.2. For instance, R is 1 if the input image is of double-precision floating-point data. R is 255, for instance, if the data is an 8-bit unsigned integer image.

3.2.2 Structural Similarity Index (SSIM)

Another metric that is commonly used for image compression studies is the structural similarity index (SSIM) [Wang et al., 2004]. SSIM accounts for the structural information that the human visual system is naturally attuned to. Therefore, SSIM is perceptually more meaningful than MSE, at least for RGB images, which treat pixels independent of each other. SSIM metric combines luminance, contrast, and structure, and the channel-wise SSIM is computed as follows:

$$SSIM_{\text{channel-wise}} = \frac{(2\,\mu_I\,\mu_{\tilde{I}}\ +\ C_1)\ (2\,\sigma_{I\tilde{I}}\ +\ C_2)}{(\mu_{\tilde{I}}^2\ +\ \mu_{\tilde{I}}^2\ +\ C_1)(\sigma_{\tilde{I}}^2\ +\ \sigma_{\tilde{I}}^2\ +\ C_2)}, \tag{3.3}$$

where μ_I and $\mu_{\tilde{I}}$ are mean intensity values for the original image and its reconstruction, respectively. Similarly, $\sigma_{\tilde{I}}^2$ is the variance of the original image and $\sigma_{I\tilde{I}}$ is the covariance value for the original image and its reconstruction. $C_1 = (k_1.L)^2$ and $C_2 = (k_2.L)^2$ are there to address division by a weak denominator. L, here, denotes the dynamic range of a pixel and $K_1 = 0.01$ and $K_2 = 0.03$. Dynamic range L is typically expressed as $2^{(\text{bits per pixel})}$.

Mean SSIM value is computed by averaging channel-wise SSIM values. SSIM values range between 0 and 1, and larger SSIM values indicate a better reconstruction. While PSNR captures the numerical accuracy of the reconstruction, SSIM measures its perceptual quality. In the context of hyperspectral image compression, where both signal fidelity and structural integrity are important, using both PSNR and SSIM offers a more comprehensive evaluation of compression performance.

3.2.3 Bits-Per-Pixel-Per-Band (bpppb)

In addition, the number of bits-per-pixel-per-band (*bpppb*) captures the level of compression achieved by a model. Lower values of *bpppb* indicate higher compression rates. The value of *bpppb* for an uncompressed HSI is 8 or 32, depending on how the pixels are stored. It is common to store the HSI pixel value (for each channel) as a 32-bit floating point. The parameter *bpppb* is calculated as follows:

$$bpppb = \frac{\#\text{parameters} \times (\text{bits per parameter})}{(\text{pixels per band}) \times \#\text{bands}}. \tag{3.4}$$

This metric is crucial for assessing the trade-off between compression ratio and image fidelity, ensuring that HSI compression achieves the desired balance between storage efficiency and minimal loss of information.

In Equation (3.4), the term parameters refers to the total number of trainable parameters in the model used to represent the hyperspectral image. The bits per parameter indicate the numerical precision used to store each parameter (e.g., 32 bits for single-precision floating-point representation). The pixels per band corresponds to the spatial resolution of each spectral band (i.e., the number of pixels in a single 2D image slice of the hyperspectral cube), while bands denotes the number of spectral bands in the dataset. Note that in hyperspectral imaging, the terms band and channel are often used interchangeably to refer to individual spectral slices of the hyperspectral cube.

3.3 Summary

This chapter discusses the datasets and the metrics used to evaluate our work performance. These datasets and metrics provide a foundation for assessing the effectiveness of the proposed methods. As we analyze and compare our results, we will refer to them in subsequent chapters.

Chapter 4

HSI Compression using INR

Hyperspectral image compression is essential for efficient storage and transmission of high-dimensional spectral data, particularly in applications such as remote sensing, environmental monitoring, and medical imaging. Traditional compression techniques, including transform-based methods and predictive coding, often struggle to balance high compression rates with minimal loss of information. Recently, implicit neural representations (INRs) have emerged as a promising alternative for compressing high-dimensional data by leveraging neural networks to encode images as continuous functions. In this chapter, we introduce an INR-based compression approach that models hyperspectral images using multi-layer perceptrons (MLPs) with periodic activation functions. Unlike conventional compression schemes, this method does not rely on explicit pixel-wise storage but instead learns a neural representation of the hyperspectral data. We describe the theoretical foundations of INR-based compression, detail the architecture of the proposed method, and present experimental evaluations demonstrating its effectiveness compared to traditional and learning-based compression techniques.

4.1 Image Compression using INRs

Let us consider a w-by-h grayscale image. We can represent this image as a function

$$I_{\text{grayscale}} : U \to [0, 1],$$

where

$$U = \{(x, y) \mid x \in \{1, \dots, w\}, y \in \{1, \dots, h\}\}.$$

This notation captures the intuition that an image is a function over a 2d grid that defines the pixel locations. The intensity at each pixel is then $I_{\text{grayscale}}(x, y)$. It is straightforward to extend this notation to hyperspectral image I as follows

$$I = \begin{pmatrix} I_1 : U \mapsto [0, 1] \\ \vdots \\ I_c : U \mapsto [0, 1] \end{pmatrix}. \tag{4.1}$$

Here for the sake of simplicity, we assume a w-by-h hyperspectral image comprising c channels. Using this notation, we can find the spectral signature of the pixel at location $(x, y) \in U$ as follows:

$$\{I_1(x, y), \cdots, I_c(x, y)\}.$$

Given this setup, it is possible to imagine a neural network that models the functions $I_1, \cdots, I_c$. Specifically, work on implicit neural representations suggests using a multilayer perceptron (MLP) with periodic activation functions to represent functions of the form shown in Equation 4.1. Consider an MLP f_Θ with parameters Θ that maps locations in U to pixel spectral signatures:

$$f_\Theta : U \mapsto \{I_1, \cdots, I_c\}.$$

	w	h	c	n_h	w_h	q	Θ
#bits	16	16	16	8	8	1	$bpp \times n_\Theta$

Table 4.1: Disk layout for I_{encoded}. Here q denotes if parameters Θ were quantized at compression time. bpp (or #bits-per-parameter) is either 32 or 16.

Under this regime, training can be defined as

$$\breve{\Theta} = \arg\min_{\Theta} \mathcal{L}(I, f_\Theta),$$

Where $\mathcal{L}$ is a loss function that is differentiable and that captures the error between the original hyperspectral image and the decompressed hyperspectral image. We use the mean-squared error to compute this loss. Given $\breve{\Theta}$, it is possible to reconstruct the original image I by sampling $f_{\breve{\Theta}}$ at the relevant locations. Parameters $\breve{\Theta}$, along with w, h, and c, plus the structure of the MLP, represent an encoding I_{encoded} of the hyperspectral image I that was used to train the MLP f_Θ. It is expected that the memory required to store I_{encoded} is an order of magnitude less than the memory needed to store the hyperspectral image. The decompression process requires constructing the sampling locations U, setting up MLP f_Θ and initializing its weights to $\breve{\Theta}$, and evaluating $f_{\breve{\Theta}}$ at locations in U.

4.2 Compression Pipeline

The proposed compression method consists of two steps. Step 1 performs an architecture search. The goal here is to find an MLP that achieves the highest reconstruction accuracy for a given $bpppb$ budget. Architecture search is performed by overfitting multiple MLPs having different numbers of hidden layers and hidden layers' widths to the hyperspectral image. Architecture search, however, means longer compression times. Step 2 involves quantizing and storing the parameters of the overfitted MLP to disk. The caveat here is that this further reduces the quality of the reconstructed image.

4.2.1 Overfitting a SIREN network

The compression procedure comprises overfitting a SIREN network f_{Θ} to a hyperspectral image I [Dupont et al., 2021]. The width w and height h of the hyperspectral image are used to set up an input location grid on $[-1,+1] \times [-1,+1]$, and the MLP is trained to reconstruct a pixel's spectral signature given its location. The parameters $\breve{\Theta}$ of this overfitted MLP are quantized as $\check{\Theta}$. MLP structure that contains the number of hidden layers n_h, widths of these layers w_h, and the width w, height h, and the number of channels c of the original hyperspectral image I along with $\check{\Theta}$ serve as a compressed encoding I_{encoded} of the hyperspectral image I (Table 4.1). Parameters $\check{\Theta}$ are either stored as 32-bit floats or as 16-bit floats. Training and inference require 32-bit floats, and quantization/dequantization is performed to move between 32 and 16 bits representations. We have yet to try an 8-bit, fixed-point representation for parameters.

The overfitted MLP contains

$$(w_h \times 2) + (w_h \times w_h)^{(n_h - 1)} + (c \times w_h) \tag{4.2}$$

parameters.

4.2.2 Decompressing I_{encoded}

The hyperspectral image is reconstructed from its compressed encoding I_{encoded} as follows: 1) use n_h, w_h, and c to reconstruct f_{Θ}, 2) dequantize $\check{Q}$ to $\tilde{\Theta}$ and use it to initialize the parameters of f_{Θ}, 3) use the width w and height h to set up the input grid between $[-1,+1] \times [-1,+1]$, and 4) evaluate $f_{\tilde{\Theta}}$ at each location in the input grid to reconstruct the image $\tilde{I}$.

4.3 Algorithm for Hyperspectral Image Compression Using INR

This section presents the detailed algorithm used for hyperspectral image compression using implicit neural representations (INRs). The algorithm involves training a multilayer perceptron (MLP) with periodic activation functions to overfit a hyperspectral image, quantizing the learned parameters, and storing them efficiently to allow fast decompression. Below, we provide a high-level pseudocode that outlines the core steps involved in the method.

4.3.1 Pseudocode of the Proposed Method

Algorithm 1 HSI Compression Using Implicit Neural Representations (INR)

```
Input: Hyperspectral image $I$ with dimensions $(w, h, c)$
Output: Compressed encoding $I_{\text{encoded}} = \{\check{\Theta}, n_h, w_h, w, h, c\}$
Step 1: Architecture Search (Using a Subset of the Image)
Uniformly sample a fixed, non-overlapping subset $S \subset U$ (e.g., 5% of pixel coordinates), without replacement
for each candidate MLP structure $(n_h, w_h)$ do
    Initialize MLP $f_\Theta$ with sinusoidal activations; Train $f_\Theta$ to minimize loss $\mathcal{L}(I, f_\Theta)$ using only $S$
    Compute PSNR for the reconstructed subset $\tilde{I}_S$
    if PSNR meets the required threshold then
        Select the best $(n_h, w_h)$ for the final MLP; Break
Step 2: Model Fitting and Compression
Train the selected MLP $f_\Theta$ on the entire hyperspectral image $I$; Quantize parameters $\breve{\Theta} \rightarrow \check{\Theta}$ (32-bit to 16-bit)
Store the compressed encoding $I_{\text{encoded}} = \{\check{\Theta}, n_h, w_h, w, h, c\}$
Step 3: Decompression
Load $\check{\Theta}$ and dequantize to $\tilde{\Theta}$; Initialize MLP $f_{\tilde{\Theta}}$ with dequantized parameters
Generate input grid $U$ for $(w, h)$ within $[-1, 1] \times [-1, 1]$
for each coordinate $(x, y) \in U$ (total: $w \times h$ locations) do
    Compute pixel value $y = f_{\tilde{\Theta}}(x, y)$
Reconstruct the hyperspectral image $\tilde{I}$
```

The above algorithm describes the full pipeline for hyperspectral image compression and decompression. In the following, we explain each stage of the algorithm to provide a deeper understanding of the workflow and practical implications of the method.

- Architecture Search: In the first step, we perform a search over different MLP architectures with varying numbers of hidden layers and layer widths. Each candidate MLP is trained on a fixed subset S of the hyperspectral image. The subset S is selected by uniformly sampling a fixed proportion of the pixel coordinates from the full spatial grid U, without replacement. This sampling strategy ensures computational efficiency during architecture search while preserving a representative view of the spatial and spectral variability of the image. The same subset S is used across all candidate architectures, ensuring consistent evaluation and avoiding repetition or overfitting to specific regions.

- Model Fitting and Compression: Once the best architecture is chosen, the MLP is trained on the entire hyperspectral image. The resulting parameters $\breve{\Theta}$ are then quantized to a lower precision (e.g., from 32-bit floats to 16-bit). This quantization helps reduce the storage size while maintaining acceptable reconstruction quality.

- Decompression: During decompression, the quantized parameters are loaded and dequantized back to 32-bit precision. The MLP is then used to predict pixel values at each coordinate of the input grid, reconstructing the hyperspectral image with minimal distortion.

This pseudocode encapsulates the essence of our approach, demonstrating how a complex hyperspectral image can be effectively compressed using neural network models. The structure of the MLP and the quantization scheme are essential elements that influence both the compression ratio and the reconstructed image quality.

4.3.2 Discussion

This algorithm offers several advantages over traditional methods, including efficient storage through parameter quantization and fast decompression through neural representations. However, the architecture search stage can be time-consuming, and there is a trade-off between compression time and reconstruction quality. Future research could explore ways to optimize the search process and further enhance storage efficiency.

By following this algorithm, we achieve a robust and scalable method for hyperspectral image compression, making it well-suited for applications such as remote sensing, environmental monitoring, and medical imaging.

4.4 Ablative Study

Similar to previous studies, as mentioned in the chapter 3, we use Peak Signal-to-Noise Ratio (PSNR) to compare the proposed method with JPEG, JPEG2000, and PCA-DCT approaches for hyperspectral image compression. JPEG method for hyperspectral image compression uses JPEG standard to encode each channel (band) separately. JPEG2000, instead, uses the JPEG2000 standard for encoding the hyperspectral image. It, too, treats each channel separately. PCA-DCT uses PCA-based analysis to reduce the number of channels, followed by a DCT-based method for encoding these channels. PCA-DCT method posts low signal-to-noise ratios; however, this can be fixed somewhat by keeping more of the original channels. We have chosen these hyperspectral image compression techniques since they are widely used for reducing the size of hyperspectral data in hyperspectral analysis pipelines. In the second part, we also compare our results with JPEG, JPEG2000, PCA-DCT, PCA-JPEG2000, MPEG, X264, X265, PCA-X264, PCA-X265, FPCA-JPEG2000, 3D-DCT, 3D-DWT-SVR, WSRC, HEVC, RPM, and 3D-SPECK.

The datasets used to evaluate our approach, along with their detailed descriptions, are provided in Chapter 3. These datasets—Indian Pines, Jasper Ridge, Pavia University, and Cuprite—have been widely utilized in prior studies on hyperspectral image compression due to their diverse spectral and spatial properties. The datasets were collected using advanced sensors such as NASA's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) and the ROSIS-03 aerial instrument, supporting research across various domains, including agriculture, land use, environmental monitoring, and geoscience.

Figure 4.1 plots PSNR vs. bpppb for the four datasets that we are using in this work. The plots confirm our intuition that higher $bpppb$ leads to better compression quality as measured by PSNR values. For our method, $bpppb$ calculations do not include the storage required to keep network structures.

4.4.1 Architecture Search

Given an image and our (MLP) parameter budget, which is measured in bits per pixel per band, or $bpppb$ for short, the first goal is to select the MLP structure—i.e., the number of hidden layers and their widths—that is able to represent this image with an acceptable PSNR value. Figure 4.2 shows PSNR values achieved for different architectures for the four datasets having a fixed $bpppb$ budget. This suggests that network structure, in addition to network capacity, affects how well a network represents the hyperspectral image.

MLP structure is chosen via hyperparameter search, which involves training feasible designs containing the right number of hidden layers having the correct width on a given hyperspectral image. The result of this process is a single MLP that is able to reconstruct the hyperspectral image with the desired PSNR value. The parameters of the final MLP are then quantized to 16-bit precision, which leads to further savings in terms of the storage needed to represent the hyperspectral image. Our experiments

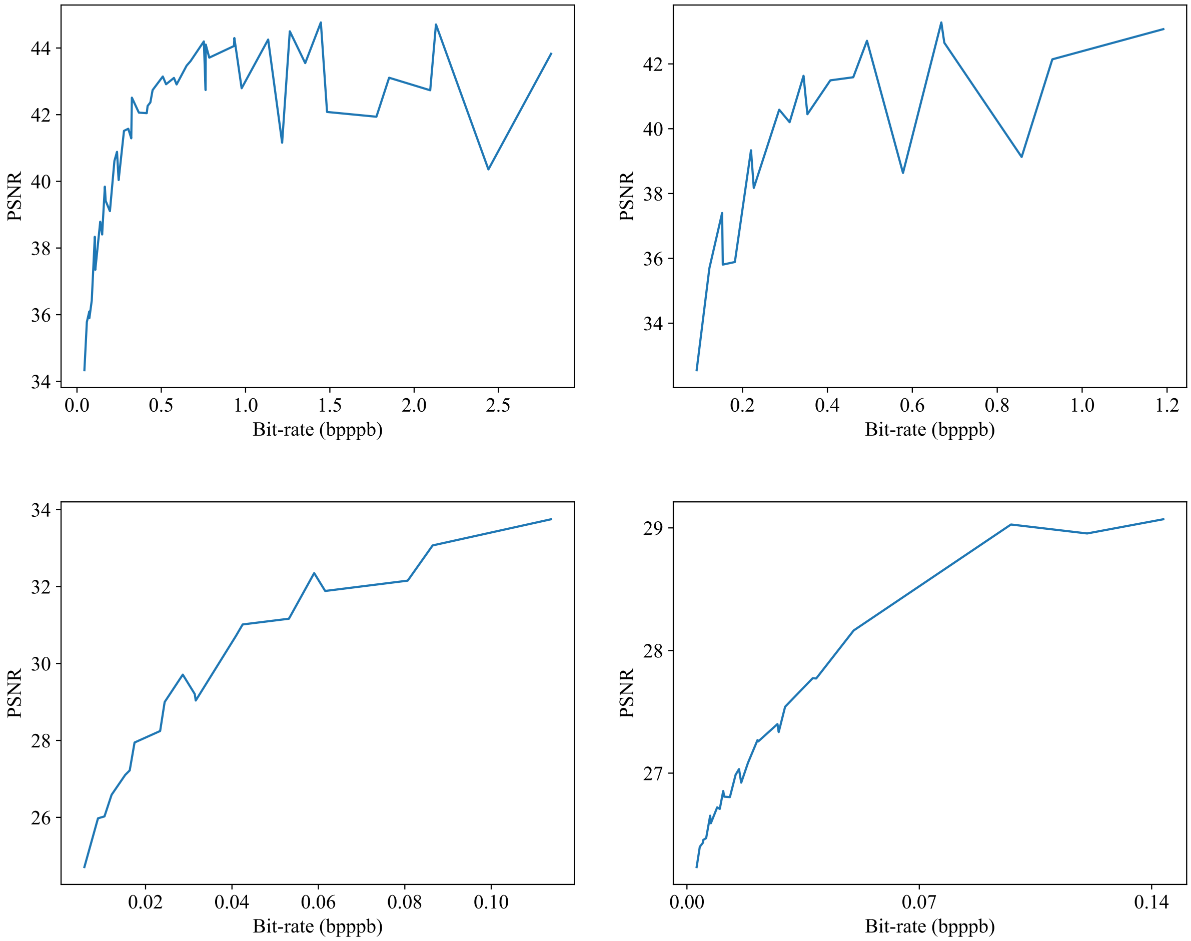


Figure 4.1: **Model capacity**. PSNR vs. *bpppb* for (L2R) Indian Pines, Jasper Ridge, Pavia University, and Cuprite datasets. The trend of these plots confirms our intuition that PSNR values increase as *bpppb* numbers are increased. The plots are not monotonically non-decreasing due to the stochastic nature of MLP overfitting.

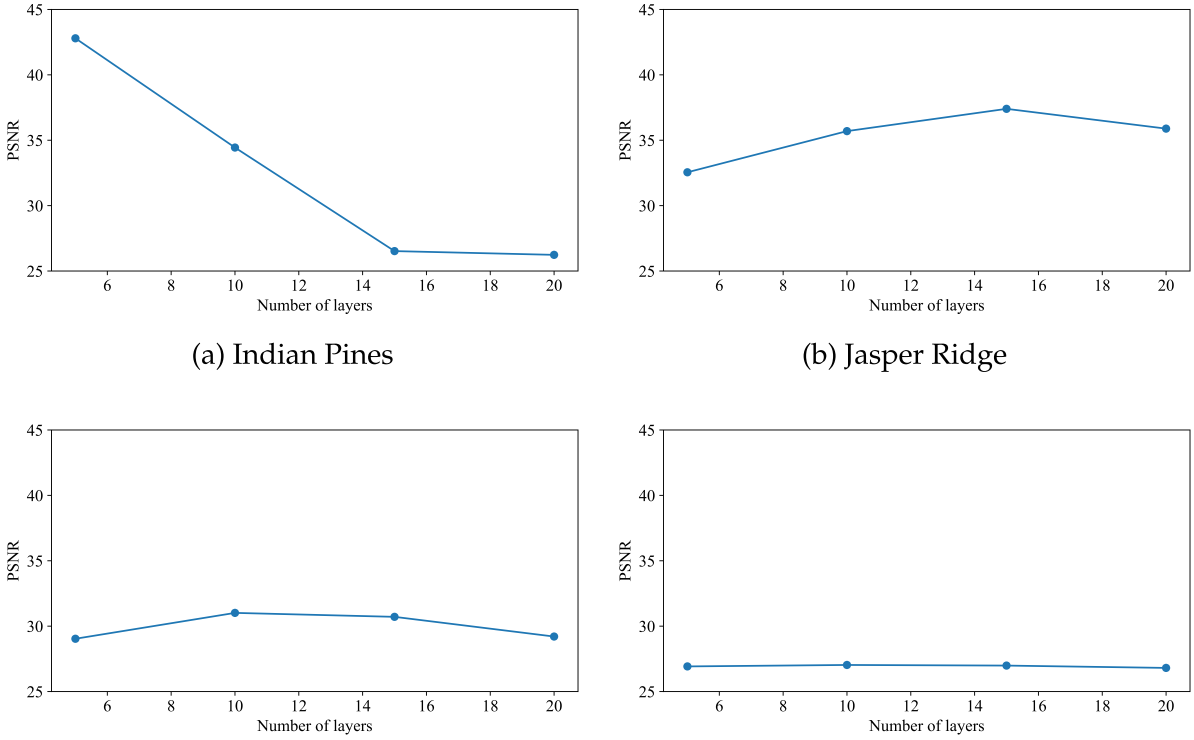
suggest that reducing the MLP parameters from 32-bit to 16-bit precision did not increase distortion and that it had little effect on the signal-to-noise ratio.

4.4.2 Comparison with other methods

Figure 4.3 shows PSNR values at various compression rates for different methods. Specifically, we compare our approach, labeled as *ours* and *hp_ours*, with JPEG, JPEG2000, and PCA-DCT methods. Here, *ours* method stores MLP weights as 32-bit floating point values, whereas *hp_ours* stores MLP weights at half-precision as 16-bit floating point values that are constructed by quantizing the MLP weights. These plots illustrate that our methods achieve higher compression quality, i.e., better PSNR, for a given value of $bpppb$. This is especially true for lower $bpppb$ values.

For the Indian Pines dataset, *ours* method achieves better PSNR up to around 0.7 $bpppb$, at which point JPEG2000 obtains better PSNR. What is curious is that the PSNR for *hp_ours* drops drastically at around 0.4 $bppb$. This merits further investigation. One possible explanation is the effect of weight quantization. In our implementation, we quantize the network weights from 32-bit to 16-bit floating-point precision to reduce storage. However, this quantization may result in a loss of critical information, particularly for weights that fall outside the representable range of 16-bit floating point. In PyTorch's 16-bit format (FP16), precision is sacrificed to gain a broader dynamic range, and values outside the representable range may be clipped or rounded, potentially degrading the network's ability to accurately reconstruct high-fidelity representations at lower bit rates. For Jasper Ridge, *hp_ours* performs better than *ours*. However, both *ours* and *hp_ours* achieve higher PSNR values than other methods. For Pavia University and Cuprite datasets, *our* method obtains better PSNR values than other methods.

We draw the following conclusions from these results: 1) the proposed method, both *ours* and *hp_ours*, perform high-quality compression at high compression rates; 2) it is beneficial to perform architecture search plus examine the effects of quantiza-



(a) Indian Pines

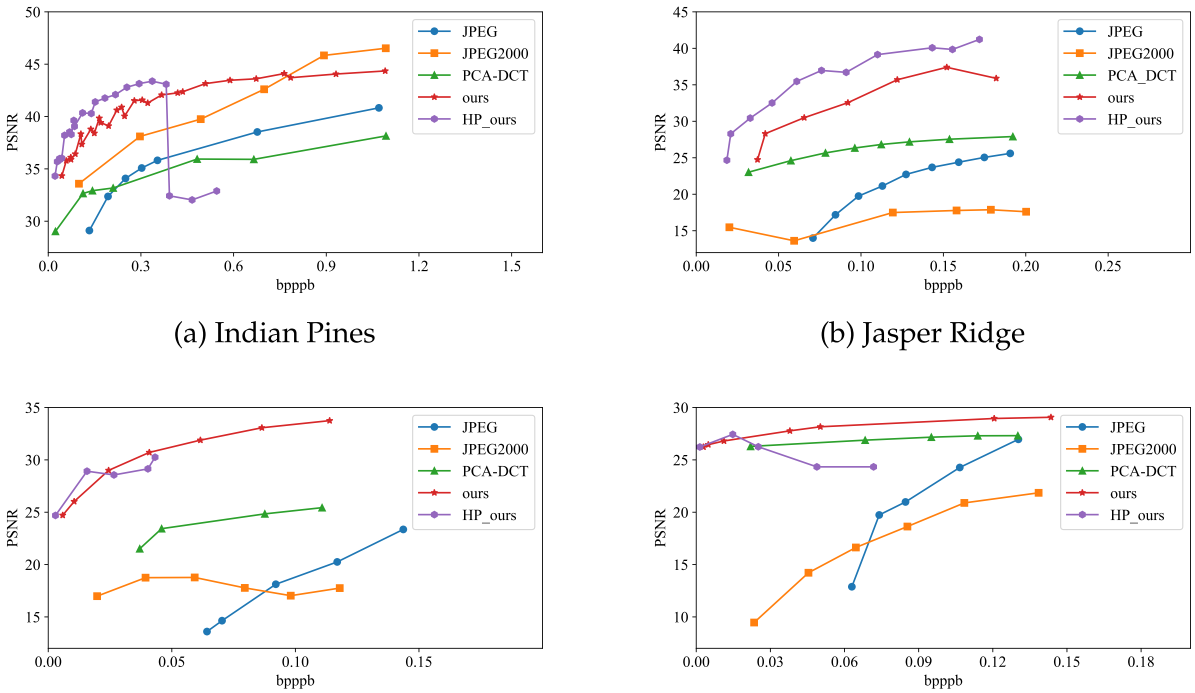
(b) Jasper Ridge

(c) Pavia University

(d) Cuprite

Figure 4.2: **Architecture search**. Exploring MLP structure that achieves the best PSNR for different datasets (for a fixed *bpppb* budget). For our purposes, the MLP structure is defined by the number of hidden layers and the width of these layers. Together, the number and width of the hidden layers define network capacity.

tion at compression time since on some datasets *hp_ours* outperforms *ours*; and 3) the compression quality obtained by the proposed method compares favorably with the three commonly used compression methods for hyperspectral images.



(a) Indian Pines

(b) Jasper Ridge

(c) Pavia University

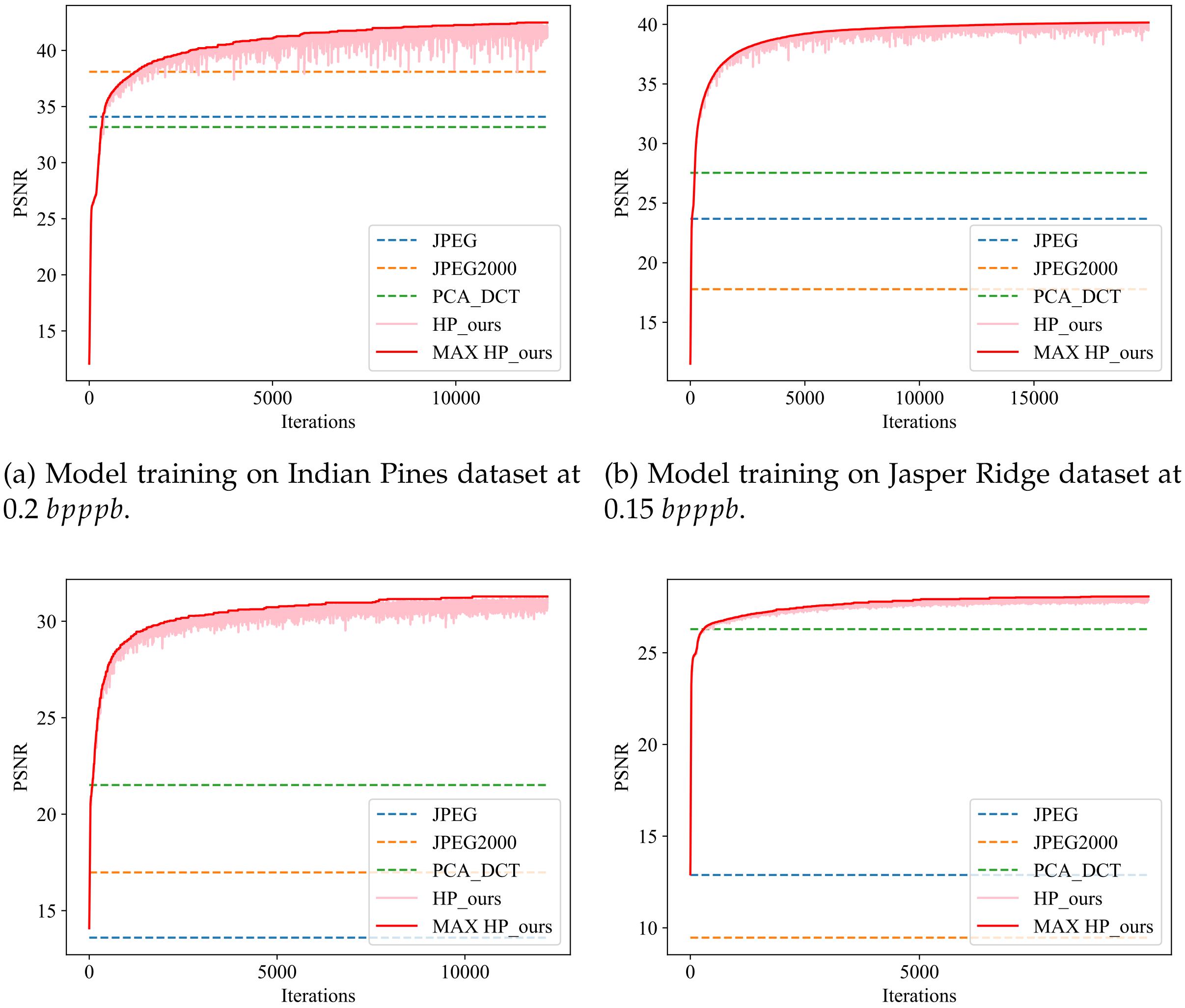
(d) Cuprite

Figure 4.3: **Compression results**. PSNR values achieved at various *bpppb* for our method, along with those obtained by JPEG, JPEG2000, and PCA-DCT schemes. "Ours" refers to our method where parameters are stored at 32-bit precision, and "HP_ours" refers to results when parameters are stored at 16-bit precision.

4.4.3 Encoding Considerations

Our method belongs to the class of "slow-encoding-fast-decoding" compression methods. The method needs to train, actually *overfit*, multiple MLPs at encoding (compression) time. This is needed to find the MLP structure that best represents the hyperspectral image given a particular storage budget. Decoding, however, only requires evaluating this MLP at various pixel locations. Decoding is fast. It can be made even faster by exploiting the parallelism inherent to this procedure. The "slow-encoding-fast-decoding" nature of this method makes it particularly suitable for applications where the hyperspectral image is compressed once only, say at capture time.

We show an example of the overfitting procedure in Figure 4.4. These plots show the encoding procedure on the four datasets: (1) Indian Pines at 0.2 $bpppb$; (2) Jasper Ridge at 0.15 $bpppb$; (3) Pavia University at 0.025 $bpppb$; and (4) Cuprite at 0.02 $bpppb$. JPEG, JPEG2000, and PCA-DCT methods do not require iterations. Consequently, their respective PSNR values are denoted with the horizontal dashed lines. The method proposed in this thesis is iterative. Note that PSNR values for *HP_ours* continue to increase with the number of iterations (up to a point). Improvement in PSNR values saturates at around $10,000$, $15,000$, $10,000$, and $5,000$ iterations for Indian Pines, Jasper Ridge, Pavia University, and Cuprite datasets, respectively. This hints at the upper bound on encoding, or compression, time for our method. Note also that at around $2,000$ iteration mark *HP_ours* method starts to obtain better PSNR values than the other three methods. As stated earlier, our method involves model fitting, which is inherently stochastic. Therefore, throughout the iterative process, we store the model parameters that obtained the highest value for PSNR thus far. In these plots, *MAX HP_ours* denote these PSNR scores. This guarantees that the model does not get worse over time.



(a) Model training on Indian Pines dataset at 0.2 *bpppb*.

(b) Model training on Jasper Ridge dataset at 0.15 *bpppb*.

(c) Model training on Pavia University dataset at 0.025 *bpppb*.

(d) Model training on Cuprite dataset at 0.02 *bpppb*.

Figure 4.4: Encoding procedure. Model training on (counter-clockwise from top-left) Indian Pines, Jasper Ridge, Pavia University, and Cuprite datasets. At around the 2000-iteration mark, our method is already achieving better PSNR values than those for JPEG, JPEG2000, and PCA-DCT. Furthermore, the PSNR value for our methods continues to improve with more iterations (up to a point).

Method	Dataset	Size	PSNR	bpppb	n_h, w_h	Dataset	Size	PSNR	bpppb	n_h, w_h
-	Indian Pines	9251 KB	∞	16	-,-	Jasper Ridge	4800 KB	∞	16	-,-
JPEG		115.6 KB	34.085	0.2	-,-		30 KB	21.130	0.1	-,-
JPEG2000		115.6 KB	38.098	0.2	-,-		30 KB	17.494	0.1	-,-
PCA-DCT		115.6 KB	33.173	0.2	-,-		30 KB	26.821	0.1	-,-
ours		115.6 KB	**40.61**	0.2	15,40		30 KB	**35.696**	0.1	10,20
hp_ours		57.5 KB	40.35	0.1	15,40		15 KB	35.467	0.06	10,20
-	Pavia University	42724 KB	∞	16	-,-	Cuprite	140836 KB	∞	16	-,-
JPEG		267 KB	20.253	0.1	-,-		880.2 KB	24.274	0.1	-,-
JPEG2000		267 KB	17.752	0.1	-,-		880.2 KB	20.889	0.1	-,-
PCA-DCT		267 KB	25.436	0.1	-,-		880.2 KB	27.302	0.1	-,-
ours		267 KB	**33.749**	0.1	20,60		880.2 KB	**28.954**	0.1	25,100
hp_ours		133.5 KB	20.886	0.05	20,60		440.1 KB	24.334	0.06	25,100

Table 4.2: Compression results

4.4.4 Model Fitting

The number of inputs for all our models was 2, and the number of outputs was equal to the number of channels (or bands) of the hyperspectral image. The activation functions for hidden layers were sinusoidal. We initialized the MLP using the guidelines provided in [Sitzmann et al., 2020b]. Adam optimizer was used during training, and the learning rate was set to $2e-4$. All experiments were conducted on an Intel i7 desktop with Nvidia RTX 2080 GPU.

4.4.5 Compression Results

Table 4.2 lists compression results obtained by *ours*, *HP_ours*, JPEG, JPEG2000, and PCA-DCT methods on the four datasets. The table also shows the size of the original, uncompressed hyperspectral images. For these results, we fix the *bpppb* for each method, and we measure the performance of each method using PSNR. Notice that the proposed method achieves higher PSNR values than those achieved by JPEG, JPEG2000, and PCA-DCT methods. Figure 4.5 presents a visual comparison between the original and reconstructed hyperspectral images for four benchmark datasets: Indian Pines, Jasper Ridge, Pavia University, and Cuprite. Each row displays the original image on the left and its corresponding reconstructed version on the right. These

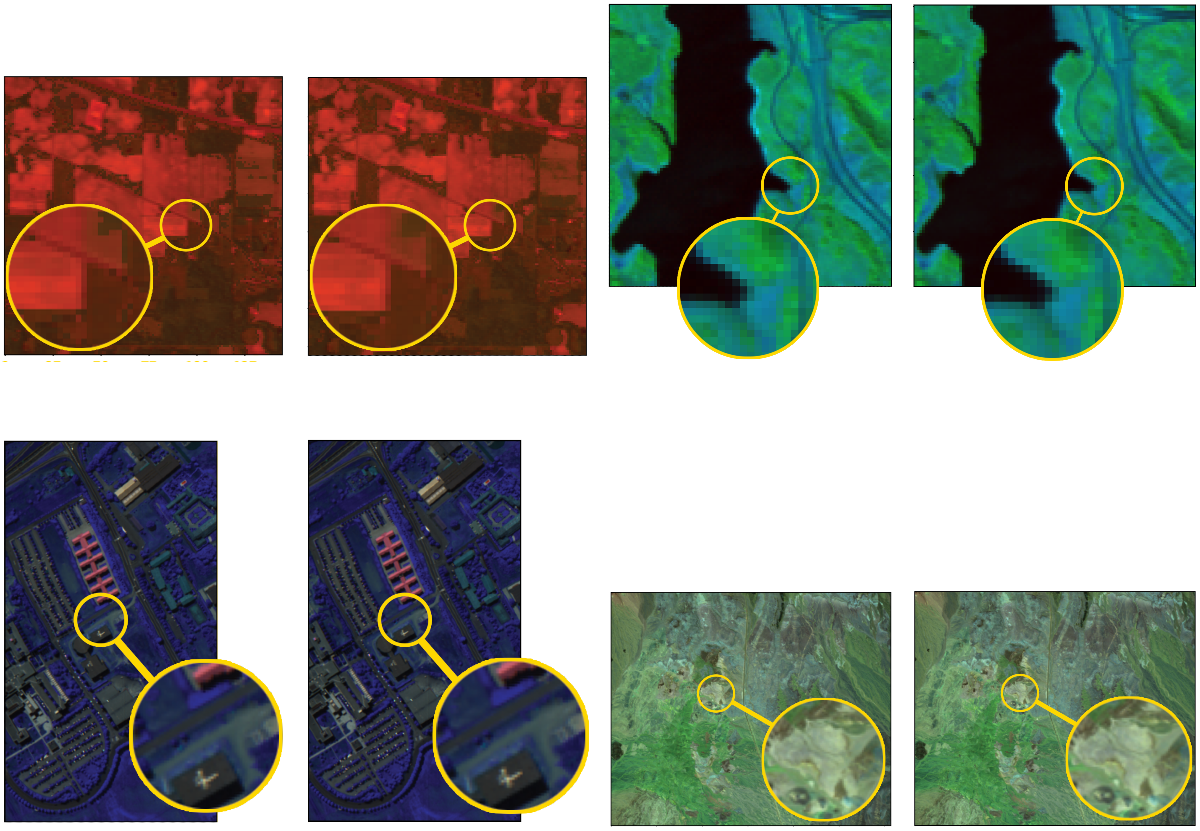


Figure 4.5: **Reconstructed images** shown in pseudo-color. Each pair (left to right) displays the original (left) and reconstructed (right) hyperspectral images for the Indian Pines, Jasper Ridge, Pavia University, and Cuprite datasets. The zoomed-in portions confirm that structural details are well preserved in the reconstructed images, validating the performance of the proposed compression technique.

images are shown in pseudo-color to highlight spectral features and make the visual differences more discernible. The magnified portions demonstrate that the key structural elements and spatial patterns are preserved effectively in the reconstructed images, validating the performance of the proposed compression technique. The close alignment between the original and reconstructed regions emphasizes the method's ability to maintain fidelity, even at high compression rates. This visualization reinforces that the proposed approach achieves a favorable balance between data reduction and image quality, essential for applications requiring both efficient storage and high-quality analysis of hyperspectral data.

4.4.6 Ablative Study For Random Sampling

Hyperspectral image compression is a crucial step in handling the vast amount of data produced by hyperspectral imaging systems. Traditional compression methods, such as transform-based and predictive coding approaches, often face challenges in balancing high compression efficiency with minimal loss of information. With the advent of implicit neural representations (INRs), a new paradigm has emerged, allowing hyperspectral images to be compressed as continuous functions parameterized by neural networks. This approach provides several advantages, including better reconstruction quality and higher adaptability to spatial variations in the image. However, the computational complexity of training INR models on high-resolution hyperspectral images remains a significant bottleneck. To mitigate this, we propose a random sampling strategy, where a subset of pixel locations is used during training. This method reduces compression time while maintaining high reconstruction fidelity, as demonstrated in our experimental results. In the following sections, we provide a detailed analysis of how sampling influences compression performance and discuss its implications for real-world applications.

4.4.6.1 HSI Compression Using Sampling and INR

During compression, the INR model is trained by iterating over $w \times h$ pixel locations for each epoch. For hyperspectral images with high-spatial resolution this leads to longer compression times. Hyperspectral images, similar to color images, exhibit spatial coherence. We use this assumption and explore the effect of pixel location sampling, hereafter referred to as sampling, during INR model training. The idea is to sample a fraction of pixel locations at training time. Figure 4.6 illustrates the sampling procedure. First, the image is divided into non-overlapping tiles (in the spatial domain). This is done in order to ensure that the same fraction of pixel locations are sampled from each region of the image. The colored dots indicate the sampled locations within each tile. The sampling is performed with rejection during each epoch to prevent the case where the same pixel location is sampled multiple times during each epoch. Furthermore, there is not restriction that the same pixel locations are samples across epochs. The sampling procedure is controlled by two parameters: (1) tiles and (2) the fraction of locations to be sampled within each tile. For the results presented in Table 4.3, the image is divided into 9-by-9 tiles and the sampling fraction is set to 20%. This table shows compression and decompression times for four datasets. As expected sampling reduces the compression times in half for all datasets. What is more intriguing, however, is that for comparable $bpppb$ values, sampling results in the highest PSNR values. We do not yet fully understand why this is so. *ours-sampling-32bit* and *ours-sampling-16bit* denote models that used sampling during training. As before prefix *HP* denotes that model parameters are stored using half-precision.

4.5 Results

Tables 4.4 and 4.5 list compression results in terms of PSNR with a number of classical and learning-based hyperspectral image compression approaches. In order to

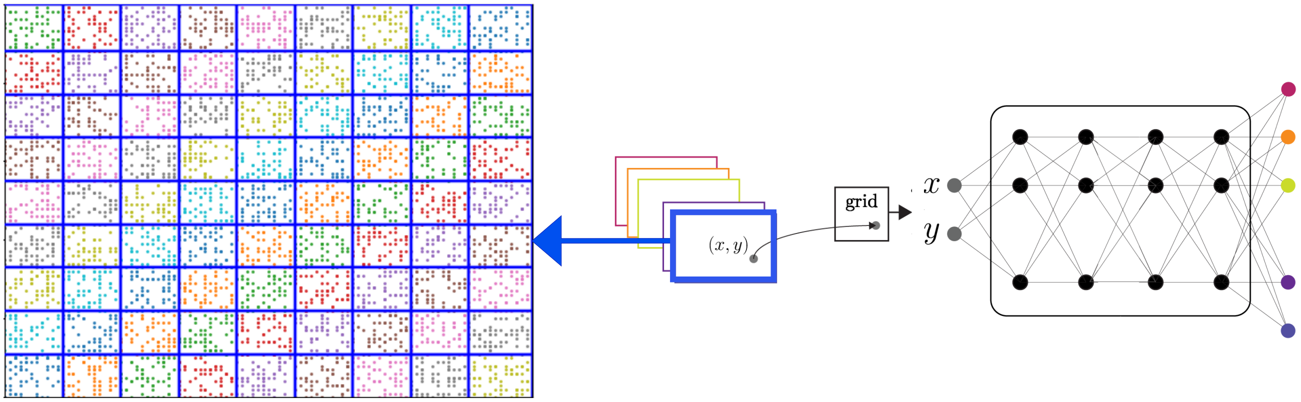


Figure 4.6: Compression with random sampling. Pixel locations are uniformly sampled during model training. This is a departure from the traditional approach of using all pixel locations when fitting INRs.

Dataset	Method	bppppb	compression time (Sec)	decompression time (Sec)	PSNR ↑
Indian Pines	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.1	7.353	3.27	27.47
	JPEG2000$^+$ [Du and Fowler, 2007]	0.1	0.1455	0.3115	33.58
	PCA-DCT$^+$ [Nian et al., 2016]	0.1	1.66	0.04	32.28
	ours-32bit	0.1	243.64	0.0001	36.98
	ours-16bit	0.05	243.64	0.0001	36.95
	ours-sampling-32bit	0.1	282.08	0.0005	**40.1**
	ours-sampling-16bit	0.05	282.08	0.0005	28.40
Jasper Ridge	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.1	3.71	1.62	21.13
	JPEG2000$^+$ [Du and Fowler, 2007]	0.1	0.138	0.395	17.49
	PCA-DCT$^+$ [Nian et al., 2016]	0.1	1.029	0.027	26.82
	ours-32bit	0.1	312.38	0.0005	32.54
	ours-16bit	0.06	312.38	0.0005	32.51
	ours-sampling-32bit	0.1	75.91	0.0005	**34.77**
	ours-sampling-16bit	0.06	75.91	0.0005	22.07
Pavia University	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.1	33.86	14.61	20.25
	JPEG2000$^+$ [Du and Fowler, 2007]	0.1	0.408	0.628	17.75
	PCA-DCT$^+$ [Nian et al., 2016]	0.1	6.525	0.235	25.43
	ours-32bit	0.1	780.16	0.0009	34.46
	ours-16bit	0.05	780.16	0.0009	34.17
	ours-sampling-32bit	0.1	72.512	0.0004	**38.08**
	ours-sampling-16bit	0.05	72.512	0.0004	27.02
Cuprite	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.06	101.195	45.02	12.88
	JPEG2000$^+$ [Du and Fowler, 2007]	0.06	1.193	2.476	15.16
	PCA-DCT$^+$ [Nian et al., 2016]	0.06	11.67	0.754	26.75
	ours-32bit	0.06	1565.97	0.001	28.02
	ours-16bit	0.03	1565.97	0.001	27.90
	ours-sampling-32bit	0.06	664.87	0.001	**37.27**
	ours-sampling-16bit	0.03	664.87	0.001	24.85

Table 4.3: The effect of sampling on compression times. ours-32bit is a learning-based method that requires us to train an MLP on the input image. Consequently, compression times for our method are significantly larger than those of the schemes listed here. We attempt to address this somewhat through sampling. A sampling rate of 20% cuts the compression time by half. The good news is our method, with our without sampling, achieves good decompression times. Indeed the proposed method achieves faster decompression times than JPEG, JPEG2000, and PCA-DCT methods shown in this table. This suggests that the proposed method is well-suited to "compress-once" sort of applications. Text decoration $^+$ indicates a classical method.

compare the performance of different methods, the target $bpppb$ is fixed across methods. For the Indian Pines dataset, the target $bpppb$ is set to 0.2, and for the other three datasets, it is set to 0.1. Our methods that store network parameters in half-precision further reduce the target $bpppb$ by one-half. These methods use the *HP* prefix. Suffix *sampling* denotes that the model was training using sampling as discussed above. For all datasets, the uncompressed $bpppb$ is 16. As can be seen, Cuprite is the most popular dataset—the results are available for this dataset for every method except PCA-JPEG2000 and FPCA-JPEG2000. The last column describes the network structure used for INR learning. n_h and w_h refer to the number of hidden layers and the width of these layers, respectively. These results confirm that the proposed method achieves better PSNR at a comparable $bpppb$ than all other methods. except for the Jasper Ridge and Cuprite dataset, in which the PCA-JPEG2000 method got better PSNR than our sampling method at 0.1 ppb value.

4.5.1 Compression Rates

Figure 4.7 shows PSNR values at various compression rates for different methods. Specifically, we compare our approach, labeled as *ours-32bit*, *ours-16bit*, *ours-sampling-32bit*, and *ours-sampling-16bit*, with JPEG, JPEG2000, PCA-DCT, MPEG, X264, X265, PCA-X264, PCA-X265 PCA-JPEG2000, and autoencoder[La Grassa et al., 2022] methods. Here, *ours-32bit* and *ours-sampling-32bit* methods store MLP weights as 32-bit floating point values, whereas *ours-16bit* and *ours-sampling-16bit* store MLP weights at half-precision as 16-bit floating point values that are constructed by quantizing the MLP weights. These plots illustrate that our methods achieve higher compression quality, i.e., better PSNR, for a given value of $bpppb$. This is especially true for lower $bpppb$ values.

For the Indian Pines dataset, *ours-32bit* method achieves better PSNR up to around 0.7 $bpppb$, at which point X264 obtains better PSNR. What is curious is that the PSNR

Method	Dataset	Size (KB)	PSNR	bpppb	n_h, w_h
-	Indian Pines	9251	∞	16	-,-
JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]		115.6	34.085	0.2	-,-
JPEG2000$^+$ [Du and Fowler, 2007]		115.6	35.84	0.2	-,-
PCA-DCT$^+$ [Nian et al., 2016]		115.6	33.173	0.2	-,-
3D-SPECK$^+$ [Ngadiran et al., 2010]		115.6	-	0.2	-,-
PCA-JPEG2000† [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a]		115.6	39.5	0.2	-,-
FPCA-JPEG2000† [Mei et al., 2018]		115.6	40.5	0.2	-.-
3D-DCT† [Yadav and Nagmode, 2018]		115.6	-	0.2	-,-
3D-DWT-SVR† [Zikiou et al., 2020]		115.6	-	0.2	-,-
WSRC† [Ouahioune et al., 2021]		115.6	-	0.2	-,-
ours-32bit		115.6	39.10	0.2	5,60
ours-16bit		57.5	38.99	0.1	5,60
ours-sampling-32bit		115.6	**42.22**	0.2	5,60
ours-sampling-16bit		57.5	29.68	0.1	5,60
ours-sampling-8bit		28.7	32.97	0.05	5,60
-	Jasper Ridge	4800	∞	16	-,-
JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]		30	21.130	0.1	-,-
JPEG2000$^+$ [Du and Fowler, 2007]		30	17.494	0.1	-,-
PCA-DCT$^+$ [Nian et al., 2016]		30	26.821	0.1	-,-
3D-SPECK$^+$ [Ngadiran et al., 2010]		30	-	0.1	-,-
PCA-JPEG2000† [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a]		30	**44.26**	0.1	-,-
FPCA-JPEG2000† [Mei et al., 2018]		30	-	0.1	-,-
3D-DCT† [Yadav and Nagmode, 2018]		30	-	0.1	-,-
3D-DWT-SVR† [Zikiou et al., 2020]		30	-	0.1	-,-
WSRC† [Ouahioune et al., 2021]		30	-	0.1	-,-
ours-32bit		30	32.54	0.1	5,20
ours-16bit		15	32.51	0.06	5,20
ours-sampling-32bit		30	34.77	0.1	5,20
ours-sampling-16bit		15	22.07	0.06	5,20
ours-sampling-8bit		7.5	24.32	0.03	5,20

Table 4.4: Compression results on the two datasets. Compression rates (i.e., the desired size of the compressed file) is fixed across methods. The quality of compression is expressed in terms of PSNR. In case of the proposed method, we include five variants: the method without sampling, both 32-bit (ours-32bit) and 16-bit (ours-16bit) versions, and the method with a sampling rate of 20%, 32-bit (ours-sampling-32bit), 16-bit (ours-sampling-16bit), and 8-bit (ours-sampling-8bit) versions. In case of our methods, the last column includes the number of layers and the width of these layers. The highest PSNR values for each dataset are shown in bold. Text decorations $^+$ and † indicate classical and learning-based methods, respectively

Method	Dataset	Size (KB)	PSNR	bpppb	n_h, w_h
-	Pavia University	42724	∞	16	-,-
JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]		267	20.253	0.1	-,-
JPEG2000$^+$ [Du and Fowler, 2007]		267	17.752	0.1	-,-
PCA-DCT$^+$ [Nian et al., 2016]		267	25.436	0.1	-,-
3D-SPECK$^+$ [Ngadiran et al., 2010]		267	-	0.1	-,-
PCA-JPEG2000† [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a]		267	33	0.1	-,-
FPCA-JPEG2000† [Mei et al., 2018]		267	-	0.1	-,-
3D-DCT† [Yadav and Nagmode, 2018]		267	-	0.1	-,-
3D-DWT-SVR† [Zikiou et al., 2020]		267	-	0.1	-,-
WSRC† [Ouahioune et al., 2021]		267	-	0.1	-,-
ours-32bt		267	34.46	0.1	10,80
ours-16bit		133.5	34.17	0.05	10,80
ours-sampling-32bit		267	**38.08**	0.1	10,80
ours-sampling-16bit		133.5	27.02	0.05	10,80
ours-sampling-8bit		66.75	24.02	0.02	10,80
-	Cuprite	140836	∞	16	-,-
JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]		880.2	24.274	0.1	-,-
JPEG2000$^+$ [Du and Fowler, 2007]		880.2	20.889	0.1	-,-
PCA-DCT$^+$ [Nian et al., 2016]		880.2	27.302	0.1	-,-
3D-SPECK$^+$ [Ngadiran et al., 2010]		880.2	27.1	0.1	-,-
PCA-JPEG2000† [Kwan and Larkin, 2019, Kwan et al., 2019b, Kwan et al., 2019a]		880.2	**40.90**	0.1	-,-
FPCA-JPEG2000† [Mei et al., 2018]		880.2	-	0.1	-,-
3D-DCT† [Yadav and Nagmode, 2018]		880.2	33.4	0.1	-,-
3D-DWT-SVR† [Zikiou et al., 2020]		880.2	28.20	0.1	-,-
WSRC† [Ouahioune et al., 2021]		880.2	35	0.1	-,-
ours-32bt		880.2	28.954	0.1	25,100
ours-16bit		440.1	24.334	0.06	25,100
ours-sampling-32bit		880.2	36.55	0.1	25,90
ours-sampling-16bit		440.1	24.91	0.06	25,90
ours-sampling-8bit		220.05	22.35	0.03	25,90

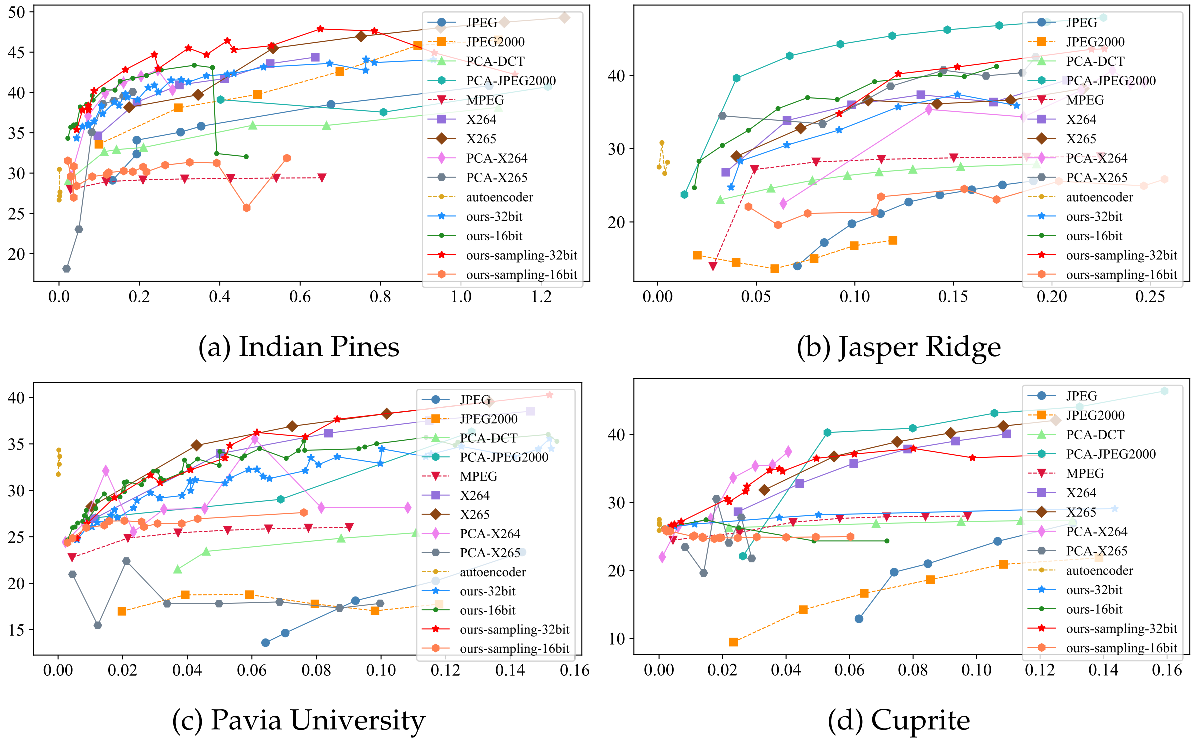
Table 4.5: Compression results on the two datasets. Compression rates (i.e., the desired size of the compressed file) is fixed across methods. The quality of compression is expressed in terms of PSNR. In case of the proposed method, we include five variants: the method without sampling, both 32-bit (ours-32bit) and 16-bit (ours-16bit) versions, and the method with a sampling rate of 20%, 32-bit (ours-sampling-32bit), 16-bit (ours-sampling-16bit), and 8-bit (ours-sampling-8bit) versions. In case of our methods, the last column includes the number of layers and the width of these layers. The highest PSNR values for each dataset are shown in bold. Text decorations $^+$ and † indicate classical and learning-based methods, respectively

for *ours-16bit* drops drastically at around 0.4 *bppb*. This merits further investigation. One possible explanation is the effect of weight quantization. In our implementation, we quantize the network weights from 32-bit to 16-bit floating-point precision to reduce storage. However, this quantization may result in a loss of critical information, particularly for weights that fall outside the representable range of 16-bit floating point. In PyTorch's 16-bit format (FP16), precision is sacrificed to gain a broader dynamic range, and values outside the representable range may be clipped or rounded, potentially degrading the network's ability to accurately reconstruct high-fidelity representations at lower bit rates. *ours-sampling-32bit* method got better PSNR than all other methods at various bpppb levels. For Jasper Ridge, *ours-sampling-32bit* and *ours-16bit* perform better than *ours-32bit*. The *ours-16bit* and *ours-sampling-32bit* methods achieve higher PSNR values than other methods except PCA-JPEG2000.

For the Pavia University dataset, *our* and *ours-16bit* methods got PSNR at lower bit rates than other methods. *ours-sampling-32bit* method got better PSNR than all other methods, around 0.1 bpppb and more. For the Cuprite datasets, *our* method got PSNR than other methods, except PCA-JPG2000, X264, and X265, specifically at lower bit rates. *ours-sampling-32bit* method also got better PSNR than other methods at various bpppb levels, except PCA-X264 and PCA-JPEG2000, specifically at lower bit rates.

4.5.2 The Effect of Sampling

As mentioned before, we use the proposed sampling method with INR to improve the PSNR for the datasets. Table 4.6 compares our results using sampling (ours-sampling-32bit method) with video coding-based hyperspectral image compression methods like X264, X265, PCA-X264, PCA-X265, MPEG, HEVC, and RPM. We fix the bpppb for each method and dataset, and we measure the performance of each method using PSNR. Our ours-sampling-32bit method got better PSNR than all other video coding-



(a) Indian Pines

(b) Jasper Ridge

(c) Pavia University

(d) Cuprite

Figure 4.7: **Compression results**. PSNR values achieved at various *bpppb* for our method, along with those obtained by other methods. Here "ours-32bit" and "ours-sampling-32bit" refer to our method where parameters are stored at 32-bit precision, and "ours-16bit" and "ours-sampling-16bit" refer to results when parameters are stored at 16-bit precision.

Dataset	**Method**	**bpppb**	**PSNR ↑**
Indian Pines	X264‡ [Kwan and Larkin, 2019]	0.1	34.61
	X265‡ [Kwan and Larkin, 2019]	0.1	38.5
	PCA-X264‡ [Kwan and Larkin, 2019]	0.1	39.8
	PCA-X265‡ [Kwan and Larkin, 2019]	0.1	38.1
	MPEG‡ [Le Gall, 1991]	0.1	28.9
	HEVC‡ [Sullivan et al., 2012]	0.1	30
	RPM‡ [Paul et al., 2016]	0.1	31
	ours-sampling-32bit	0.1	**40.1**
Jasper Ridge	X264‡ [Kwan and Larkin, 2019]	0.15	37.35
	X265‡ [Kwan and Larkin, 2019]	0.15	36.12
	PCA-X264‡ [Kwan and Larkin, 2019]	0.15	35.35
	PCA-X265‡ [Kwan and Larkin, 2019]	0.15	39.94
	MPEG‡ [Le Gall, 1991]	0.15	28.75
	HEVC‡ [Sullivan et al., 2012]	0.15	-
	RPM‡ [Paul et al., 2016]	0.15	-
	ours-sampling-32bit	0.15	**41.15**
Pavia University	X264‡ [Kwan and Larkin, 2019]	0.1	37.17
	X265‡ [Kwan and Larkin, 2019]	0.1	37.90
	PCA-X264‡ [Kwan and Larkin, 2019]	0.1	28.13
	PCA-X265‡ [Kwan and Larkin, 2019]	0.1	17.82
	MPEG‡ [Le Gall, 1991]	0.1	26.01
	HEVC‡ [Sullivan et al., 2012]	0.1	-
	RPM‡ [Paul et al., 2016]	0.1	-
	ours-sampling-32bit	0.1	**38.08**
Cuprite	X264‡ [Kwan and Larkin, 2019]	0.03	28.6
	X265‡ [Kwan and Larkin, 2019]	0.03	31.8
	PCA-X264‡ [Kwan and Larkin, 2019]	0.03	**35.5**
	PCA-X265‡ [Kwan and Larkin, 2019]	0.03	21.7
	MPEG‡ [Le Gall, 1991]	0.03	25.5
	HEVC‡ [Sullivan et al., 2012]	0.03	25
	RPM‡ [Paul et al., 2016]	0.03	29
	ours-sampling-32bit	0.03	34.9

Table 4.6: Comparing the proposed method (with sampling rate of 20%) against video-based schemes. The proposed method achieves higher PSNR for all datasets except Cuprite. For each dataset, the compression rate is fixed across methods: 0.1 for the Indian Pines, 0.15 for the Jasper Ridge datasets, 0.1 for the Pavia University dataset, and 0.03 for the Cuprite dataset. Text decoration ‡ indicates a video-based method.

based compression methods for all four datasets, except Cuprite.

Dataset	bpppb	Method	SSIM ↑
Cuprite	0.1	WSRC [Ouahioune et al., 2021]	0.75
		ours-sampling-32bit	**0.9798**
	0.01	3D-SPECK [Ngadiran et al., 2010]	0.142
		3D-SPIHT [Fowler and Rucker, 2007]	0.136
		3D-WBTC [Bajpai et al., 2019a]	0.141
		3D-LSK [Ngadiran et al., 2010]	0.138
		3D-NLS [Sudha and Sudhakar, 2013]	0.135
		3D-LMBTC [Bajpai et al., 2019b]	0.140
		3D-ZM-SPECK [Bajpai et al., 2022]	0.141
		ours-32bit	**0.9565**
		ours-16bit	0.9514
		ours-sampling-32bit	0.9527
		ours-sampling-16bit	0.9390
Pavia University	0.1	3D-SPHIT [Fowler and Rucker, 2007]	0.4
		3D-DCT [Yadav and Nagmode, 2018]	0.85
		ours-32bit	0.9564
		ours-16bit	0.9527
		ours-sampling-32bit	**0.9901**
		ours-sampling-16bit	0.8518

Table 4.7: SSIM comparison for the Cuprite and Pavia University datasets.

4.5.3 SSIM comparison

Table 4.7 uses SSIM metric to compare our method on Cuprite datasest against other schemes for fixed *bpppb*. FOR WSRC [Ouahioune et al., 2021] method, the *bpppb* is fixed to 0.1, whereas for 3D-SPECK, 3D-SPHIT [Fowler and Rucker, 2007], 3D-WBTC [Bajpai et al., 2019a], 3D-LSK [Ngadiran et al., 2010], 3D-NLS [Sudha and Sudhakar, 2013], 3D-LMBTC [Bajpai et al., 2019b], and 3D-ZM-SPECK [Bajpai et al., 2022] the *bpppb* is fixed to 0.01. Furthermore, we compare our method against 3D-SPHIT and 3D-DCT methods on Pavia University dataset for 0.1 *bpppb*. In all cases, the proposed method achieves better SSIM scores (for the selected *bpppb*). The SSIM scores for other methods for both Cuprite and Pavia University datasets are lifted from their respective articles.

4.5.4 Summary

We draw the following conclusions from these results presented thus far: 1) the proposed method, both *ours-32bit* and *ours-16bit*, perform high-quality compression at high compression rates; 2) it is beneficial to perform architecture search plus examine the effects of quantization at compression time since on some datasets *ours-16bit* outperforms *ours-32bit*; 3) the compression quality obtained by the proposed method compares favorably with the three commonly used compression methods for hyperspectral images; and 4) sampling improves compression times while maintaining acceptable reconstruction quality as measured by PSNR.

4.6 Conclusion

In this work, we employ implicit neural representations to compress hyperspectral images. Multi-layer perceptron neural networks with sinusoidal activation layers are overfitted to a hyperspectral image. The network is trained to map pixel locations to pixels' spectral signatures. The parameters of the network, along with its structure, represent a compressed encoding of the original hyperspectral image. We also use a sampling method with two factors: window size and sampling rate to reduce the compression time. We have tested our approach on four datasets, and the proposed method achieves better PSNR than those achieved by JPEG, JPEG2000, PCA-DCT, MPEG, X264, X265, PCA-X264, PCA-X265, and PCA-JPEG2000 methods, especially at lower bitrates. Besides, we compare our results with the learning-based methods like PCA-JPEG2000, FPCA-JPEG2000, 3D-DCT, 3D-DWT-SVR, and WSRC and show that we got better PSNR and SSIM than those methods. We also show that our methods with sampling achieve better speed and performance than our methods without sampling.

We plan to experiment with meta-networks to achieve smaller sizes for the compressed encodings and lower the compression times.

Chapter 5

HSI Compression using INR and Meta-Learned Based Network

Hyperspectral image compression is a challenging task due to the high-dimensional nature of hyperspectral data. The previous work in this thesis on hyperspectral image compression using implicit neural representations (INR) and sampling demonstrated that neural networks could effectively learn representations of spectral data. However, a limitation of this approach is the long network training time required for each image. This inefficiency arises because the method does not exploit the spatial and spectral structural similarities shared across images, necessitating independent training for each hyperspectral image. Consequently, compression times remain high. To address this issue, we propose a meta-learned base network that leverages shared representations across multiple hyperspectral images, enabling faster encoding while maintaining high compression quality. Our method introduces modulated neural networks that adapt image-specific details efficiently, reducing training time and improving compression performance.

5.1 A Meta-Learned Approach to Hyperspectral Image Compression

Consider a hyperspectral image $\mathbf{I} \in \mathbb{R}^{W \times H \times C}$, where W and H denote the width and the height of this image and C denotes the number of channels. $\mathbf{I}(x, y) \in \mathbb{R}^{C}$ represents the spectrum recorded at location (x, y) where $x \in [1, W]$ and $y \in [1, H]$. In Chapter 4, we demonstrate that it is possible to learn implicit neural represenations that map pixel locations to pixel spectra. Specifically, we can learn a function $\Phi_{\Theta} : (x, y) \mapsto \mathbf{I}(x, y)$. Here, Θ represent function parameters. The implicit neural network is trained by minimizing the loss

$$\mathcal{L}(\mathbf{I}, \Phi_{\Theta}) = \sum_{\forall x,y} \|\mathbf{I}(x, y) - \Phi_{\Theta}(x, y)\|.$$

Others [Tancik et al., 2020, Sitzmann et al., 2020b] have shown that SIREN networks—multi-layer perceptrons with Sine activation functions—are particularly well-suited to encode high-frequency data that sits on a grid. SIREN networks are widely used to learn implicit neural representations. For our purposes, a SIREN network (Φ_{Θ}) comprises of K hidden layers. Each layer uses a sinosoidal activation function. The K hidden features at each layer are $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \cdots, \mathbf{h}_K$. Specifically, we define the SIREN network as:

$$\mathbf{h}_i = sin\left(\mathbf{W}_i \mathbf{h}_{i-1} + \mathbf{b}_i\right),$$

where $\mathbf{h}_0 \in \mathbb{R}^2$ denotes the 2D pixel locations, $\mathbf{W}_1 \in \mathbb{R}^{d \times 2}$, $\mathbf{b}_1 \in \mathbb{R}^{d}$, and for $i \in [2, K]$, $\mathbf{W}_i \in \mathbb{R}^{d \times d}$ and $_i, \mathbf{b}_i \in \mathbb{R}^{d}$. For notational convenience, we refer to d as the width of the hidden layer. The output of the network is

$$\mathbf{h}_{K+1} = \mathbf{W}_{K+1} \mathbf{h}_K + \mathbf{b}_{K+1},$$

with $\mathbf{W}_{K+1} \in \mathbb{R}^{C \times d}$ and $\mathbf{h}_{K+1}, \mathbf{b}_{K+1} \in \mathbb{R}^{C}$. $\mathbf{h}_{K+1}$ is the output of the model, in our case pixel spectrum. $\mathbf{W}_i$ and $\mathbf{b}_i$ denote the weights and biases for layer $i \in [1, K+1]$ and represent the learnable parameters of the network. Once this network is trained on a given hyperspectral image, it is sufficient to store the parameters $\Theta = \{\mathbf{W}_i, \mathbf{b}_i | i \in [1, K+1]\}$, since it is possible to recover the original image by evaluating Φ_Θ at pixel locations (x, y). Savings are achieved when it takes fewer bits to encode Φ_Θ than those required to encode the original image.

While we have successfully employed SIREN networks to compress hyperspectral images, the current scheme suffers from two drawbacks: 1) slow compression times and 2) its inability to exploit spatial and spectral structure that is shared between hyperspectral images, not unlike how spatial structure is used when analyzing RGB images. Both (1) and (2) are due to the a fact that a new SIREN network needs to be trained for scratch for each hyperspectral image. Training is time consuming process that often requires multiple epochs, and no information is shared between multiple images.

5.2 Modulated SIREN Network

In this work we address the two shortcomings by using a meta learning approach that employs a SIREN network (henceforth referred to as the base network) that is shared between multiple hyperspectral images. Image specific details are stored within modulations—scales and shifts—applied to the features $\mathbf{h}_i$, $i \in [0, N]$ of the base network. This is inspired by the work of Perez *et al.*, which introduced FiLM layers [Perez et al., 2018]

$$FiLM(\mathbf{h}_i) = \gamma_i \odot \mathbf{h}_i + \beta_i$$

that apply scale γ_i and shift β_i to a hidden feature $\mathbf{h}_i$. Here $\odot$ denotes element-wise product. Applying shift and scale at each layer in effect allow us to parameterize family of neural networks using a common (fixed) base network. Chan *et al*. propose a scheme where a SIREN network is used to parameterize the generator in a generative-adversarial setting [Chan et al., 2021]. There new samples are generated by applying modulations (scale γ and shift β) as follows:

$$\mathbf{h}_i = sin\left(\gamma_i\left(\mathbf{W}_i\mathbf{h}_{i-1} + \mathbf{b}_i\right) + \beta_i\right).$$

Similarly, Mehta *et al*. show that it is possible to parameterize a family of implicit neural representation by applying modulations to the hidden features as (scale α_i) [Mehta et al., 2021]

$$\mathbf{h}_i = \alpha_i \odot sin\left(\mathbf{W}_i\mathbf{h}_{i-1} + \mathbf{b}_i\right).$$

Both of these approach show that it is possible to map a low-dimensional latent vector to the modulations that are applied to the hidden features. E.g., [Chan et al., 2021] uses an MLP to map a latent vector to scale γ_i and shift β_i. Mehta *et al*., on the other hand, construct the modulation α_i recursively using a fixed latent vector. These schemes, however, require that the parameters of the base network, plus the parameters of the networks needed to compute the modulations are stored. As a consequence these schemes are not well-suited to the problem of data compression.

Work by Dupont *et al*. using modulations to improve SIREN networks [Dupont et al., 2022]. They concluded that it is sufficient to just use shifts β_is, and that using scale modulations do not result in a significant improvement in performance. Furthermore, their work also suggests that applying scale modulations alone does not result in an improvement. We follow their advice and apply shift modulations to the features of the SIREN networks:

$$\mathbf{h}_i = sin\left(\left(\mathbf{W}_i\mathbf{h}_{i-1} + \mathbf{b}_i\right) + \beta_i\right), \tag{5.1}$$

here $\beta_i \in \mathbb{R}^d$. It is easy to imagine that storing modulations $\beta_0, \cdots, \beta_K$ takes less space than storing weights $\mathbf{W}_i$ and biases $\mathbf{b}_i$ of the base network (under the assumption that the cost of storing the base network parameters is amortized over multiple images). It is possible to achieve further savings by mapping a low-dimensional latent vector $\phi \in \mathbb{R}^{d_{\text{latent}}}$ to modulations. Dupont *et al*. also showed that it is sufficient to use a linear mapping to construct modulations β_i given a latent vector and that using a multi-layer perceptron network offers little benefit. Therefore, we use a linear mapping to construct modulations given a latent vector as:

$$\beta = \mathbf{W}_M \phi + \mathbf{b}_M, \tag{5.2}$$

with $\mathbf{W}_M \in \mathbb{R}^{(d)(K) \times d_{\text{latent}}}$ and $\mathbf{b}_M \in \mathbb{R}^{(d)(K)}$, the weights and biases of the linear layer used to project latent vector to modulations $\beta = [\beta_0 | \cdots | \beta_K]$. We refer to the linear layer that maps the latent vector to the modulation as the meta network. Under this setup, it is possible to reconstruct the original hyperspectral image $\mathbf{I}$ by evaluating the modulated base network $\Phi_\Theta(x, y; \beta_0, \cdots, \beta_K)$ at image pixel locations (x, y). Similarly, when using the latent code, we can achieve the same result by evaluating $\Phi_\Theta(x, y; \phi, \Theta_M)$, where $\Theta_M = \{\mathbf{W}_M, \mathbf{b}_M\}$, at image pixel locations (see Figure 5.1).

5.3 Meta learning

Model Agnostic Meta-Learning (MAML) learns an *initialization* of model parameters Θ, such that the model can be quickly adapted to a new (related) task [Finn et al., 2017]. It has been shown that MAML approaches can benefit implicit neural representations by reducing the number of epochs needed to fit the representation to a new data point [Sitzmann et al., 2020a]. We begin by discussing MAML within our context. Say we are given a set of hyperspectral images $\mathbf{I}^{(1)}, \cdots, \mathbf{I}^{(T)}$. Furthermore, assume we want to initialize the parameters Θ of the model Φ_Θ over this set of images. MAML

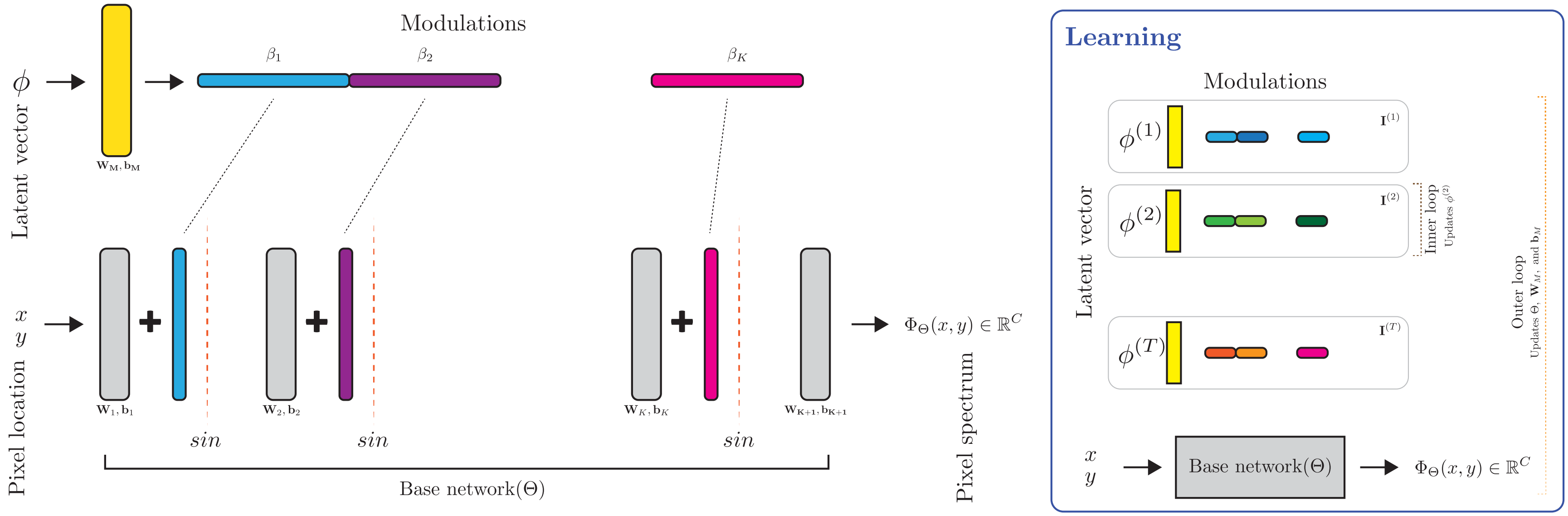


Figure 5.1: The base network captures the shared structure between multiple hyperspectral images, whereas the modulations (or latent vectors) store image-specific information. Meta-learning is used to learn both the shared parameters (Θ, $\mathbf{W}_M$, and $\mathbf{b}_M$) and the image-specific latent vectors ϕ. Once an image is compressed, it is sufficient to store the latent vector associated with this image.

comprises of two loops: (1) in the inner loop, MAML computes an image-specific update

$$\Theta^{(t)} = \Theta - \alpha_{\text{inner}} \nabla_{\Theta} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{\Theta}\right);$$

and (2) in the outer loo,p it updates Θ with respect to the performance of the model (after the inner loop update) on the entire set:

$$\Theta = \Theta - \alpha_{\text{outer}} \nabla_{\Theta} \sum_{t \in [1,T]} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{\Theta^{(t)}}\right).$$

In practice, image t is randomly chosen in the inner loop step, and it is often sufficient to sample a subset of images in the outer loop step. The result is model initialization parameters Θ that will allow the model to be quickly adapted to a previously unseen hyperspectral image, reducing encoding in time.

The approach discussed above is not directly applicable in our setting since we seek to learn image specific modulations that are applied to a base network that is shared between multiple hyperspectral images. We follow the strategy discussed in [Zintgraf et al., 2019] where they partition the parameters into two sets. The first set, termed context parameters, are “task” specific, and these are adapted in the inner loop, whereas the second set is shared across “tasks” and is meta-learned in the outer loop.

We apply this approach to our problem as follows. Given a set of hyperspectral images, parameters Θ of the base networks and image specific modulations $\beta^t = \{\beta_0^{(t)}, \cdots, \beta_K^{(t)}\}$, we first update image-specific modulations in the inner loop as

$$\beta^{(t)} = \beta - \alpha_{\text{inner}} \nabla_{\beta} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{[\Theta|\beta]}\right);$$

and then update the parameters Θ in the outer loop

$$\Theta = \Theta - \alpha_{\text{outer}} \sum_{t \in [1,T]} \nabla_{\Theta} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{[\Theta|\beta^{(t)}]}\right).$$

Starting value for β is fixed and [Zintgraf et al., 2019] suggests to set the initial values for $\beta = \mathbf{0}$. $\Phi_{[\Theta|\beta]}$ denotes the modulated SIREN network (see Equation 5.1).

To achieve further savings, we employ linear mapping defined in Equation 5.2 to construct modulations from a given latent vector ϕ. As before, we can initialize $\phi = \mathbf{0}$. Here, the goal is to learn image-specific latent vectors $\phi^{(t)}$. The procedure is similar, first image specific latent vectors are updated in the inner loop as

$$\phi^{(t)} = \phi - \alpha_{\text{inner}} \nabla_{\phi} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{[\Theta^{+}|\phi]}\right).$$

Next, parameters Θ^{+} are updated in the outer loop

$$\Theta^{+} = \Theta^{+} - \alpha_{\text{outer}} \sum_{t \in [1,T]} \nabla_{\Theta^{+}} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{[\Theta^{+}|\beta^{(t)}]}\right).$$

Here $\Theta^{+} = \{\Theta, \mathbf{W}_M, \mathbf{b}_M\}$ denotes parameters of the base network plus the parameters of the linear mapping used to construct modulations from latent vectors. Parameters Θ^{+} are shared between images and latent vectors ϕ encode information specific to corresponding image.

5.4 Algorithm for Meta-Learned Hyperspectral Image Compression

This section provides the detailed algorithm used for the proposed meta-learned hyperspectral image compression method. The approach involves compressing hyperspectral images using a base network modulated with latent parameters. Meta-learning further optimizes the initialization of the network parameters to minimize the number of gradient steps required during training and testing.

The primary challenge is efficiently fitting hyperspectral images with a shared base network while maintaining individual image information through modulations. The pseudocode below outlines the procedure, covering both the inner and outer loops of optimization, as well as compression and decompression processes.

5.4.1 Pseudocode of the Proposed Method

Algorithm 2 Meta-Learned Hyperspectral Image Compression using Modulated SIREN Network

```
Input: Dataset of hyperspectral images $\mathcal{I} = \{\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \ldots, \mathbf{I}^{(T)}\}$, with normalized pixel coordinates $(x, y) \in [-1, 1] \times [-1, 1]$ and spectra $\mathbf{I}(x, y)$
Output: Compressed latent vectors $\phi^{(t)}$ for each hyperspectral image $\mathbf{I}^{(t)}$ and shared base network parameters $\Theta^{+}$
Initialization:
Initialize base network parameters $\Theta = \{\mathbf{W}_i, \mathbf{b}_i \mid i \in [1, K+1]\}$
Initialize meta-network parameters $\Theta_M = \{\mathbf{W}_M, \mathbf{b}_M\}$
Initialize latent vectors $\phi^{(t)}$ for each hyperspectral image $\mathbf{I}^{(t)}$
Set learning rates $\alpha_{\text{inner}}$ and $\alpha_{\text{outer}}$
for epoch in range(max_epochs) do
    for each hyperspectral image $\mathbf{I}^{(t)} \in \mathcal{I}$ do
        Inner Loop: Update latent vector $\phi^{(t)}$
        $\phi^{(t)} \leftarrow \phi^{(t)} - \alpha_{\text{inner}} \nabla_{\phi^{(t)}} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{[\Theta^{+} | \phi^{(t)}]}\right)$
    Outer Loop: Update shared parameters $\Theta^{+} = \{\Theta, \Theta_M\}$
    $\Theta^{+} \leftarrow \Theta^{+} - \alpha_{\text{outer}} \sum_{t=1}^{T} \nabla_{\Theta^{+}} \mathcal{L}\left(\mathbf{I}^{(t)}, \Phi_{[\Theta^{+} | \phi^{(t)}]}\right)$
Compression:
Store optimized base network parameters $\Theta^{+}$ (shared)
Store latent vectors $\phi^{(t)}$ for each hyperspectral image $\mathbf{I}^{(t)}$
Decompression (Inference Phase):
for each pixel coordinate $(x, y) \in [-1, 1] \times [-1, 1]$ do
    Compute pixel spectrum using modulated base network:
    $\mathbf{I}(x, y) \leftarrow \Phi_{[\Theta^{+} | \phi^{(t)}]}(x, y)$
Reconstruct image: Reconstructed_Image$^{(t)}$ $\leftarrow \{\mathbf{I}(x, y) \mid (x, y) \in [-1, 1] \times [-1, 1]\}$
```

The workflow of the proposed method is outlined in Algorithm 2.

- **Initialization:**
  - Initialize the base network parameters $\Theta = \{\mathbf{W}_i, \mathbf{b}_i \mid i \in [1, K+1]\}$.
  - Initialize the meta-network parameters $\Theta_M = \{\mathbf{W}_M, \mathbf{b}_M\}$.
  - Initialize the latent vectors $\phi^{(t)}$ for each hyperspectral image $\mathbf{I}^{(t)}$.
  - Set the learning rates α_{inner} and α_{outer} for the inner and outer loop updates, respectively.
- **Meta-Learning:**
  - **Inner Loop:**
    * Update the latent vectors $\phi^{(t)}$ for each hyperspectral image $\mathbf{I}^{(t)}$ using gradient descent.
    * Minimize the reconstruction loss $\mathcal{L}(\mathbf{I}^{(t)}, \Phi_{[\Theta^+|\phi^{(t)}]})$ to ensure that the modulations β_i generated from $\phi^{(t)}$ capture image-specific information.
  - **Outer Loop:**
    * Update the shared parameters $\Theta^+ = \{\Theta, \Theta_M\}$ using all hyperspectral images in the dataset.
    * Minimize the overall loss across all images to ensure that the base network generalizes and captures common features.
- **Compression:**
  - After training is complete, store the shared parameters Θ^+ of the base network.
  - Store the latent vectors $\phi^{(t)}$ for each hyperspectral image $\mathbf{I}^{(t)}$ as the compressed representation.
- **Decompression:**

  - During inference, use the stored parameters Θ^+ and the latent vector $\phi^{(t)}$ to reconstruct each hyperspectral image.
  - For each pixel coordinate (x, y) in the image grid:
    * Evaluate the modulated base network $\Phi_{[\Theta^+|\phi^{(t)}]}$ to compute the pixel spectrum $\mathbf{I}(x, y)$.
  - Assemble the reconstructed pixel spectra to form the complete image $\mathbf{I}^{(t)}$.

5.4.2 Discussion

The proposed meta-learned hyperspectral image compression method offers several notable advantages and addresses key challenges in hyperspectral data processing. As outlined in Algorithm 2, this approach is designed to overcome the limitations of traditional compression methods by leveraging the strengths of both neural representations and meta-learning. Below, we discuss its core strengths, trade-offs, and future potential.

- Advantages of Meta-Learned Compression
  - Efficient Generalization Across Datasets: A key advantage of the proposed method is that it uses a shared base network to generalize across multiple datasets. This significantly reduces redundancy, as the shared parameters Θ^+ capture common features present in hyperspectral data (e.g., natural scenes). By storing only the latent vectors $\phi^{(t)}$ for individual datasets, the storage requirements are reduced compared to storing separate models for each dataset.
  - Fast Decompression and Compact Storage: The method excels in applications where fast decompression is essential, such as real-time analysis of large hyperspectral datasets. Once the modulations and shared parameters

are stored, the decompression phase requires only a few forward passes through the base network. This makes the algorithm ideal for scenarios where compression is performed once, but decompression needs to be done frequently, like remote sensing or onboard satellite data analysis.

  - Reduced Memory Footprint: Traditional methods, such as JPEG2000 and PCA-DCT, require storing large matrices or transformations. By contrast, our method only stores modulations, which are low-dimensional parameters, making it more memory-efficient.

  - Training Acceleration through Meta-Learning: The use of meta-learning ensures that fewer gradient descent steps are needed during training and testing. This saves computational time, especially when dealing with large-scale data, as the initialized parameters Θ^{+} are already close to optimal. Thus, the method is particularly well-suited for iterative tasks requiring fast adaptation, such as incremental learning or dynamic environmental monitoring.

- Trade-offs and Limitations

  - Compression Time: One of the trade-offs in the proposed approach is the relatively longer compression time during the training phase. As highlighted in the experimental results (see Table 5.3), the compression process can take longer than traditional methods due to the multiple gradient steps involved. This may limit its applicability for real-time compression scenarios where quick encoding is required. However, this limitation can be mitigated through sampling techniques or by leveraging hardware accelerations (e.g., GPUs).

  - Dependency on Training Data: The quality of compression is heavily influenced by the training data used to fit the base network. If the datasets used for training are not diverse enough, the base network may not generalize well to new types of hyperspectral data, leading to lower reconstruction quality during decompression.

  - Lossy Nature and PSNR Variation: Although the method achieves competitive Peak Signal-to-Noise Ratio (PSNR) scores compared to other techniques, the reconstructed images may still exhibit some minor artifacts at extremely low bit rates. In particular, performance can vary across datasets with different spectral and spatial properties, such as Cuprite versus Indian Pines. The performance variability suggests that fine-tuning is necessary when applying the method across highly heterogeneous datasets.

- Scalability and Future Directions

  - Handling Large Datasets: The modular design of the proposed method makes it scalable to larger datasets. For example, the shared base network can be pretrained on large datasets, and modulations can be quickly fine-tuned for new hyperspectral images. This scalability is crucial for real-world applications, such as climate monitoring, where hyperspectral images are collected over time and need to be efficiently compressed.

  - Incorporating Additional Layers of Modulation: Future work can explore more sophisticated modulation schemes incorporating shifts and scales (rather than just shifts) to enhance performance. Additionally, using hierarchical or multi-scale modulation layers could improve the network's ability to capture fine-grained details across different spectral bands.

  - Transfer Learning and Domain Adaptation: Transfer learning techniques could further enhance the method by enabling the shared base network to adapt quickly to unseen datasets. Similarly, domain adaptation strategies can be integrated to handle datasets with different spatial resolutions or spectral characteristics, improving the generalizability of the approach.

- Hardware Optimization and Parallelization: As the method relies on neural networks, further research could focus on optimizing hardware implementations. For instance, deploying the method on specialized hardware such as TPUs or FPGAs could significantly speed up both the compression and decompression phases, making the method suitable for real-time applications.

This discussion highlights both the advantages and limitations of the proposed method. The method demonstrates a strong potential to become a state-of-the-art solution for hyperspectral image compression, especially in scenarios requiring high compression rates, fast decompression, and efficient storage.

5.5 Implementation Details and Practical Considerations

5.5.1 Dealing with Images Having Different Numbers of Channels

To handle hyperspectral images with varying numbers of channels, we standardize the output by padding the channel dimensions. The padding ensures that all images, regardless of their original number of spectral channels, match the fixed output dimensionality the base network requires. This padding is performed in a way that does not distort the spectral information of the image. Specifically, zero-padding is employed to extend the smaller channel dimensions to align with the maximum channel size expected by the network. This approach allows the base network to process diverse hyperspectral datasets without requiring structural modifications to the network.

5.5.2 File Format of the Compressed Image

The compressed representation of an image is stored in a .pt file, a format commonly used in PyTorch for saving tensors and models. In this file, the latent modulations, which are vectors representing the compressed image, are saved. These modulations encapsulate the image-specific information required to reconstruct the image when used with the shared base network. This efficient representation significantly reduces storage requirements compared to conventional compression methods, as only the modulations need to be saved for each image, while the shared base network parameters remain constant. We do not need to save the base network parameters, as we assume that people who want to use the transferred hyperspectral image already have access to them. When the modulations of a new hyperspectral image are sent, the recipients only need these modulations, along with the existing base network parameters, to decompress the image.

Only the image-specific modulations need to be saved for each image, while the shared base network parameters are assumed to be available at the decompression stage. The total number of bits required to save the compressed representation of an image can be formally expressed as:

$$\text{Total bits required for storage} = \underbrace{\text{bits per parameter}}_{bpp} \times \underbrace{\text{number of modulation parameters}}_{n_\phi}.$$

Here, bpp denotes the precision of each parameter (for example, 32 for float32 or 16 for float16), and n_ϕ represents the total number of modulation parameters specific to the image. This approach ensures efficient compression, as the storage requirements scale with the size of the modulations rather than the full network parameters.

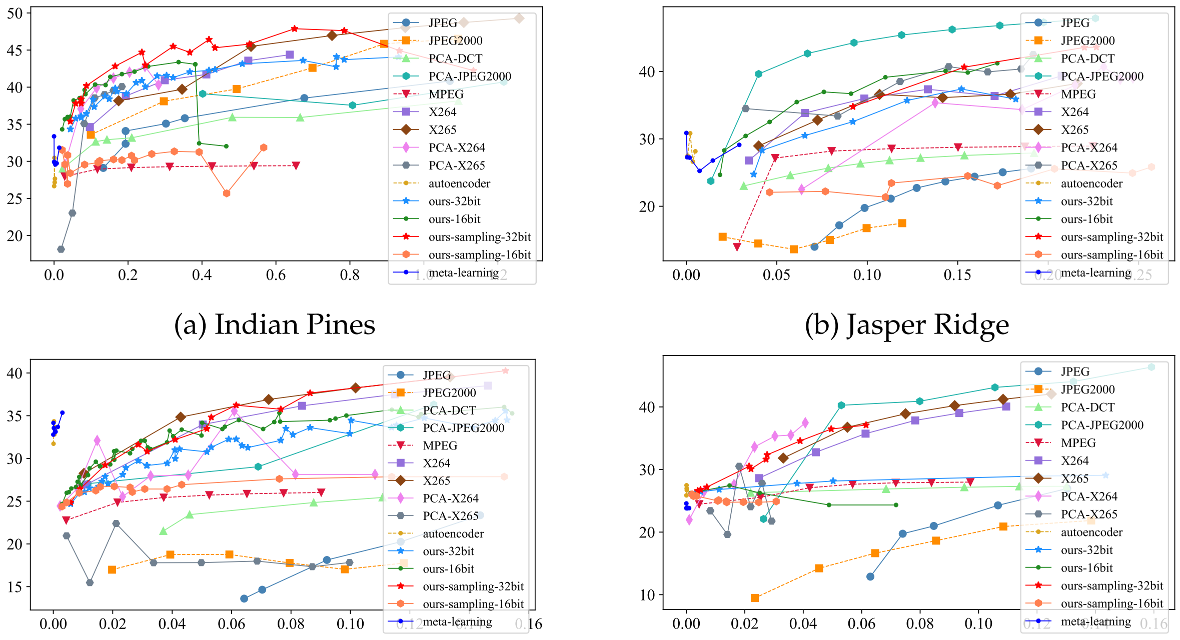
5.5.3 Compressing a New Image Using the Base Network

Compressing a new image using the proposed method does not require retraining the shared base network. Instead, the modulations specific to the new image are trained while keeping the base network parameters fixed. This involves initializing a latent vector for the new image and optimizing it through gradient descent to minimize the reconstruction loss. Once optimization is complete, the trained modulations for the new image are saved in a .pt file, representing the compressed image. This approach leverages the generalization capabilities of the pre-trained base network, enabling fast and efficient compression of new hyperspectral images.

5.6 Experiments

In line with prior studies, we utilize key performance metrics to assess the efficacy of our proposed compression technique. These metrics, including Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and bits-per-pixel-per-band ($bpppb$), are elaborated in Chapter 3. PSNR, a widely used measure in image compression, quantifies the difference in quality between the original image and its compressed counterpart, with higher values indicating superior quality. MSE provides a complementary metric by measuring the overall deviation between the original and compressed images, with smaller values indicating better reconstruction. Finally, $bpppb$ quantifies the compression rate achieved, where smaller values reflect higher compression. Details of these metrics, along with their formulations and significance, are provided in Chapter 3.

The datasets used to evaluate the effectiveness of our methodology are detailed in Chapter 3. We selected these datasets—Indian Pines, Jasper Ridge, Pavia University, and Cuprite—due to their frequent use in hyperspectral image compression studies. Each dataset presents unique challenges and characteristics that make them especially



(a) Indian Pines

(b) Jasper Ridge

(c) Pavia University

(d) Cuprite

Figure 5.2: **Compression results**. PSNR values achieved at various *bpppb* for our method, along with those obtained by other methods. Here "ours-32bit" and "ours-sampling-32bit" refer to our method where parameters are stored at 32-bit precision, and "ours-16bit" and "ours-sampling-16bit" refer to results when parameters are stored at 16-bit precision [Rezasoltani and Qureshi, 2023b], and meta-learning refers to the proposed method in this thesis.

suitable for studying meta-learning-based hyperspectral image compression, where the goal is to learn a base model that can quickly adapt to new scenes or acquisition conditions with minimal fine-tuning. By spanning a wide spectrum of scene types, the selected datasets enable a comprehensive evaluation of both the cross-domain generalization and the adaptability of our meta-learned hyperspectral compression model. This diversity ensures that the performance metrics obtained reflect not only the reconstruction quality but also the flexibility and robustness of the proposed method in practical, real-world scenarios. Moreover, the datasets vary in spatial resolution, number of spectral bands, and noise levels, providing a challenging yet realistic environment to validate the benefits of meta-learning over traditional compression techniques that typically lack the capacity to generalize across domains without retraining.

Indian Pines				
Method	Size (KB)	PSNR	bpppb	n_h, w_h
-	9251	∞	16	-,-
JPEG [Good et al., 1994, Qiao et al., 2014]	115.6	34.085	0.2	-,-
JPEG2000 [Du and Fowler, 2007]	115.6	35.84	0.2	-,-
PCA-DCT [Nian et al., 2016]	115.6	33.173	0.2	-,-
PCA-JPEG2000 [Du and Fowler, 2007]	115.6	39.5	0.2	-,-
FPCA-JPEG2000 [Mei et al., 2018]	115.6	40.5	0.2	-.-
3D SPECK [Tang and Pearlman, 2006]	115.6	-	0.2	-,-
3D-DCT [Yadav and Nagmode, 2018]	115.6	-	0.2	-,-
3D-DWT-SVR [Zikiou et al., 2020]	115.6	-	0.2	-,-
WSRC [Ouahioune et al., 2021]	115.6	-	0.2	-,-
ours-32bit [Rezasoltani and Qureshi, 2023a]	115.6	42.22	0.2	5,60
ours-16bit [Rezasoltani and Qureshi, 2023a]	57.5	29.68	0.1	5,60
ours-sampling-32bit [Rezasoltani and Qureshi, 2023b]	115.6	**42.22**	0.2	5,60
ours-sampling-16bit [Rezasoltani and Qureshi, 2023b]	57.5	29.68	0.1	5,60
meta-learning	0.003	33.36	6.9e-6	10,128
Jasper Ridge				
Method	Size (KB)	PSNR	bpppb	n_h, w_h
-	4800	∞	16	-,-
JPEG [Good et al., 1994, Qiao et al., 2014]	30	21.130	0.1	-,-
JPEG2000 [Du and Fowler, 2007]	30	21.130	0.1	-,-
PCA-DCT [Nian et al., 2016]	30	26.821	0.1	-,-
PCA-JPEG2000 [Du and Fowler, 2007]	30	-	0.1	-,-
FPCA-JPEG2000 [Mei et al., 2018]	30	-	0.1	-,-
3D SPECK [Tang and Pearlman, 2006]	30	-	0.1	-,-
3D DCT [Yadav and Nagmode, 2018]	30	-	0.1	-,-
3D-DWT-SVR [Zikiou et al., 2020]	30	-	0.1	-,-
WSRC [Ouahioune et al., 2021]	30	-	0.1	-,-
ours-32bit [Rezasoltani and Qureshi, 2023a]	30	32.54	0.1	5,20
ours-16bit [Rezasoltani and Qureshi, 2023a]	15	22.07	0.06	5,20
ours-sampling-32bit [Rezasoltani and Qureshi, 2023b]	30	**34.77**	0.1	5,20
ours-sampling-16bit [Rezasoltani and Qureshi, 2023b]	15	22.07	0.06	5,20
meta-learning	0.003	30.87	1.4e-5	10,128

Table 5.1: Compression results. Our proposed method (meta-learning) got a comparable PSNR with other method for the two datasets, even with a higher compression rate (small bpppb) and smaller image compression size

Pavia University				
Method	Size (KB)	PSNR	bpppb	n_h, w_h
-	42724	∞	16	-,-
JPEG [Good et al., 1994, Qiao et al., 2014]	267	20.253	0.1	-,-
JPEG2000 [Du and Fowler, 2007]	267	17.752	0.1	-,-
PCA-DCT [Nian et al., 2016]	267	25.436	0.1	-,-
PCA-JPEG2000 [Du and Fowler, 2007]	267	33	0.1	-,-
FPCA-JPEG2000 [Mei et al., 2018]	267	-	0.1	-,-
3D-SPECK [Tang and Pearlman, 2006]	267	-	0.1	-,-
3D-DCT [Yadav and Nagmode, 2018]	267	-	0.1	-,-
3D-DWT-SVR [Zikiou et al., 2020]	267	-	0.1	-,-
WSRC [Ouahioune et al., 2021]	267	-	0.1	-,-
ours-32bit [Rezasoltani and Qureshi, 2023a]	267	34.46	0.1	10,80
ours-16bit [Rezasoltani and Qureshi, 2023a]	133.5	34.17	0.05	10,80
ours-sampling-32bit [Rezasoltani and Qureshi, 2023b]	267	**38.08**	0.1	10,80
ours-sampling-16bit [Rezasoltani and Qureshi, 2023b]	133.5	27.49	0.05	10,80
meta-learning	0.0032	32.79	1.4e-6	10,128
Cuprite				
Method	Size (KB)	PSNR	bpppb	n_h, w_h
-	140836	∞	16	-,-
JPEG [Good et al., 1994, Qiao et al., 2014]	880.2	24.274	0.1	-,-
JPEG2000 [Du and Fowler, 2007]	880.2	20.889	0.1	-,-
PCA-DCT [Nian et al., 2016]	880.2	27.302	0.1	-,-
PCA-JPEG2000 [Du and Fowler, 2007]	880.2	**40.90**	0.1	-,-
FPCA-JPEG2000 [Mei et al., 2018]	880.2	-	0.1	-,-
3D-SPECK [Tang and Pearlman, 2006]	880.2	27.1	0.1	-,-
3D-DCT [Yadav and Nagmode, 2018]	880.2	33.4	0.1	-,-
3D-DWT-SVR [Zikiou et al., 2020]	880.2	28.20	0.1	-,-
WSRC [Ouahioune et al., 2021]	880.2	35	0.1	-,-
ours-32bit [Rezasoltani and Qureshi, 2023a]	880.2	28.954	0.1	25,100
ours-16bit [Rezasoltani and Qureshi, 2023a]	440.1	24.334	0.06	25,100
ours-sampling-32bit [Rezasoltani and Qureshi, 2023b]	880.2	36.55	0.1	25,90
ours-sampling-16bit [Rezasoltani and Qureshi, 2023b]	440.1	24.91	0.06	25,90
meta-learning	0.003	24.57	4.5e-7	10,128

Table 5.2: Compression results. Our proposed method (meta-learning) got a comparable PSNR with other method for the two datasets, even with a higher compression rate (small bpppb) and smaller image compression size

Dataset	Method	bppppb	compression time (Sec)	decompression time (Sec)	PSNR ↑
Indian Pines	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.1	7.353	3.27	27.47
	JPEG2000$^+$ [Du and Fowler, 2007]	0.1	0.1455	0.3115	33.58
	PCA-DCT$^+$ [Nian et al., 2016]	0.1	1.66	0.04	32.28
	ours-32bit	0.1	243.64	0.0001	36.98
	ours-16bit	0.05	243.64	0.0001	36.95
	ours-sampling-32bit	0.1	282.08	0.0005	**40.1**
	ours-sampling-16bit	0.05	282.08	0.0005	28.40
	meta-learning	6.9e-6	**0.033**	**0.000717**	33.36
Jasper Ridge	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.1	3.71	1.62	21.13
	JPEG2000$^+$ [Du and Fowler, 2007]	0.1	0.138	0.395	17.49
	PCA-DCT$^+$ [Nian et al., 2016]	0.1	1.029	0.027	26.82
	ours-32bit	0.1	312.38	0.0005	32.54
	ours-16bit	0.06	312.38	0.0005	32.51
	ours-sampling-32bit	0.1	75.91	0.0005	**34.77**
	ours-sampling-16bit	0.06	75.91	0.0005	22.07
	meta-learning	1.4e-5	**0.025**	**0.0007**	30.87
Pavia University	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.1	33.86	14.61	20.25
	JPEG2000$^+$ [Du and Fowler, 2007]	0.1	0.408	0.628	17.75
	PCA-DCT$^+$ [Nian et al., 2016]	0.1	6.525	0.235	25.43
	ours-32bit	0.1	780.16	0.0009	34.46
	ours-16bit	0.05	780.16	0.0009	34.17
	ours-sampling-32bit	0.1	72.512	0.0004	**38.08**
	ours-sampling-16bit	0.05	72.512	0.0004	27.02
	meta-learning	1.4e-6	**0.43**	**0.0006**	32.79
Cuprite	JPEG$^+$ [Good et al., 1994, Qiao et al., 2014]	0.06	101.195	45.02	12.88
	JPEG2000$^+$ [Du and Fowler, 2007]	0.06	1.193	2.476	15.16
	PCA-DCT$^+$ [Nian et al., 2016]	0.06	11.67	0.754	26.75
	ours-32bit	0.06	1565.97	0.001	28.02
	ours-16bit	0.03	1565.97	0.001	27.90
	ours-sampling-32bit	0.06	664.87	0.001	**37.27**
	ours-sampling-16bit	0.03	664.87	0.001	24.85
	meta-learning	4.5e-7	**1.11**	**0.0007**	24.57

Table 5.3: The effect of using the meta-learning method on compression times. ours-32bit is a learning-based method that requires us to train an MLP on the input image. Consequently, compression times for our method are significantly larger than those of other schemes listed here. We attempt to address this somewhat through sampling. A sampling rate of 20% cuts the compression time by half. The good news is our method, with or without sampling, achieves good decompression times. Indeed the proposed method achieves faster decompression times than JPEG, JPEG2000, and PCA-DCT methods shown in this table. This suggests that the proposed method is well-suited to "compress-once" sort of applications. It is evident that our approach, meta-learning, enhances the compression time. Text decoration $^+$ indicates a classical method.

5.6.1 Comparison with other methods

Figure 5.2 shows PSNR values at various compression rates for different methods. Specifically, we compare our approaches, labeled as *meta-learning*, *ours-32bit*, *ours-16bit*, *ours-sampling-32bit*, and *ours-sampling-16bit*, with JPEG, JPEG2000, PCA-DCT, MPEG, X264, X265, PCA-X264, PCA-X265 PCA-JPEG2000, and autoencoder [La Grassa et al., 2022] methods. Here, *ours-32bit* and *ours-sampling-32bit* methods store MLP weights as 32-bit floating point values, whereas *ours-16bit* and *ours-sampling-16bit* store MLP weights at half-precision as 16-bit floating point values that are constructed by quantizing the MLP weights. These plots illustrate that our methods achieve higher compression quality, i.e., better PSNR, for a given value of $bpppb$. This is especially true for lower $bpppb$ values.

For the Indian Pines dataset, the *ours-32bit* method achieves superior PSNR up to approximately 0.7 $bpppb$, after which the X264 method surpasses it in performance. Interestingly, the PSNR for the *ours-16bit* method drops significantly at around 0.4 $bpppb$, which suggests that this behavior requires further investigation. One possible explanation is the effect of weight quantization. In our implementation, we quantize the network weights from 32-bit to 16-bit floating-point precision to reduce storage. However, this quantization may result in a loss of critical information, particularly for weights that fall outside the representable range of 16-bit floating point. In PyTorch's 16-bit format (FP16), precision is sacrificed to gain a broader dynamic range, and values outside the representable range may be clipped or rounded, potentially degrading the network's ability to accurately reconstruct high-fidelity representations at lower bit rates. However, the *ours-sampling-32bit* method outperforms all other methods across various $bpppb$ levels, demonstrating its strength in balancing quality and compression.

In the case of Jasper Ridge, the *ours-sampling-32bit* and *ours-16bit* methods show better results than the *ours-32bit* method, with higher PSNR values achieved at multiple bit rates. These two methods surpass most of the competing approaches except for PCA-JPEG2000, which slightly edges them in certain conditions.

For the Pavia University dataset, the *our* and *ours-16bit* methods exhibit superior performance at lower bit rates compared to other methods. Additionally, the *ours-sampling-32bit* method achieves the highest PSNR among all the approaches when evaluated at around 0.1 $bpppb$ and higher bit rates, confirming its effectiveness in maintaining quality under varying compression constraints.

Regarding the Cuprite dataset, the *our* method delivers higher PSNR than most other methods, with the exception of PCA-JPEG2000, X264, and X265, particularly at lower bit rates. Similarly, the *ours-sampling-32bit* method consistently performs better than other methods at multiple $bpppb$ levels, although PCA-X264 and PCA-JPEG2000 remain competitive at certain low bit rates.

Across all four datasets, the meta-learning-based compression approach demonstrates a distinct advantage by achieving higher PSNR values at extremely low $bpppb$ levels, where most competing methods are unable to perform effective compression.

From these results, the following conclusions can be drawn: (1) the proposed method achieves high-quality compression even at high compression ratios, indicating its robustness and efficiency, and (2) the quality of compression achieved by our method consistently surpasses that of the three widely used compression methods for hyperspectral images. This highlights the potential of our approach in addressing the challenges of hyperspectral image compression more effectively than traditional techniques.

5.6.2 Model Fitting

We utilize PyTorch [Paszke et al., 2019] to implement all of our models. In the inner loop, we employ Stochastic Gradient Descent (SGD) with a learning rate of 1e-2. In the outer loop, we utilize the Adam optimization algorithm with a learning rate of either 1e-6 or 3e-6. We standardize the values of the coordinates x to be within the range of -1 to 1 and the features y to be within the range of 0 to 1.

5.6.3 Compression Results

The compression results produced by the proposed method in this chapter (meta-learning), our previous methods (ours-32bit) [Rezasoltani and Qureshi, 2023a], (ours-16bit) [Rezasoltani and Qureshi, 2023a], (ours-32bit-sampling) [Rezasoltani and Qureshi, 2023b], and (ours-16bit-sampling) [Rezasoltani and Qureshi, 2023b]), and a number of classical and learning-based hyperspectral image compression approaches on the four datasets are listed in Tables 5.1 and 5.2. The table further displays the dimensions of the original, uncompressed hyperspectral images.

Our proposed method (meta-learning) got better PSNR than JPEG, JPEG2000, and PCA-DCT for all four datasets, even with a higher compression rate (small bpppb) and smaller image compression size. Moreover, our proposed method got nearly similar PSNR to our previous methods (ours-32bit, ours-16bit, ours-32bit-sampling, and ours-16bit-sampling) with a higher compression rate and smaller image compression size.

Table 5.3 displays the compression time, decompression time, and PSNR (Peak Signal-to-Noise Ratio) results for the approach presented in this chapter (meta-learning), our previous methods(ours-32bit, ours-16bit, ours-32bit-sampling, and ours-16bit-sampling), as well as JPEG, JPEG2000, and PCA-DCT for the four datasets. It is evident that our approach, meta-learning, enhances the compression time. We got better compression and decompression time than JPEG, JPEG2000, PCA-DCT, ours-32bit, ours-16bit, ours-32bit-sampling, and ours-16bit-sampling, while PSNR remains better than JPEG, JPEG2000, and PCA-DCT and close to ours-32bit, ours-16bit, ours-32bit-sampling, and ours-16bit-sampling.

5.7 Ablative Study

In this section, we investigate the impact of training our meta-learned compression network on different combinations of datasets and evaluate its generalization ability. Specifically, we perform two experiments:

- Case 1: Train on Indian Pines, Jasper Ridge, and Pavia University, and test on Cuprite.
- Case 2: Train only on Indian Pines and Jasper Ridge, and test on Cuprite.

The goal of this study is to examine how training with a broader or narrower set of hyperspectral datasets influences the model's performance, particularly on unseen data. This analysis gives insights into how the diversity of training data affects the compression times, decompression times, and PSNR on new datasets. Below are the results for the two cases.

5.7.1 Results and Analysis

In Case 1, training on Indian Pines, Jasper Ridge, and Pavia University results in the following:

- The PSNR values for the training datasets are 36.47, 38.61, and 35.58, respectively.
- The PSNR value for the test dataset (Cuprite) is 14.09 with 10 inner steps.
- The compression time is approximately 0.0317 seconds, and the decompression time is 0.0016 seconds.

In Case 2, where training was done only on Indian Pines and Jasper Ridge, the results are as follows:

- The PSNR values for the training datasets are 37.79 and 40.44, respectively.
- The PSNR value for the test dataset (Cuprite) is 13.88, with 10 inner steps.
- The compression time is 0.0295 seconds, and the decompression time is 0.0010 seconds.

These results suggest that including additional training datasets may help the model achieve better generalization, but the improvements are nuanced. Although Case 1, with three datasets, achieves slightly better PSNR on Cuprite than Case 2, the compression and decompression times are slightly higher.

5.7.2 Discussion

As seen in Table 5.4, training with three datasets in Case 1 achieves a marginally better PSNR (14.09) on the Cuprite dataset compared to Case 2 (13.88). However, the compression time in Case 1 is slightly longer (0.0317 seconds) than in Case 2 (0.0295 seconds). This indicates a trade-off between the diversity of the training data and the computational efficiency of the model.

The decompression times are quite small in both cases, indicating that the meta-learned model can reconstruct images rapidly, regardless of the size of the training set. Notably, while the PSNR difference between the two cases is minor, the inclusion of Pavia University in the training set seems to have contributed positively to generalization, albeit with a slight increase in compression time.

These findings demonstrate the potential value of including diverse datasets in the training phase to improve generalization to unseen datasets like Cuprite. However, the marginal gain in PSNR suggests that further experimentation might be needed to explore the optimal balance between training data diversity and computational cost.

Case	Training Datasets	Test Dataset	Compression Time (Sec)	Decompression Time (Sec)	PSNR (Test)
1	Indian Pines, Jasper Ridge, Pavia	Cuprite	0.0317	0.0016	14.09
2	Indian Pines, Jasper Ridge	Cuprite	0.0295	0.0010	13.88

Table 5.4: Ablative study results showing the effect of training with different datasets on the compression performance when tested on the Cuprite dataset. The table provides a comparison of compression and decompression times, along with PSNR values for each test configuration.

5.8 Large dataset

We apply a large dataset to our meta-learned network, demonstrating that our method is capable of compressing large images and performing progressive compression. To manage the complexity of the data, we divide the image into 7 by 7 grids, effectively treating each row as a separate cell. Each row is compressed using a distinct network specifically customized to capture the unique characteristics and patterns of that row. In total, we train seven networks, one for each row, ensuring that the model is tailored to the data in each cell. characteristics and patterns of that row. The dataset used in our experiments is substantial, with a size of 28.2 GB and dimensions of 4192 by 6708 pixels across 270 channels. This large-scale application not only validates the scalability of our approach but also showcases its effectiveness in handling high-dimensional data while maintaining compression efficiency and image quality. Additionally, the flexibility of our method allows it to be adapted to various image sizes, making it a versatile tool in image compression. The size of the compressed image with the meta-leaned-based approach is 141 KB. Figure 5.3 shows an image of this dataset. In this study, we compare the performance of our proposed method against ours-sampling-32bit, ours-sampling-16bit, JPEG, and MPEG, with a focus on the (PSNR) achieved by each method. Due to the size limitations inherent in JPEG compression, we divided the image into smaller segments and applied JPEG compression to each piece separately. The PSNR was then calculated for the recombined image. Table 5.5 shows this comparison. Our method (meta-learning) achieves a PSNR comparable to both MPEG

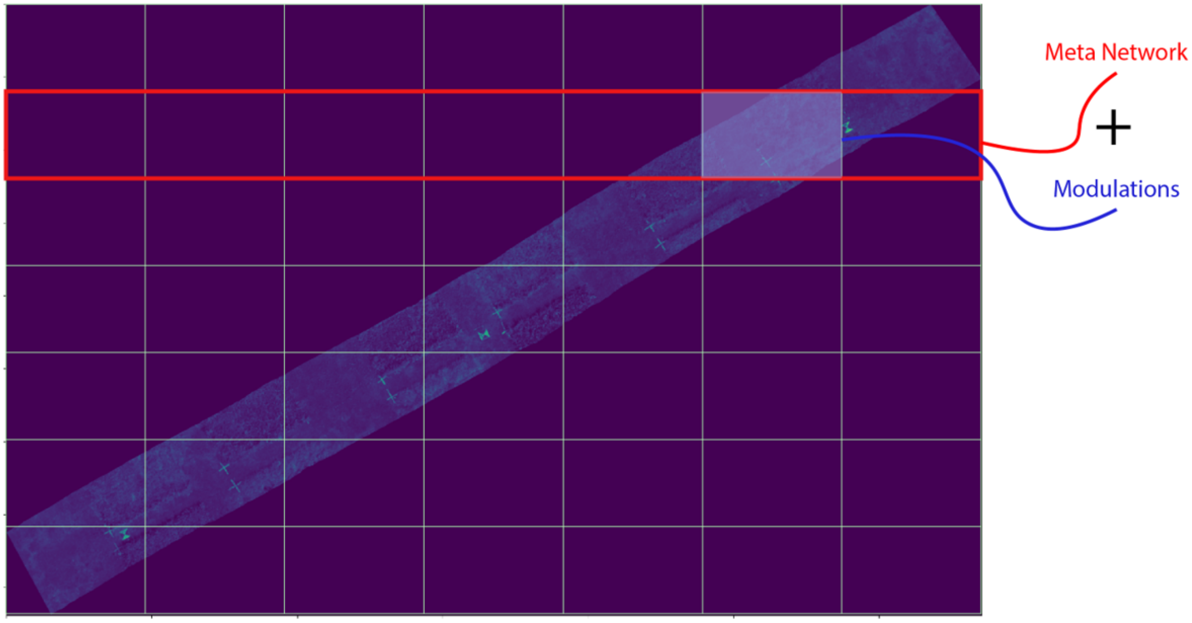


Figure 5.3: Visualization of the dataset used in this study, consisting of images with dimensions 4192 by 6708 pixels and 270 channels. The dataset has a total size of 28.2 GB, demonstrating the scale and complexity of the data processed by our meta-learned network for image compression.

and JPEG while offering a significant reduction in bits per pixel per band (bpppb), resulting in smaller compressed image sizes.

5.9 Conclusion

In this work, we use the meta-learned base network in which we train a base network shared across hyperspectral images and apply modulations to this network to parameterize individual hyperspectral images. Our system demonstrates substantial enhancements in performance as compared to our previous methods [Rezasoltani and Qureshi, 2023a, Rezasoltani and Qureshi, 2023b], namely in terms of compression and encoding time. Furthermore, it remains competitive with existing codecs like JPEG, JPEG2000, and PCA-DCT.

method	PSNR	bpppb	size(KB) ↓
meta-learning	23.40	4.6e-6	141
ours-sampling-32bit	25.23	0.017	16133
ours-sampling-16bit	25.23	0.008	7592
JPEG	20.21	0.242	229669
MPEG	25.20	0.024	22777

Table 5.5: Comparison of different image compression methods, including our meta-learning approach, ours-sampling-32bit, ours-sampling-16bit, JPEG, and MPEG, based on Peak Signal-to-Noise Ratio (PSNR), bits per pixel per band (bpppb), and compressed file size (in bytes). The meta-learning method demonstrates a competitive PSNR and significantly reduced file size compared to JPEG and MPEG.

Chapter 6

Region-Specific Image Compression

In many real-world applications, different regions of an image hold varying levels of importance depending on the specific task or analysis being performed. The relevance of a particular region is inherently task-dependent. For instance, in a hyperspectral image capturing both a lake and a farmland, a researcher studying water quality may prioritize spectral information from the lake, whereas an agricultural analyst assessing crop health may focus on the farmland. This variability in importance suggests that a compression scheme that treats all regions in an image the same may not be ideal. This observation led us to explore a differential compression scheme that compresses critical or high-importance regions with higher fidelity than regions that are perhaps less relevant to the task at hand. Less important regions are compressed at a lower quality, resulting in faster compression times.

The chapter is organized as follows. First, we demonstrate that it is possible to achieve differential compression using implicit neural representation by controlling the sampling rates for different regions. Next, we explore two schemes for selecting regions of interest. Firstly, we use K-Means clustering to group pixels into contiguous regions based upon spectral similarity. Next, we employ a U-net-based deep learning model trained to segment aerial images into regions of interest. Both approaches segment the hyperspectral images into distinct regions that are subsequently compressed at different rates, showcasing the benefits of the approach developed in this chapter.

6.1 On Sampling and Compression Rates

Previously, we have shown that (1) it is possible to compress a hyperspectral image using implicit neural representations and that (2) it is not necessary to consider every pixel location when learning the mapping between pixel coordinates and pixel spectra. Furthermore, it is possible to achieve high-quality compression even when sampling a subset of pixels. An additional advantage of (2) is that it results in significant savings in terms of compression times. Here, we take this idea further and show that it is possible to train a single implicit neural network while using different sampling rates in different regions of an image. Furthermore, the reconstruction quality of these regions depends upon the sampling rates used for these regions. Regions where pixels were sampled more densely post low reconstruction errors, whereas regions where pixels were sampled sparsely show higher reconstruction errors. We note that compression times increase as sampling rates increase, which suggests that using lower sampling rates in less important regions of the image may result in significant time savings.

6.1.1 Experiments with Differential Compression

In order to study whether or not it is possible to perform differential compression by leveraging varying sampling rates across different regions of a hyperspectral image, we set up the following experiment. We take the four benchmarks that we have access to—Indian Pines, Jasper Ridge, Pavia University, and Cuprite—and we split each image into two halves. Next, we compress each hyperspectral image by fitting an implicit neural network model as discussed previously. At training time, however, one of the two halves is sampled at 90% whereas the other half is sampled at 5%. The purpose of this exercise is to confirm if it is possible to use a single implicit neural network that maps pixel coordinates to their spectra under conditions where different regions are sampled at different rates during learning (i.e., at compression time).

6.2 Hyperspectral Image Segmentation Using K-Means

K-means clustering is a widely used unsupervised learning algorithm that effectively partitions data into distinct groups based on feature similarity. In the context of hyperspectral image segmentation, K-Means can be particularly valuable due to its ability to handle the high-dimensional data characteristic of hyperspectral images [Macqueen, 1967].

The K-Means algorithm operates by initializing a predetermined number of cluster centroids (K). It iteratively assigns each pixel to the nearest centroid based on a distance metric (e.g., Euclidean distance). The algorithm aims to minimize the within-cluster variance, which is mathematically expressed as:

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2, \tag{6.1}$$

where J represents the objective function, C_k is the set of points in cluster k, x_i is the i-th pixel, and μ_k is the centroid of cluster k. During each iteration, each pixel x_i is assigned to the nearest cluster centroid based on the following rule:

$$C_k = \{x_i : \|x_i - \mu_k\|^2 \leq \|x_i - \mu_j\|^2 \, \forall j, \, 1 \leq j \leq K\}, \tag{6.2}$$

where C_k is the set of all pixels assigned to cluster k, after the assignments are made, the centroids are recalculated as the mean of the pixels within each cluster:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i. \tag{6.3}$$

This process repeats until the centroids stabilize or the number iterations are exhausted, leading to a final set of clusters [Grewal et al., 2023].

It is straightforward to apply K-Means clustering to hyperspectral images. The hyperspectral image is reshaped such that each row represents a pixel and each column represents a spectral band. K-Means clustering is applied to the reshaped image, resulting in a cluster label for each pixel. Pixels that belong to the same cluster exhibit similar spectral signatures if spectral similarity is used as a distance measure during the clustering process. We note that this process does not respect the spatial structure present in an image. As such, the process may yield non-contiguous regions. However, in practice, the process yields contiguous regions.

6.2.1 Why K-Means for Hyperspectral Image Segmentation

In hyperspectral images, each pixel contains spectral information at several wavelengths, allowing detailed material identification and classification. By applying K-Means clustering to hyperspectral data, it is possible to segment the image into distinct regions based upon spectral characteristics. Within this context, K-Means offers a number of advantages:

- Simplicity: K-Means is straightforward to implement and computationally efficient, making it suitable for real-time applications.

- Scalability: The algorithm scales well with the number of pixels and can be applied to large hyperspectral datasets.

- Interpretability: The resulting segments can be easily interpreted, as they correspond to distinct spectral classes.

K-Means clustering offers a lightweight scheme for grouping pixels together based on their spectral characteristics. It is no wonder that K-Means clustering continues to be widely used in the hyperspectral community to facilitate downstream analysis tasks. For the purposes of the work proposed in this chapter, the proposed scheme is able to compress these regions using different sampling rates.

6.3 Hyperspectral Image Segmentation using U-Net

To carry out the idea of differential compression further, we also experimented with a U-Net-based model for hyperspectral image segmentation. U-Net is a popular choice for image segmentation due to its ability to delineate objects in complex imagery. We used the U-Net model developed to perform semantic segmentation in aerial images [in the Loop, 2020]. The code is available at [Hajebi, 2020].

Once trained, the U-Net was applied to segment each hyperspectral dataset into regions based on predefined classes of interest. This segmentation process can be formulated as a pixel-wise classification problem. For each pixel x_i in the hyperspectral image, the U-Net predicts a class label $y_i \in \{1, 2, \ldots, C\}$, where C is the total number of classes. The prediction for each pixel can be expressed as:

$$y_i = \arg \max_{c \in \{1,2,\ldots,C\}} P(y_i = c \mid x_i; \Theta), \tag{6.4}$$

where $P(y_i = c \mid x_i; \Theta)$ is the probability that pixel x_i belongs to class c, and Θ represents the model parameters.

U-Net provides segmentation masks that identify pixels representing different “items,” “objects,” or “features.” This information can be subsequently used to identify *regions of interest*. It is then possible to treat these regions differently, compressing regions of interest in higher fidelity by controlling sampling rates appropriately.

6.3.1 U-Net Architecture and Customization for Aerial Image Segmentation

The U-Net architecture, originally developed for biomedical image segmentation, effectively captures local and global image features. It follows an encoder-decoder design with skip connections, which enables the model to retain spatial resolution and fine details during upsampling. The encoder path is defined as:

$$E_l = f(W_l * E_{l-1} + b_l), \tag{6.5}$$

where E_l represents the output of the l-th encoder layer, W_l and b_l are the weights and biases for that layer, $*$ denotes the convolution operation, and f is the activation function.

The decoder path reconstructs the segmentation map using upsampling layers and concatenated skip connections:

$$D_l = f(W_l' * \text{Upsample}(D_{l+1}) + S_l + b_l'), \tag{6.6}$$

where D_l is the output of the l-th decoder layer, S_l is the skip connection from the encoder, and W_l' and b_l' are the weights and biases for the decoder layer.

Key modifications to the U-Net for aerial image segmentation included adjustments to convolutional filter sizes and additional dropout layers to improve generalization. The training process employed a categorical cross-entropy loss function, defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c}, \tag{6.7}$$

where N is the number of pixels in the training set, C is the number of classes, $y_{i,c}$

is a binary indicator for whether pixel i belongs to class c, and $\hat{y}_{i,c}$ is the predicted probability for pixel i belonging to class c.

The modular design of U-Net was adapted to hyperspectral data by adjusting input layers and the number of filters in each layer to handle additional spectral channels. By using U-Net, we segmented hyperspectral datasets into ROIs with high precision, feeding these regions into a task-aware compression pipeline. Critical spectral details in the ROIs were preserved through high-fidelity compression, while background regions were compressed more aggressively to reduce overall data size.

6.4 Hyperspectral Image Compression using Sampling and INR

In this section, we use INRs to compress the region of interest in the hyperspectral image. The suggested compression technique comprises two stages. Step 1 does an architectural search. The objective is to identify an MLP that attains the maximum reconstruction accuracy within a specified $bpppb$ budget. Architecture search is conducted by overfitting various MLPs with differing quantities of hidden layers and varying widths of hidden layers to the hyperspectral image. Architecture search, however, results in extended compression durations. Step 2 entails quantizing and archiving the parameters of the overfitted MLP onto disk.

The compression process entails overfitting a SIREN network f_Θ to the region of interest in a hyperspectral image I [Dupont et al., 2021]. The dimensions w and h of the hyperspectral image are utilized to establish an input location grid on $[-1, +1] \times [-1, +1]$, and the MLP is trained to reconstruct the spectral signature of a pixel based on its location. The parameters $\breve{\Theta}$ of this overfitted MLP are quantized as $\check{\Theta}$. The MLP architecture comprises the number of hidden layers n_h, the widths of these layers w_h, as well as the width w, height h, and number of channels c of the original hyperspectral image I, in conjunction with $\check{\Theta}$, which collectively function as a compressed representation I_{encoded} of the hyperspectral image I. Parameters $\check{\Theta}$ are stored as either 32-bit floats or 16-bit floats. Training and inference necessitate 32-bit floating-point numbers, while quantization and dequantization are executed to transition between 32 and 16 bit representations. The overfitted MLP comprises

$$(w_h \times 2) + (w_h \times w_h)^{(n_h-1)} + (w_h \times c) \tag{6.8}$$

parameters.

The region of interest in the hyperspectral image is reconstructed from its compressed encoding I_{encoded} in the following manner: 1) Utilize n_h, w_h, and c to reconstruct f_Θ; 2) Dequantize $\check{\Theta}$ to obtain $\tilde{\Theta}$ and employ it to initialize the parameters of f_Θ; 3) Utilize the width w and height h to establish the input grid within the bounds of $[-1, +1] \times [-1, +1]$; 4) Evaluate $f_{\tilde{\Theta}}$ at each point in the input grid to reconstruct the image $\tilde{I}$.

6.5 Algorithm for Task-aware Hyperspectral Image Compression

In this section, we outline the step-by-step procedure of our task-aware hyperspectral image compression method. The proposed algorithm leverages K-means clustering in one part and U-net in another part to segment the image into regions of interest (ROI) and background, ensuring that critical areas are preserved with higher fidelity while less important regions are compressed more aggressively. The key challenge addressed by this approach is balancing compression efficiency with the need to maintain the spectral quality of essential regions. The method involves searching for an optimal MLP architecture to encode the ROI, followed by a quantization step to further reduce storage requirements. Meanwhile, the background is compressed using a simpler technique to minimize the bitstream size. During decompression, both the ROI and background are reconstructed separately and then merged to generate the final image. The following pseudocode summarizes the full process of task-aware hyperspectral image compression.

6.5.1 Pseudocode of the Proposed Method

Algorithm 3 Algorithm for Task-aware Hyperspectral Image Compression

```
Input: Hyperspectral image $I$ with dimensions $(w, h, c)$
Output: Compressed representation $I_{\text{encoded}}$
Step 1: Segmentation (K-means or U-Net)
Option A: K-means Clustering – Reshape $I$ into a 2D array; initialize $K$ clusters
repeat
    Assign each pixel $x_i$ to the nearest centroid using $C_k = \{x_i : \|x_i - \mu_k\|^2 \leq \|x_i - \mu_j\|^2\}$; Update centroids: $\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$
until Centroids converge
Generate mask $M$ for ROI and background
Option B: U-Net Segmentation – Load pre-trained U-Net; preprocess $I$ for input; Generate mask $M$: $y_i = \arg\max_c P(y_i = c \mid x_i; \Theta)$; Divide $I$ into ROI and background using $M$
Step 2: Architecture Search for ROI Compression
for each MLP structure $(n_h, w_h)$ do
    Initialize $f_\Theta$; train to minimize $\mathcal{L} = \|I_{\text{ROI}} - f_\Theta(X)\|^2$; Compute PSNR; if PSNR meets threshold, select $(n_h, w_h)$
Step 3: Compression and Quantization – Train selected $f_\Theta$ on ROI; quantize $\check{\Theta} \rightarrow \check{\Theta}$; Store $I_{\text{encoded, ROI}} = \{\check{\Theta}, n_h, w_h\}$
Step 4: Background Compression – Compress background region and store $I_{\text{encoded, BG}}$
Step 5: Decompression – Load $I_{\text{encoded, ROI}}$, dequantize $\check{\Theta} \rightarrow \tilde{\Theta}$; Initialize $f_{\tilde{\Theta}}$; reconstruct grid $[-1, +1] \times [-1, +1]$; Evaluate $f_{\tilde{\Theta}}$ at each $(x, y)$ in ROI: $y = f_{\tilde{\Theta}}(x, y)$; Reconstruct background and combine with ROI to form $\tilde{I}$
```

The pseudocode outlines a task-aware hyperspectral image compression pipeline. Below is a detailed breakdown:

- **Segmentation (Step 1):** This step isolates regions of interest (ROI) and background to enable task-aware compression. Users can choose between:

- **K-means Clustering (Option A):** An unsupervised method that partitions pixels into K clusters based on spectral similarity. Each pixel is assigned to the nearest centroid using:

  $$C_k = \{x_i : \|x_i - \mu_k\|^2 \leq \|x_i - \mu_j\|^2, \forall j, 1 \leq j \leq K\}. \tag{6.9}$$

  The centroids are updated iteratively:

  $$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i, \tag{6.10}$$

  until convergence, this approach is computationally efficient and suitable when no labeled data is available.

- **U-Net Segmentation (Option B):** A supervised deep learning-based method that employs a pre-trained U-Net model for high-precision segmentation. The U-Net predicts the class label for each pixel:

  $$y_i = \arg \max_{c \in \{1,2,\ldots,C\}} P(y_i = c \mid x_i; \Theta), \tag{6.11}$$

  leveraging its encoder-decoder structure with skip connections to generate a segmentation mask that separates the ROI from the background.

- **Architecture Search for ROI Compression (Step 2):** The ROI is compressed using an implicit neural representation (INR). The algorithm performs a search for the optimal multi-layer perceptron (MLP) structure, minimizing the reconstruction loss:

  $$\mathcal{L} = \|I_{\text{ROI}} - f_\Theta(X)\|^2. \tag{6.12}$$

  The compression quality is evaluated using the Peak Signal-to-Noise Ratio

(PSNR), ensuring the preserved spectral details meet the required threshold.

- **Compression and Quantization (Step 3):** The selected MLP is trained on the ROI. Its parameters are quantized:

$$\breve{\Theta} \rightarrow \check{\Theta}, \tag{6.13}$$

  where 32-bit floating-point values are converted to 16-bit to reduce storage size while maintaining fidelity, the compressed representation includes the quantized parameters and the MLP architecture details.

- **Background Compression (Step 4):** The background region, being less critical for downstream tasks, is compressed using a low-complexity algorithm to minimize the overall bitstream size.

- **Decompression (Step 5):** During decompression, the ROI and background are reconstructed separately and combined to form the final image. The process includes:

  - Dequantizing the parameters:

    $$\check{\Theta} \rightarrow \tilde{\Theta}, \tag{6.14}$$

    to initialize the MLP.

  - Reconstructing the ROI by evaluating the MLP on an input grid:

    $$y = f_{\tilde{\Theta}}(x, y). \tag{6.15}$$

  - Reconstructing the background and merging it with the ROI to generate the final image $\tilde{I}$.

This section presents a structured approach for task-aware hyperspectral image compression. By employing segmentation and selective compression, this method ensures that the critical spectral information in the ROI is preserved, while the background is efficiently compressed to minimize the total storage size. This makes the approach particularly effective for applications where specific regions of interest are more valuable than the entire image.

6.6 Experiments

In this thesis, we follow the established practice of utilizing Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) to assess the performance of the proposed compression method, as described in Chapter 3. These metrics allow us to quantify the quality loss between the original and compressed images. PSNR, expressed in decibels, indicates how closely the compressed image resembles the original—higher PSNR values correspond to higher image quality. Detailed descriptions of this metric, along with the relevant equations, can be found in Chapter 3.

To evaluate the performance of our proposed method, we used five benchmark datasets, as described in Chapter 3. These datasets—Indian Pines, Cuprite, Jasper Ridge, Pavia University, and a large dataset—are widely used in hyperspectral image compression research. Three of these datasets (Indian Pines, Cuprite, and Jasper Ridge) were captured by NASA's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, which gathers spectroradiometer data for surface characterization. The Pavia University dataset was collected using the ROSIS (Reflective Optics System Imaging Spectrometer), operated by the German Aerospace Centre. Detailed descriptions of these datasets, including their dimensions and acquisition methods, can be found in Chapter 3. The fifth dataset used in this chapter is the large dataset. The large dataset used in this experiment is substantial, with a size of 28.2 GB and dimensions of 4192 by 6708 pixels across 270 channels.

6.6.1 Architecture Search

Considering an image and our parameter budget for the multilayer perceptron (MLP), quantified in bits per pixel per band (bpppb), the primary objective is to determine the MLP architecture—specifically, the quantity of hidden layers and their respective widths—that can adequately represent the region of interest in a hyperspectral image while achieving an acceptable peak signal-to-noise ratio (PSNR).

The MLP architecture is selected through hyperparameter optimization, which entails training viable configurations with an appropriate number of hidden layers and suitable width on a specific hyperspectral image. The outcome of this procedure is a singular MLP capable of reconstructing the region of interest in the hyperspectral image with the specified PSNR value. The parameters of the final MLP are quantized to 16-bit precision, resulting in additional storage savings for representing the region of interest in the hyperspectral image.

6.6.2 Encoding Considerations

Our approach is categorized as a "slow-encoding-fast-decoding" compression technique. The approach requires training, specifically *overfitting*, several MLPs during the encoding (compression) phase. This is necessary to identify the MLP architecture that most accurately depicts the area of interest in the hyperspectral image, considering a specific storage constraint. Decoding, however, necessitates the evaluation of this MLP at many pixel places. The process of decoding is rapid. The "slow-encoding-fast-decoding" characteristic of this technology renders it especially appropriate for situations where the hyperspectral image is compressed solely at the time of capture. Encoding is inherently computationally intensive, a factor that must be considered when envisioning hyperspectral sensors that can compress hyperspectral images during capture, utilizing the suggested method. Our approach, as previously said, entails model fitting, which is fundamentally stochastic. Consequently, during the iterative phase, we retain the model parameters that have achieved the best PSNR value to date.

6.6.3 Model Fitting

All our models have two inputs, and the number of outputs corresponds to the number of channels (or bands) in the region of interest of the hyperspectral image. The activation function for hidden layers is sinusoidal. We initialized the MLP according to the instructions outlined in [Sitzmann et al., 2020b]. The Adam optimizer was employed during training, with a learning rate established at 2e-4. All studies were performed on a desktop equipped with an Intel i7 processor and an Nvidia RTX 2080 GPU.

6.6.4 Results-Region-aware Hyperspectral Image Compression

In our first study in this chapter, we explore region-aware image compression, where each dataset is divided into two regions, and different sampling rates are applied to each part (Figure 6.1). Specifically, in our experiments, we used a 5 percent sampling rate for the left side and a 90 percent sampling rate for the right side. As a result, we observed distinct PSNR values for each region, highlighting the impact of region-specific compression strategies (Table 6.1). This approach allows us to optimize compression based on the significance of different regions within an image, ensuring that more important areas are preserved with higher quality. The variations in PSNR demonstrate the effectiveness of targeted sampling in improving compression efficiency without sacrificing critical details. This technique lays the groundwork for our next project, which focuses on task-aware compression, where compression parameters are further tailored to specific image analysis tasks, enhancing overall perfor-

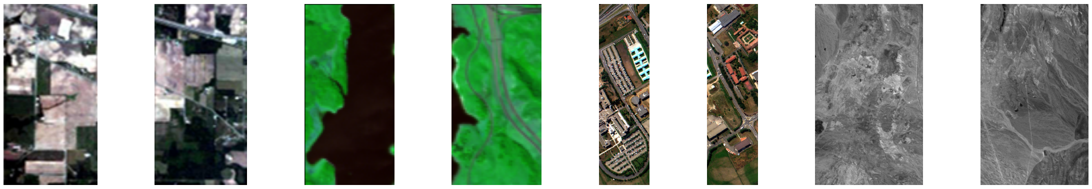


Figure 6.1: Region-aware image compression applied to four datasets—(from left to right) Indian Pines, Jasper Ridge, Pavia University, and Cuprite. Each dataset is divided into two regions with distinct sampling rates: 5 percent for the left side and 90 percent for the right side. The resulting PSNR values vary between the regions, demonstrating the effectiveness of applying region-specific compression strategies.

Dataset	Left Side PSNR	Right Side PSNR	PSNR
Indian Pines	33.82	33.15	33.47
Jasper Ridge	29.82	26.94	28.15
Pavia University	29.40	32.11	30.54
Cuprite	23.66	23.67	23.66

Table 6.1: PSNR comparison between the left and right regions of four datasets—Indian Pines, Jasper Ridge, Pavia University, and Cuprite—following region-aware image compression. The left side of each image is sampled at a lower rate (5 percent), while the right side is sampled at a higher rate (90 percent). The resulting PSNR values reflect the impact of differential sampling on image quality across the regions, demonstrating the balance achieved between compression efficiency and image fidelity.

mance.

6.6.4.1 Ablation Study on Region-Aware Compression Using the Pavia Dataset

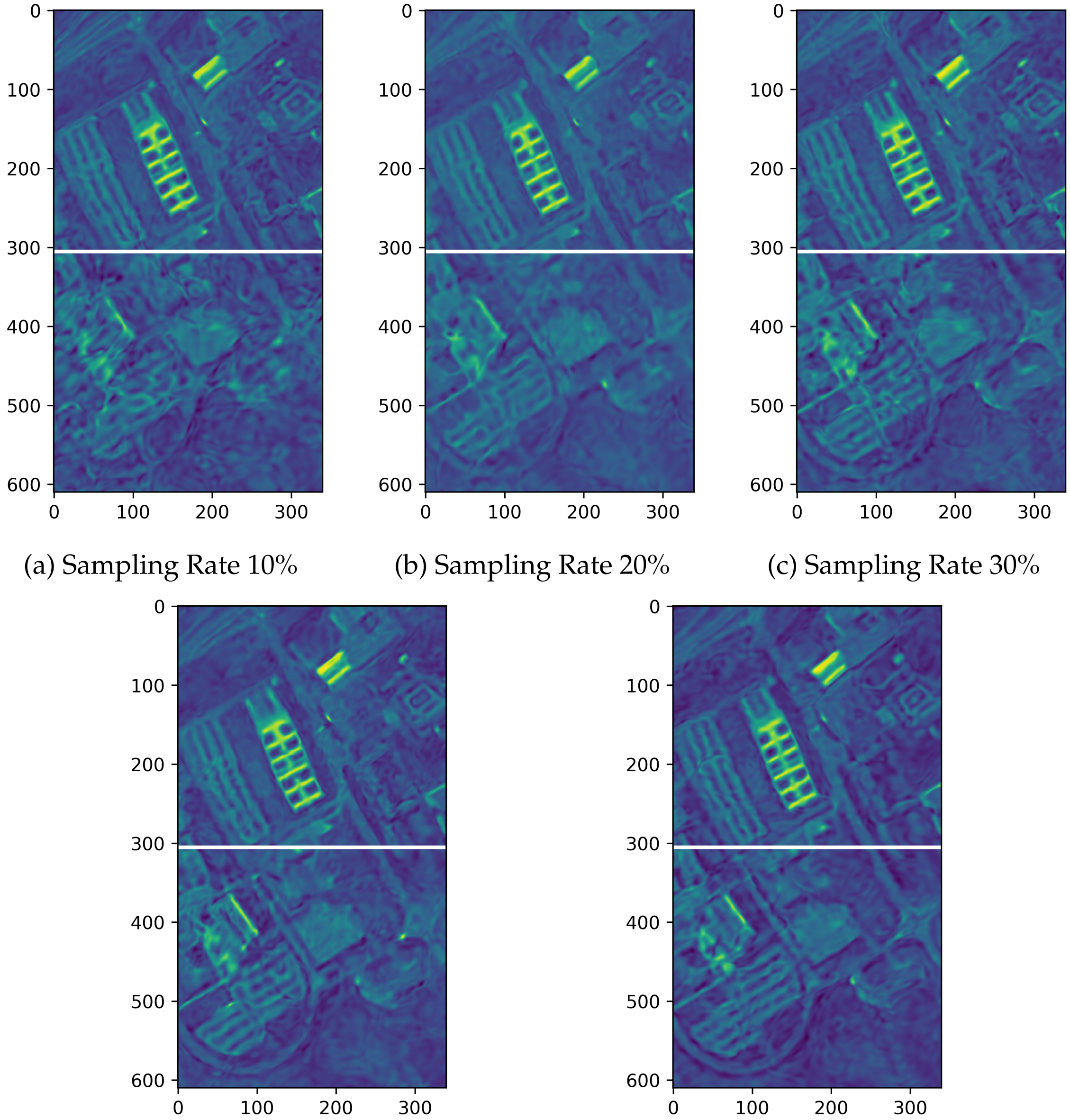
In the presented study on region-aware hyperspectral image compression, we extended our experiments to the Pavia dataset, where we applied distinct sampling rates to the image's two regions to observe the resulting PSNR values. This dataset was divided into two parts: the first region, referred to as Slice 1, was sampled consistently at 100 percent, while the second region, referred to as Slice 2, had its sampling rate varied between 10 percent and 100 percent. The primary goal of this ablation study was to assess how different levels of compression in the second region affected the overall image quality and to compare the preservation of critical details in each region.

As seen in Table 6.2, the PSNR values for Slice 1 remain relatively stable as expected due to the high sampling rate applied consistently. However, the PSNR values for Slice 2 show significant improvement as the sampling rate increases. At a 10 percent sampling rate, the PSNR for Slice 2 starts at 23.84 dB, whereas it rises to 26.21 dB at 100 percent. This suggests that as we allocate more resources to this region, the image quality in that specific region improves markedly. However, the relatively stable PSNR values for Slice 1 (around 26 dB) throughout the study demonstrate that preserving quality in one region does not significantly impact the other region's quality. This result reinforces the value of targeted compression techniques, particularly when balancing compression efficiency with image quality in hyperspectral imaging applications.

For further visual analysis, Figures 6.2 and 6.3 present reconstructed images of the Pavia dataset across different sampling rates. These images clearly illustrate the visual fidelity preserved across both regions, with noticeable improvements in detail preservation in the second region as its sampling rate increases. The visual and quantitative results of this study provide strong evidence that the region-aware compression technique can be effectively used to prioritize important regions within hyperspectral images without compromising overall quality.

6.6.5 Results-Task-aware Hyperspectral Image Compression using K-Means

In our second study in this chapter, we explore task-aware compression. Table 6.3 presents a detailed overview of the PSNR (Peak Signal-to-Noise Ratio) for the region of interest (ROI) across five hyperspectral image datasets, namely Indian Pines, Jasper Ridge, Pavia University, Cuprite, and the large dataset. The PSNR for the ROI highlights the effectiveness of task-aware compression in maintaining high fidelity for the most crucial parts of the image, where high-quality information is essential for specific tasks such as segmentation or classification.



(a) Sampling Rate 10%

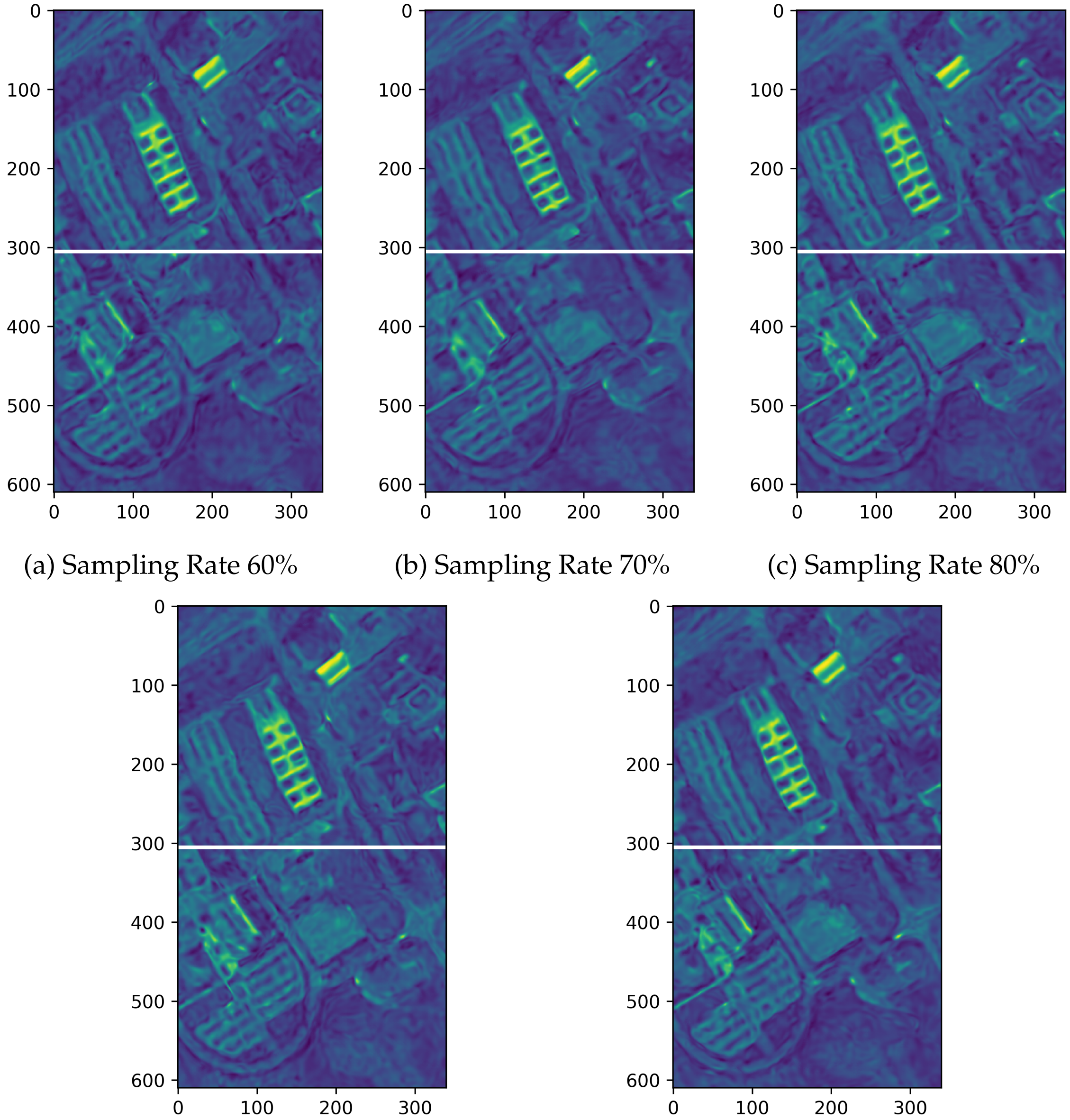
(b) Sampling Rate 20%

(c) Sampling Rate 30%

(d) Sampling Rate 40%

(e) Sampling Rate 50%

Figure 6.2: Reconstructed images of the Pavia dataset for sampling rates of 10%, 20%, 30%, 40%, and 50% in Slice 2. The images show the increasing quality of Slice 2 with higher sampling rates.



(a) Sampling Rate 60%

(b) Sampling Rate 70%

(c) Sampling Rate 80%

(d) Sampling Rate 90%

(e) Sampling Rate 100%

Figure 6.3: Reconstructed images of the Pavia dataset for sampling rates of 60%, 70%, 80%, 90%, and 100% in Slice 2. These images demonstrate the continued improvement in Slice 2 as the sampling rate approaches 100%.

Slice 2 Sampling Rate	Slice 1 PSNR	Slice 2 PSNR
10	26.72	23.84
20	26.60	24.72
30	26.51	25.22
40	26.49	25.54
50	26.23	25.69
60	26.17	25.79
70	26.25	25.95
80	26.06	26.12
90	26.06	26.15
100	25.99	26.21

Table 6.2: PSNR values for the Pavia dataset with varying sampling rates applied to Slice 2. Slice 1 was consistently sampled at 100 percent. The table shows the impact of differential sampling on the image quality of each region, illustrating the relationship between compression rate and PSNR.

Additionally, the table provides the PSNR for the entire image, which serves as an indicator of overall image quality post-compression. This metric is essential in assessing the trade-off between focusing on the ROI and preserving acceptable quality across the rest of the image. The bpppb (bits per pixel per band) quantifies the efficiency of the compression, reflecting how much data is retained per pixel, per spectral band after applying our method.

Moreover, the table includes the size of the compressed image in comparison to its original size, showcasing the significant reduction in storage requirements achieved through the task-aware approach. By strategically compressing different regions at varying levels, our method allows for a substantial decrease in the total size of the compressed image while preserving critical information quality in regions that are task-sensitive. This approach demonstrates the viability of our method for practical use in hyperspectral image compression, where both storage efficiency and task-specific image quality are important considerations.

Dataset	PSNR-ROI	PSNR	bpppb-ROI	bpppb	Compressed-size (KB)	Original-size (KB)
Indian Pines	42.28	25.94	0.072	0.28	332.6	9251
Jasper Ridge	38.21	13.62	1.54	0.59	333.5	4800
Pavia University	37.67	20.40	0.14	0.05	304	42724
Cuprite	37.54	22.03	0.08	0.01	333.5	140836
Large dataset	38.05	16.89	7.06	0.008	344.8	2.82e+7

Table 6.3: PSNR (Peak Signal-to-Noise Ratio) for the region of interest (ROI) and the entire image for five hyperspectral datasets. The table also includes the bits per pixel per band (bpppb), the size of the compressed image, and the original image size.

6.6.5.1 Visual Representation of Task-Aware Compression

For each of the five hyperspectral datasets—Indian Pines, Jasper Ridge, Pavia University, Cuprite, and the large dataset—three images are provided to visually illustrate the process and results of the task-aware compression (Figure 6.4). These images highlight both the original data and the impact of the compression method:

- Original Image: This image represents the uncompressed version of the hyperspectral dataset, showing the full spectral and spatial content before any compression is applied.

- Region of Interest (ROI): This image indicates the region within the hyperspectral data that is most critical for specific tasks, such as classification or segmentation. The highlighted region reflects where higher fidelity is maintained during the compression process.

- Reconstructed Image: This image shows the result of the task-aware compression and reconstruction. While the region of interest retains high visual quality and fidelity, the non-essential regions of the image are compressed more aggressively, demonstrating the trade-off between compression efficiency and image quality.

These images offer a clear visualization of how task-aware compression prioritizes regions of interest while optimizing overall data size, preserving essential information where it is most needed for task-specific analysis.

6.6.6 Results-Task-aware Hyperspectral Image Compression using U-Net

In our third study in this chapter, we explore task-aware compression using U-Net. We conducted experiments on five hyperspectral datasets—Pavia University, Jasper Ridge, Indian Pines, Cuprite, and the Large dataset. For each dataset, we performed segmentation, applied task-aware compression, and reconstructed the images. Figures 6.5 present the original images, segmented regions, and reconstructed images, allowing for a visual assessment of the compression effectiveness and fidelity preservation within the segmented ROIs.

The performance metrics, including PSNR for both the region of interest (ROI) and the full image, bits per pixel per band (bpppb) in the ROI and overall, and compressed and original image sizes, are summarized in Table 6.4. These metrics provide insights into the trade-offs between compression efficiency and quality preservation across different datasets.

Dataset	PSNR-ROI	PSNR	bpppb-ROI	bpppb	Compressed-size(KB)	Original-size(KB)
Indian Pines	35.07	25.46	0.66	0.28	332.6	9251
Jasper Ridge	26.43	13.79	2.11	0.59	333.5	4800
Pavia University	32.39	21.28	0.140	0.05	304.0	42724
Cuprite	26.21	23.75	0.040	0.018	333.5	140836
Large dataset	36.009	27.82	0.180	0.008	344.8	2.82e+7

Table 6.4: Performance metrics for task-aware compression on five hyperspectral datasets. The table includes PSNR values for the region of interest (ROI) and the full image, bits per pixel per band (bpppb) for both the ROI and full image, and compressed and original image sizes.

The results in Table 6.4 highlight the efficiency and quality of our task-aware compression approach across diverse datasets. Each dataset exhibits a significant increase in PSNR within the segmented region, confirming that our method effectively preserves critical spectral information in the regions of interest, which is essential for task-specific applications.

For example, the Indian Pines dataset achieved a PSNR of 35.07 for the ROI, substantially higher than the overall image PSNR of 25.46. This indicates that the method successfully applies more aggressive compression to the background while maintaining high fidelity within the segmented ROI. Similarly, the Pavia University dataset achieved a PSNR of 32.39 in the ROI, significantly exceeding the overall PSNR of 21.28, further validating the effectiveness of our selective compression strategy.

The large dataset stands out due to its high PSNR in the ROI (36.009) and full image PSNR of 27.82, demonstrating the method's ability to maintain spectral quality even for substantial data sizes. Additionally, the large dataset achieves an extremely low bpppb of 0.008 for the overall image, reflecting impressive data reduction without compromising essential information in the ROI.

The varying bpppb values for the ROIs and full images further illustrate the efficiency of this approach. The lower bpppb values for the full image compared to the segmented regions reflect the aggressive compression applied to the background, which significantly reduces data size while preserving necessary spectral details in the ROI. The large dataset's low bpppb value of 0.008 demonstrates the method's scalability and efficiency for high-volume data.

Overall, these results demonstrate that our task-aware compression method successfully balances compression efficiency and spectral fidelity, making it suitable for hyperspectral applications that prioritize specific regions of interest.

6.6.6.1 Conclusion

In this chapter, we demonstrated a task-aware hyperspectral image compression method that leverages segmentation to distinguish and prioritize critical regions within hyperspectral datasets. By using U-Net for semantic segmentation, we effectively identified ROIs in each dataset and applied differential compression to maintain high fidelity in task-relevant areas. The results indicate that this method achieves significant data reduction while preserving essential information in the ROI, making it highly suitable for applications where storage efficiency and task-specific data quality are crucial.

6.6.7 Limitations of Traditional Compression Methods

Conventional image compression methods, such as JPEG and JPEG2000, though widely used for compressing 2D images, are not suited for hyperspectral image data due to their fundamental design. JPEG and JPEG2000 are optimized for natural images (RGB) with three color channels, and they work by reducing spatial redundancies using techniques like Discrete Cosine Transform (JPEG) or wavelet-based approaches (JPEG2000). These methods are primarily spatial-based and lack the ability to handle the complex spectral correlation present in hyperspectral images, which contain hundreds of spectral bands [Babu et al., 2015].

Hyperspectral images are fundamentally different, as they consist of a high-dimensional spectral signature at each pixel, which requires specialized treatment of both spatial and spectral redundancies [Dua et al., 2020]. JPEG and JPEG2000 are not designed to prioritize certain regions of interest (ROI) based on task requirements, nor do they provide a framework for adapting compression rates in specific regions based on their importance for downstream tasks like classification or segmentation. As a result, these methods perform uniform compression across the entire image, which is suboptimal for task-aware scenarios.

6.6.7.1 Role of Implicit Neural Representations

In contrast, our proposed task-aware hyperspectral image compression method leverages implicit neural representations (INRs) to achieve a level of flexibility and adaptability that traditional methods cannot match. Implicit neural representations model the image as a continuous function, allowing for a more fine-grained, task-specific focus on regions of interest without relying on explicit pixel-based storage. By encoding the hyperspectral image as a neural network that learns to represent both spatial and spectral correlations, our method inherently supports task-aware compression.

Unlike traditional techniques, which treat all parts of the image uniformly, INRs allow us to allocate different levels of compression fidelity based on the task's specific needs. This is particularly critical for hyperspectral data, where certain regions of the image—those crucial for classification, material identification, or anomaly detection—require higher fidelity than others. The task-aware nature of the proposed method ensures that critical information in the region of interest is preserved with high precision, while less important regions can be compressed more aggressively, which is not feasible using methods like JPEG or JPEG2000.

6.6.7.2 Infeasibility for Traditional Methods

The inability of traditional methods like JPEG and JPEG2000 to handle task-aware hyperspectral compression is evident from several factors. First, they lack the flexibility to apply non-uniform compression rates across different regions of the image. Task-aware compression requires selectively compressing different parts of the image based on their importance for specific tasks, and this requires an adaptive mechanism that can identify and prioritize important regions. Neither JPEG nor JPEG2000 offers such an adaptive, task-driven framework, as their compression techniques are rigid and global.

Furthermore, hyperspectral images contain information across hundreds of bands, and traditional methods struggle to efficiently compress this high-dimensional data. JPEG2000, which provides some improvements over JPEG in handling more complex images, is still fundamentally designed for RGB images and lacks mechanisms to address the spectral redundancy present in hyperspectral data. As a result, the PSNR values achieved for hyperspectral data using JPEG2000 are significantly lower compared to those obtained with task-aware methods like ours that incorporate implicit neural representations.

The success of our method in preserving the quality of task-critical regions while achieving high compression ratios across non-essential areas demonstrates that traditional methods cannot match this level of performance. This makes implicit neural representations an essential tool for task-aware compression, as no existing traditional technique offers the same degree of flexibility and task-specific optimization.

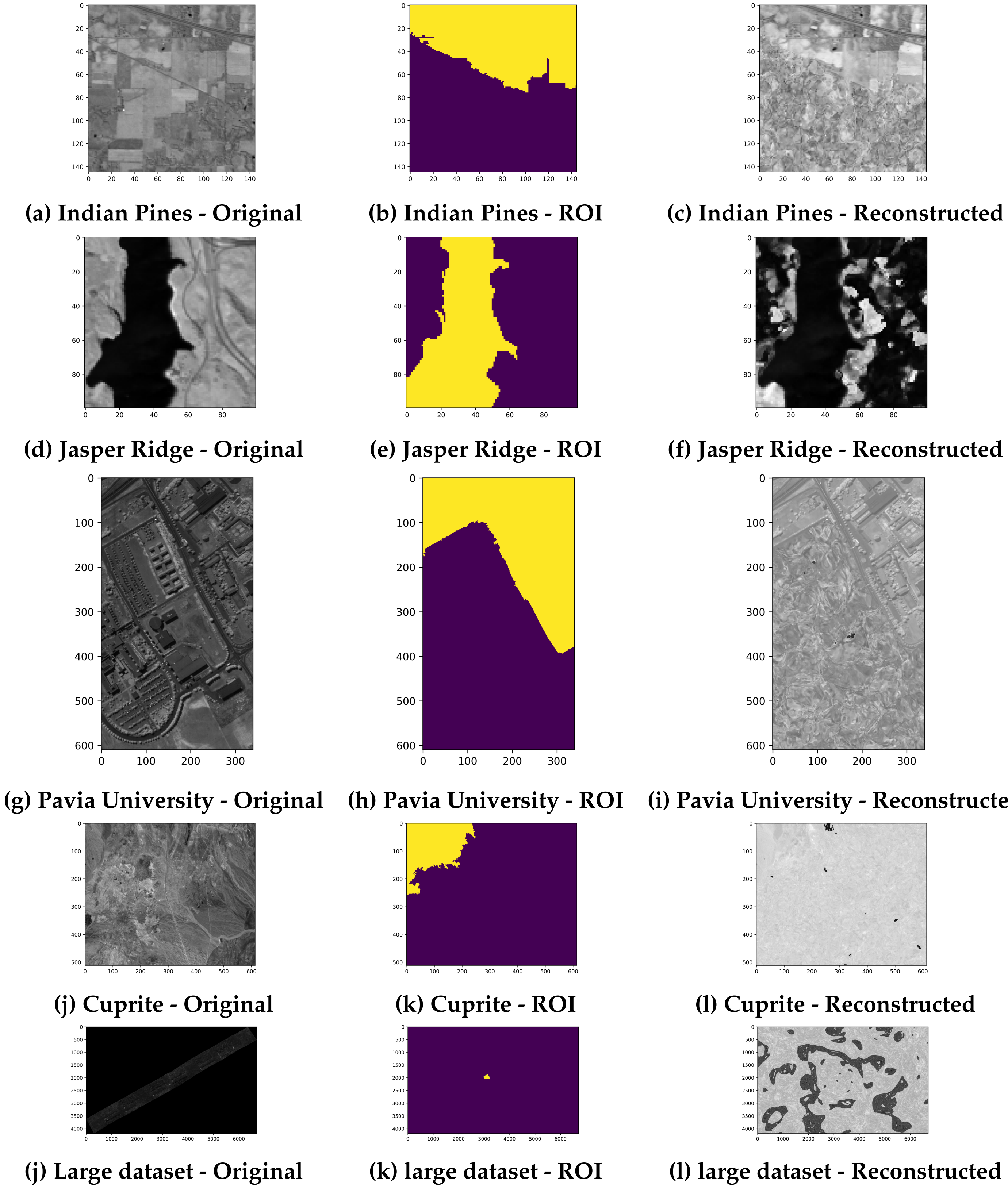
6.7 Conclusion

In this work, we proposed a task-aware hyperspectral image compression method designed to optimize the balance between data reduction and the preservation of task-critical regions. By leveraging the concept of region of interest (ROI), the method prioritizes compression in a way that maintains high fidelity in regions essential for specific tasks, such as classification or segmentation, while compressing non-essential regions more aggressively to achieve a substantial reduction in storage size.

Our experimental results, presented across five benchmark hyperspectral datasets—Indian Pines, Jasper Ridge, Pavia University Cuprite, and the large dataset—demonstrate the effectiveness of the proposed approach. The PSNR values in the region of interest confirm that the quality of the task-sensitive areas is preserved, while the PSNR for the entire image reflects a reasonable trade-off between global quality and compression efficiency. The bits per pixel per band (bpppb) and compressed image sizes show significant reductions in data volume, further supporting the practicality of the method for real-world applications where storage or transmission bandwidth is limited.

Moreover, the visual representations of the original images, region of interest, and reconstructed images illustrate how task-aware compression can maintain image integrity where it matters most, while allowing for aggressive compression in less critical regions. This method proves particularly useful for scenarios where accurate task performance is more important than preserving the quality of the entire image, such as in remote sensing, medical imaging, and hyperspectral analysis for object detection.

Overall, the results indicate that the task-aware approach is an effective solution for hyperspectral image compression, offering a balance between high compression ratios and the preservation of critical image information. This method holds great potential for applications requiring efficient storage and transmission without compromising the quality necessary for downstream tasks.



(a) Indian Pines - Original (b) Indian Pines - ROI (c) Indian Pines - Reconstructed

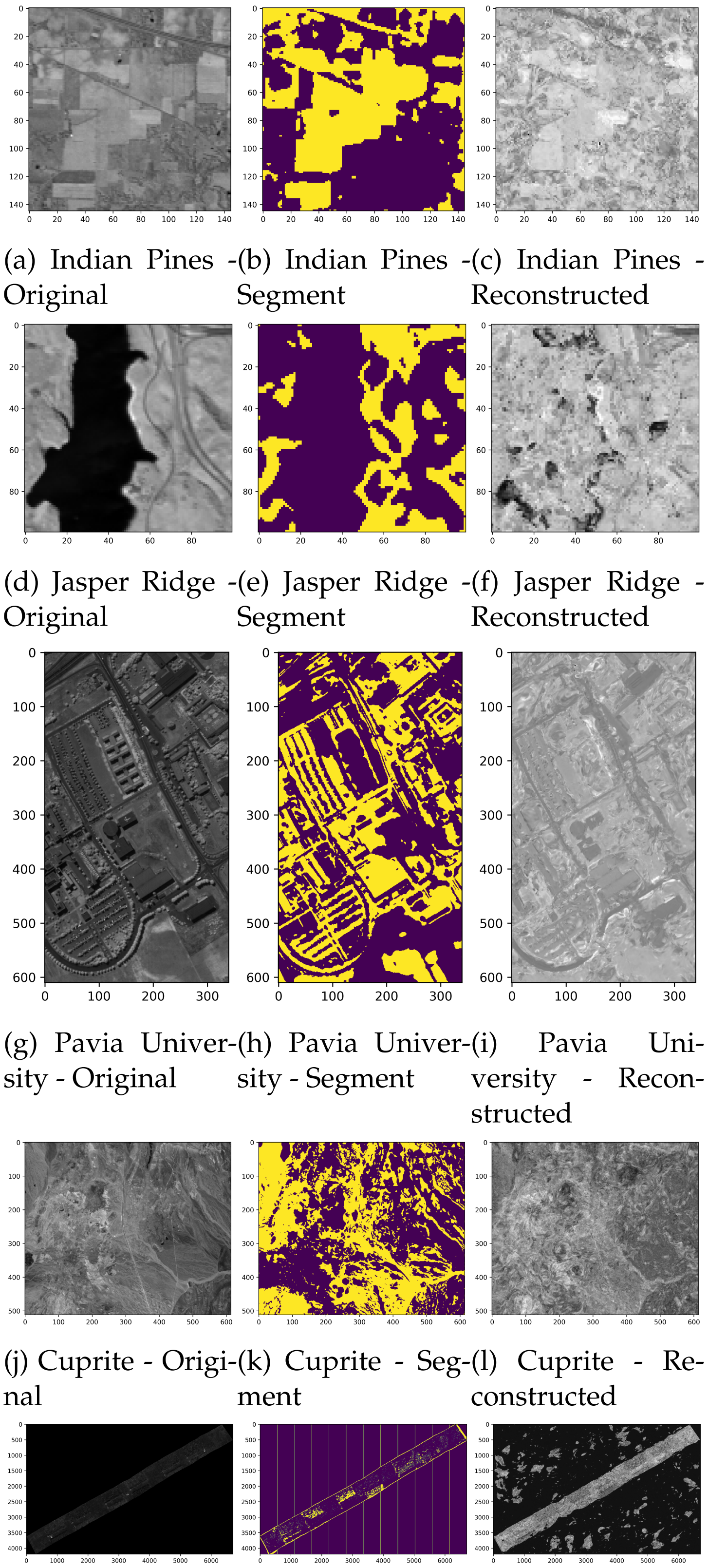
(d) Jasper Ridge - Original (e) Jasper Ridge - ROI (f) Jasper Ridge - Reconstructed

(g) Pavia University - Original (h) Pavia University - ROI (i) Pavia University - Reconstructe

(j) Cuprite - Original (k) Cuprite - ROI (l) Cuprite - Reconstructed

(j) Large dataset - Original (k) large dataset - ROI (l) large dataset - Reconstructed

Figure 6.4: Visual representation of task-aware compression for five hyperspectral datasets. Each row corresponds to a dataset (Indian Pines, Jasper Ridge, Pavia University, Cuprite, large dataset), showing the original image (left), the region of interest (middle), and the reconstructed image after compression (right). The region of interest is preserved with high fidelity in the reconstructed images, while non-essential regions are more aggressively compressed.



(a) Indian Pines - Original (b) Indian Pines - Segment (c) Indian Pines - Reconstructed

(d) Jasper Ridge - Original (e) Jasper Ridge - Segment (f) Jasper Ridge - Reconstructed

(g) Pavia University - Original (h) Pavia University - Segment (i) Pavia University - Reconstructed

(j) Cuprite - Original (k) Cuprite - Segment (l) Cuprite - Reconstructed

(m) Large dataset - Original (n) Large dataset - Segment (o) Large dataset - Reconstructed

Figure 6.5: Task-aware compression results for five hyperspectral datasets. Each row corresponds to a dataset (Indian Pines, Jasper Ridge, Pavia University, Cuprite, Large dataset) with columns representing the original image, the segmented region of interest, and the reconstructed image after task-aware compression.

Chapter 7

Conclusions

In this thesis, we focused on developing and exploring innovative methodologies to address the challenges of hyperspectral image compression. Hyperspectral data, with its high dimensionality and vast spectral bands, presents unique challenges that traditional compression techniques often struggle to manage efficiently. Recognizing the limitations of existing methods, we proposed new approaches that leverage advanced neural networks to handle both the complexity and scale of this data. Central to our work was the concept of implicit neural representations (INRs), which offer a compact and effective way of mapping spatial coordinates to spectral values, thereby reducing the storage burden while maintaining high image quality.

The use of INRs in this context provides a fundamentally different approach compared to conventional methods. Traditional algorithms treat each pixel's spectral values as independent hyperspectral images, often resulting in either loss of spectral fidelity or computational inefficiencies. By adopting INRs, we capture the underlying relationships between pixels and bands, encoding them in a neural function that can be stored efficiently. This novel approach allows us to represent hyperspectral images in a continuous space, ensuring that the compression is not only efficient but also flexible enough to accommodate complex data patterns.

Another significant focus of our research was achieving a delicate balance between compression efficiency and image fidelity. Hyperspectral images are often used in critical applications, such as environmental monitoring, remote sensing, and scientific research, where the accuracy of the spectral data is paramount. Any loss of quality could jeopardize subsequent analysis or decision-making processes. Therefore, it was essential to develop a methodology that minimized data loss during compression, preserving the critical spectral information necessary for accurate interpretation.

Through our experiments and evaluations, we demonstrated that the combination of INRs with meta-learning techniques provides a powerful framework for hyperspectral image compression. Even at high compression rates, our approach retains the integrity of the data, ensuring that the compressed images remain usable for analytical tasks. This research contributes to the field by showing that modern neural techniques can overcome the inherent limitations of traditional compression schemes, paving the way for more scalable and efficient solutions for managing hyperspectral data.

Throughout this thesis, several key contributions have been made to advance the field of hyperspectral image compression:

- Implicit Neural Representation (INR) for Compression: We introduced the concept of using implicit neural representations for hyperspectral image compression. This novel approach learns a function that maps spatial coordinates to spectral values, reducing data size while achieving competitive PSNR values. Our experiments demonstrated that INR-based compression offers superior performance over traditional methods, particularly at low bit rates.

- Sampling-based Compression Technique: To address the computational burden of encoding large hyperspectral images, we proposed a sampling strategy. This method applies differential sampling rates to different regions, speeding up the encoding process while maintaining high reconstruction quality. This sampling approach was evaluated across multiple datasets and showed promising results in reducing compression time without compromising accuracy.

- Meta-learning for Hyperspectral Compression: We developed a meta-learning-based approach to overcome the limitations of separate training for each image. By learning a base network with a shared structure, we enabled faster encoding through modulations applied to this network. This method significantly reduces encoding time and offers a practical solution for large-scale image datasets.
- Task-aware and Region-aware Compression: Our research introduced task-aware compression, prioritizing critical regions according to specific tasks like classification or segmentation. Additionally, we employed region-aware compression, applying higher sampling rates to more important areas. These approaches demonstrated their effectiveness in optimizing storage without sacrificing essential information, proving their applicability in fields such as remote sensing and environmental monitoring.
- Comprehensive Evaluation and Comparison: We thoroughly evaluated our methods using four benchmark datasets—Indian Pines, Jasper Ridge, Pavia University, and Cuprite—and a large-scale dataset. Our proposed techniques consistently outperformed traditional methods such as JPEG, JPEG2000, and PCA-DCT in terms of PSNR and SSIM metrics. Furthermore, we achieved superior compression ratios with reduced storage requirements.

In conclusion, this thesis provides a comprehensive exploration of cutting-edge methodologies for hyperspectral image compression, with a particular focus on leveraging implicit neural representations, meta-learning, and task-aware techniques. The approaches developed throughout this research not only address the challenges of compressing large-scale hyperspectral data but also offer solutions that balance efficiency, quality, and adaptability. Through our extensive evaluation across multiple datasets, we demonstrated that modern neural techniques could outperform traditional compression methods, setting a new benchmark for future research. The contributions presented here form a robust foundation for further innovation, with broad applicability across various domains such as remote sensing, environmental monitoring, and scientific research. Moving forward, these advancements open new doors for real-time data processing, ethical data management, and energy-efficient architectures, ensuring that hyperspectral imaging remains a vital tool in addressing the challenges of tomorrow.

Chapter 8

Limitations and Future work

While the methodologies proposed in this thesis have demonstrated significant advancements in hyperspectral image compression, there remain challenges and opportunities for further exploration. Hyperspectral imaging continues to evolve, with increasing demands for higher data quality, real-time processing, and domain-specific adaptations. As technology advances, it is essential to refine and extend the proposed approaches to meet the emerging requirements across various fields. This chapter outlines the key limitations of our current work and identifies promising directions for future research. We discuss the potential for expanding the meta-learning framework, the need for adaptive compression techniques, and the challenges in scaling our methods to real-time applications. Additionally, we address the ethical and computational considerations associated with large-scale compression and explore opportunities to apply these techniques in emerging fields. Each of the following sections delves into specific aspects of these limitations and future directions, providing a roadmap for ongoing research in this domain.

8.1 Extending the Meta-Learning Framework to Other Types of Imagery

While this thesis focused on hyperspectral image compression, the principles and methodologies developed, particularly the meta-learning framework, can be extended to other forms of imaging, such as multispectral or medical images. Multispectral data shares similarities with hyperspectral images, albeit with fewer spectral bands, and is widely used in fields like agriculture, remote sensing, and environmental monitoring. Applying meta-learning to multispectral data could further improve compression performance for datasets with different spectral characteristics. Similarly, medical imaging encompasses a variety of data types, including MRIs, CT scans, and ultrasounds, which require compression to efficiently manage storage and transmission. Extending our framework to medical imaging would not only improve storage but also ensure rapid access to large datasets, facilitating real-time diagnostic support. Future research could tailor the neural compression models to meet the specific needs of these domains, such as the need for lossless compression in sensitive medical applications.

8.2 Investigating Adaptive Compression Techniques

A promising area for future research lies in the development of adaptive compression techniques. Current methods, including the ones proposed in this work, apply fixed compression parameters across entire datasets or image regions. However, real-world images exhibit a wide range of complexities and characteristics. An adaptive approach could dynamically adjust compression parameters based on the content of each region or band within the image, optimizing the trade-off between compression ratio and quality. For example, less critical areas in an image could undergo higher compression, while key regions (such as objects of interest) could retain higher fidelity.

Research could also explore content-aware neural architectures that learn to prioritize different regions during the compression process, reducing redundant information while preserving crucial details for analytical tasks.

8.3 Enhancing Scalability for Real-Time Data Streams

One of the major limitations of the current work is the focus on static datasets. Scalability to real-time data streams remains a challenge, especially for applications like satellite-based remote sensing, autonomous vehicles, and real-time monitoring systems. These applications generate vast amounts of data continuously, requiring efficient methods for both compression and immediate processing. Future research could focus on developing incremental learning techniques that allow the model to update as new data becomes available, without the need to retrain from scratch. Moreover, implementing compression methods that can run on edge devices would enable real-time decision-making for autonomous systems, such as drones or self-driving vehicles, without relying heavily on cloud infrastructure.

8.4 Ethical and Social Implications of Hyperspectral Image Compression

As hyperspectral data becomes more widely used, ethical and social considerations surrounding its compression and usage must also be explored. When compression is applied to sensitive data, such as personal information in medical or biometric images, there is a potential risk of compromising privacy. Lossy compression, while efficient, could unintentionally alter critical details, raising questions about the reliability of decisions made using the compressed data. Future research must develop privacy-preserving compression frameworks that balance data reduction with security and privacy requirements. Additionally, regulatory and ethical guidelines may need to be established to govern the use of compressed hyperspectral data in various domains, ensuring that decisions made using this data are both fair and transparent.

8.5 Developing More Robust Compression Algorithms

The proposed methods, although effective, are still susceptible to challenges such as overfitting, compression artifacts, and performance degradation at extremely high compression ratios. Addressing these limitations will require the development of more robust neural architectures that can generalize across a wider variety of datasets and conditions. Future research could investigate hybrid compression techniques that combine traditional approaches like JPEG2000 with deep learning methods to achieve better performance. Additionally, ensemble learning techniques could be explored to improve model robustness, ensuring consistent results across datasets with varying characteristics.

8.6 Integrating the Model into Next-Generation Hyperspectral Sensors

An exciting direction for future research is the integration of neural compression models, such as the one proposed in this thesis, directly into the pipeline of next-generation hyperspectral sensors. Traditionally, hyperspectral sensors capture and store raw data onboard before transmission or offline processing. However, the increasing spatial and spectral resolution of modern sensors poses substantial storage and bandwidth challenges. Embedding lightweight, learnable compression modules within the sensor system—potentially at the hardware or firmware level—could enable on-the-fly compression, reducing data transmission costs and allowing real-time decision-making in the field.

The coordinate-based structure of Implicit Neural Representations (INRs) offers a natural fit for such integration. Since INRs map coordinates to spectral values with compact model parameters, they can be employed in low-memory environments, including edge computing platforms or field-deployable sensing units. Additionally, with a meta-learned base network that generalizes across various scenes, the model could adapt rapidly to different sensing contexts without requiring full retraining onboard.

To realize this potential, future work should focus on optimizing INR architectures for low-latency inference, reducing model size via pruning or quantization, and exploring hardware-friendly neural operations (e.g., replacing sinusoidal activations with approximations that are more efficient to compute on embedded platforms). Close collaboration with hardware designers and sensor manufacturers will be necessary to co-design systems where neural compression models are natively supported during data acquisition.

8.7 Extending the Framework to Region-Aware Compression

Another promising avenue for future research is the development of a region-aware compression framework based on the current model. In many real-world applications, not all parts of a hyperspectral image carry equal importance. For instance, regions containing objects of interest (e.g., vehicles, vegetation, or minerals) may require higher reconstruction fidelity, while background areas or homogeneous regions can tolerate more aggressive compression.

Extending the current approach to support region-aware or task-aware compression would involve dynamically adjusting the model's sampling rate, network capacity, or loss weighting based on semantic segmentation, object detection, or saliency analysis. For example, features from a pretrained detection model could guide the INR-based compression to allocate more representation capacity to regions deemed important for downstream tasks, such as classification or anomaly detection.

Moreover, the meta-learning framework lends itself naturally to region-aware adaptation. By conditioning the base network or modulation parameters on region-specific metadata or masks, the model can fine-tune its representation locally while maintaining global coherence. Future work could explore integrating attention mechanisms, spatial priors, or auxiliary task supervision (e.g., via segmentation labels) to guide this process.

Such a region-aware extension would improve the interpretability, utility, and efficiency of the compression model, especially in applications where computational resources are limited but accurate information retrieval in critical regions is essential. This could have a substantial impact on domains like precision agriculture, surveillance, and geological surveys, where selective fidelity is often more valuable than uniform quality across the scene.

8.8 Addressing Computational Efficiency and Energy Consumption

While our methods offer significant improvements in compression quality and speed, the computational demands of neural networks remain a bottleneck, particularly for large-scale datasets. Training and inference processes consume substantial amounts of energy, which may pose environmental and economic concerns, especially when deployed at scale. Future research should explore energy-efficient neural architectures and hardware optimizations, such as the use of specialized processors like GPUs and TPUs, to reduce the environmental impact of compression algorithms. Additionally, research into quantized neural networks or pruning techniques could offer ways to reduce the computational footprint without sacrificing performance, enabling faster and more efficient compression on constrained devices.

8.9 Exploring Compression for Emerging Applications

Finally, future research could focus on applying the proposed methodologies to emerging domains that require large-scale data management, such as urban planning, climate modeling, and space exploration. These fields generate enormous amounts of hyperspectral data, and efficient compression is essential to store and analyze this data effectively. Another promising direction could be the integration of hyperspectral compression techniques with generative models, such as diffusion models or generative adversarial networks (GANs), to reconstruct missing data and enhance the quality of compressed images further. This approach could unlock new possibilities in data synthesis, restoration, and augmentation, expanding the scope of hyperspectral image analysis.

8.10 Summary

In summary, while this thesis offers novel contributions to the field of hyperspectral image compression, there remain several exciting challenges and avenues for future exploration. Expanding the applicability of our methods to other imaging domains, developing adaptive and scalable compression techniques, and addressing ethical considerations are essential steps in advancing this research. As hyperspectral imaging continues to play a crucial role in various fields, including environmental monitoring, autonomous systems, and medical diagnostics, it is critical to develop robust, efficient, and responsible compression frameworks. Future work in this direction has the potential to not only optimize data management but also unlock new possibilities in real-time analysis and application-specific processing, ensuring that hyperspectral technologies remain relevant and impactful in the years to come.

Bibliography

[Adams, 1993] Adams, J. B. (1993). Imaging spectroscopy: Interpretation based on spectral mixture analysis. *Remote geochemical analysis: Elemental and mineralogical composition*, pages 145–166.

[Adão et al., 2017] Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., and Sousa, J. J. (2017). Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote sensing*, 9(11):1110.

[Ahanonu et al., 2019] Ahanonu, E., Marcellin, M. W., and Bilgin, A. (2019). Clustering regression wavelet analysis for lossless compression of hyperspectral imagery. In *DCC*, page 551.

[Aidini et al., 2019] Aidini, A., Giannopoulos, M., Pentari, A., Fotiadou, K., and Tsakalides, P. (2019). Hyperspectral image compression and super-resolution using tensor decomposition learning. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1369–1373. IEEE.

[Al-Hourani et al., 2023] Al-Hourani, A., Balendhran, S., Walia, S., and Hourani, T. (2023). Line scan hyperspectral imaging framework for open source low-cost platforms. *Remote Sensing*, 15(11):2787.

[Alemi et al., 2018] Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R., and Murphy, K. (2018). Fixing a broken elbo. In *Proc. 35th International Conference on Machine Learning*, volume 80, pages 159–168.

[Altamimi and Ben Youssef, 2021] Altamimi, A. and Ben Youssef, B. (2021). A systematic review of hardware-accelerated compression of remotely sensed hyperspectral images. *Sensors*, 22(1):263.

[Amrani et al., 2016] Amrani, N., Serra-Sagrista, J., Laparra, V., Marcellin, M. W., and Malo, J. (2016). Regression wavelet analysis for lossless coding of remote-sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(9):5616–5627.

[Babu et al., 2015] Babu, K. S., Ramachandran, V., Thyagharajan, K., and Santhosh, G. (2015). Hyperspectral image compression algorithms—a review. *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014, Volume 2*, pages 127–138.

[Bajpai et al., 2019a] Bajpai, S., Kidwai, N. R., and Singh, H. V. (2019a). 3d wavelet block tree coding for hyperspectral images. *International Journal of Innovative Technology and Exploring Engineering*, 8(6C):64–68.

[Bajpai et al., 2019b] Bajpai, S., Kidwai, N. R., Singh, H. V., and Singh, A. K. (2019b). Low memory block tree coding for hyperspectral images. *Multimedia Tools and Applications*, 78:27193–27209.

[Bajpai et al., 2022] Bajpai, S., Kidwai, N. R., Singh, H. V., and Singh, A. K. (2022). A low complexity hyperspectral image compression through 3d set partitioned embedded zero block coding. *Multimedia Tools and Applications*, pages 1–32.

[Bajpai et al., 2023] Bajpai, S., Sharma, D., Alam, M., Chandel, V. S., Pandey, A. K., and Tripathi, S. L. (2023). Curvelet transform based compression algorithm for low resource hyperspectral image sensors. *Journal of Electrical and Computer Engineering*, 2023(1):8961271.

[Balle' et al., 2017] Balle', J., Laparra, V., and Simoncelli, E. (2017). End-to-end optimized image compression. In *Proc. International Conference on Learning Representations (ICLR)*.

[Bansod et al., 2018] Bansod, B., Singh, R., and Thakur, R. (2018). Analysis of water quality parameters by hyperspectral imaging in ganges river. *Spatial Information Research*, 26:203–211.

[Bao et al., 2020] Bao, L., Cai, Y., Cui, Y., Liu, S., and Shi, L. (2020). Compression and rendering of space environment volume data based on improved hvq algorithm. In *Journal of Physics: Conference Series*, volume 1627, page 012025. IOP Publishing.

[Barrios et al., 2018] Barrios, Y., Sánchez, A., Santos, L., López, S., López, J. F., and Sarmiento, R. (2018). Hardware implementation of the ccsds 123.0-b-1 lossless multispectral and hyperspectral image compression standard by means of high level synthesis tools. In *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5. IEEE.

[Bascones et al., 2018] Bascones, D., Gonzalez, C., and Mozos, D. (2018). Hyperspectral image compression using vector quantization, pca and jpeg2000. *Remote sensing*, 10(6):907.

[Birajdar et al., 2019] Birajdar, A., Agarwal, H., Bolia, M., and Gupte, V. (2019). Image compression using run length encoding and its optimisation. In *2019 Global Conference for Advancement in Technology (GCAT)*, pages 1–6. IEEE.

[Boardman, 1995] Boardman, J. W. (1995). Analysis, understanding, and visualization of hyperspectral data as convex sets in n space. In *Imaging spectrometry*, volume 2480, pages 14–22. SPIE.

[Castrodad et al., 2011] Castrodad, A., Xing, Z., Greer, J. B., Bosch, E., Carin, L., and Sapiro, G. (2011). Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4263–4281.

[Chamain et al., 2022] Chamain, L. D., Qi, S., and Ding, Z. (2022). End-to-end image classification and compression with variational autoencoders. *IEEE Internet of Things Journal*, 9(21):21916–21931.

[Chan et al., 2021] Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., and Wetzstein, G. (2021). pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809.

[Charles et al., 2011] Charles, A. S., Olshausen, B. A., and Rozell, C. J. (2011). Learning sparse codes for hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):963–978.

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

[Cheng et al., 2014] Cheng, K.-J. et al. (2014). An improved ezw hyperspectral image compression. *Journal of Computer and Communications*, 2(02):31.

[Christophe et al., 2008] Christophe, E., Mailhes, C., and Duhamel, P. (2008). Hyperspectral image compression: adapting spiht and ezw to anisotropic 3-d wavelet coding. *IEEE Transactions on Image Processing*, 17(12):2334–2346.

[Crane et al., 2007] Crane, N. J., Bartick, E. G., Perlman, R. S., and Huffman, S. (2007). Infrared spectroscopic imaging for noninvasive detection of latent fingerprints. *Journal of forensic sciences*, 52(1):48–53.

[Cruz-Guerrero et al., 2022] Cruz-Guerrero, I. A., Leon, R., Granados-Castro, L., Fabelo, H., Ortega, S., Campos-Delgado, D. U., and Callico, G. M. (2022). Reflectance calibration with normalization correction in hyperspectral imaging. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pages 855–862. IEEE.

[Dodla et al., 2013] Dodla, S., Raju, Y. D. S., and Mohan, K. M. (2013). Image compression using wavelet and spiht encoding scheme. *International Journal of Engineering Trends and Technology (IJETT)–Volume*, 4.

[Du and Fowler, 2007] Du, Q. and Fowler, J. E. (2007). Hyperspectral image compression using jpeg2000 and principal component analysis. *IEEE Geoscience and Remote sensing letters*, 4(2):201–205.

[Dua et al., 2020] Dua, Y., Kumar, V., and Singh, R. S. (2020). Comprehensive review of hyperspectral image compression algorithms. *Optical Engineering*, 59(9):090902–090902.

[Dua et al., 2021] Dua, Y., Singh, R. S., Parwani, K., Lunagariya, S., and Kumar, V. (2021). Convolution neural network based lossy compression of hyperspectral images. *Signal Processing: Image Communication*, 95:116255.

[Duan et al., 2023] Duan, P., Kang, X., Ghamisi, P., and Li, S. (2023). Hyperspectral remote sensing benchmark database for oil spill detection with an isolation forest-guided unsupervised detector. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11.

[Dudhagara and Patel, 2017] Dudhagara, C. R. and Patel, M. M. (2017). A comparative study and analysis of ezw and spiht methods for wavelet based image compression. *Oriental journal of computer science and technology*, 10(3):669–673.

[Dupont et al., 2021] Dupont, E., Goliński, A., Alizadeh, M., Teh, Y. W., and Doucet, A. (2021). Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*.

[Dupont et al., 2022] Dupont, E., Loya, H., Alizadeh, M., Goliński, A., Teh, Y. W., and Doucet, A. (2022). Coin++: Data agnostic neural compression. *arXiv preprint arXiv:2201.12904*.

[Dusselaar and Paul, 2017] Dusselaar, R. and Paul, M. (2017). Hyperspectral image compression approaches: opportunities, challenges, and future directions: discussion. *JOSA A*, 34(12):2170–2180.

[Edelman et al., 2012] Edelman, G. J., Gaston, E., Van Leeuwen, T. G., Cullen, P., and Aalders, M. C. (2012). Hyperspectral imaging for non-contact analysis of forensic traces. *Forensic science international*, 223(1-3):28–39.

[Fan et al., 2021] Fan, G., Ma, Y., Mei, X., Fan, F., Huang, J., and Ma, J. (2021). Hyperspectral anomaly detection with robust graph autoencoders. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14.

[Fei, 2019] Fei, B. (2019). Hyperspectral imaging in medical applications. In *Data handling in science and technology*, volume 32, pages 523–565. Elsevier.

[Feng et al., 2019] Feng, F., Wang, S., Wang, C., and Zhang, J. (2019). Learning deep hierarchical spatial–spectral features for hyperspectral image classification based on residual 3d-2d cnn. *Sensors*, 19(23):5276.

[Feretb et al., 2019] Feretb, T. M., Mouquetg, P., Schmidth, F., and Dalla Murac, M. (2019). Applications in remote sensingdnatural landscapes. *Hyperspectral Imaging*, page 371.

[Finn et al., 2017] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

[Fjeldtvedt et al., 2018] Fjeldtvedt, J., Orlandić, M., and Johansen, T. A. (2018). An efficient real-time fpga implementation of the ccsds-123 compression standard for hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(10):3841–3852.

[Fowler and Rucker, 2007] Fowler, J. E. and Rucker, J. T. (2007). Three-dimensional wavelet-based compression of hyperspectral imagery. *Hyperspectral Data Exploitation: Theory and Applications*, pages 379–407.

[Fu et al., 2015] Fu, Y., Lam, A., Sato, I., and Sato, Y. (2015). Adaptive spatial-spectral dictionary learning for hyperspectral image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 343–351.

[Fuchs and Demir, 2023] Fuchs, M. H. P. and Demir, B. (2023). Hyspecnet-11k: A large-scale hyperspectral dataset for benchmarking learning-based hyperspectral image compression methods. In *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, pages 1779–1782. IEEE.

[Fuchs et al., 2024] Fuchs, M. H. P., Rasti, B., and Demir, B. (2024). Hycot: Hyperspectral compression transformer with an efficient training strategy. *arXiv preprint arXiv:2408.08700*.

[George and Manimekalai, 2014] George, R. and Manimekalai, M. (2014). A novel approach for image compression using zero tree coding. In *Proc. International Conference on Electronics and Communication Systems (ICECS)*, pages 1–5. IEEE.

[Giordano and Guccione, 2017] Giordano, R. and Guccione, P. (2017). Roi-based on-board compression for hyperspectral remote sensing images on gpu. *Sensors*, 17(5):1160.

[Goetz et al., 1985] Goetz, A. F., Vane, G., Solomon, J. E., and Rock, B. N. (1985). Imaging spectrometry for earth remote sensing. *science*, 228(4704):1147–1153.

[González-Conejero et al., 2009] González-Conejero, J., Bartrina-Rapesta, J., and Serra-Sagrista, J. (2009). Jpeg2000 encoding of remote sensing multispectral images with no-data regions. *IEEE Geoscience and Remote Sensing Letters*, 7(2):251–255.

[Good et al., 1994] Good, W. F., Maitz, G. S., and Gur, D. (1994). Joint photographic experts group (jpeg) compatible data compression of mammograms. *Journal of Digital Imaging*, 7(3):123–132.

[Gowtham et al., 2021] Gowtham, B. et al. (2021). Hyperspectral image analysis using principal component analysis and siamese network. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(7):1191–1198.

[Green et al., 1988] Green, A. A., Berman, M., Switzer, P., and Craig, M. D. (1988). A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Transactions on geoscience and remote sensing*, 26(1):65–74.

[Grewal et al., 2023] Grewal, R., Kasana, S. S., and Kasana, G. (2023). Hyperspectral image segmentation: a comprehensive survey. *Multimedia Tools and Applications*, 82(14):20819–20872.

[Guerra et al., 2018] Guerra, R., Barrios, Y., Díaz, M., Santos, L., López, S., and Sarmiento, R. (2018). A new algorithm for the on-board compression of hyperspectral images. *Remote Sensing*, 10(3):428.

[Guerri et al., 2024] Guerri, M. F., Distante, C., Spagnolo, P., Bougourzi, F., and Taleb-Ahmed, A. (2024). Deep learning techniques for hyperspectral image analysis in agriculture: A review. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, page 100062.

[Gunasheela and Prasantha, 2018] Gunasheela, K. and Prasantha, H. (2018). Compressive sensing approach to satellite hyperspectral image compression. In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 1*, pages 495–503. Springer.

[Guo et al., 2021] Guo, Y., Chong, Y., Ding, Y., Pan, S., and Gu, X. (2021). Learned hyperspectral compression using a student's t hyperprior. *Remote Sensing*, 13(21):4390.

[Guo et al., 2023] Guo, Y., Chong, Y., and Pan, S. (2023). Hyperspectral image compression via cross-channel contrastive learning. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–18.

[Guo et al., 2022] Guo, Y., Tao, Y., Chong, Y., Pan, S., and Liu, M. (2022). Edge-guided hyperspectral image compression with interactive dual attention. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–17.

[Hajebi, 2020] Hajebi, A. (2020). Unet for aerial image segmentation. Accessed: 2024-11-04.

[Hinton and Zemel, 1993] Hinton, G. E. and Zemel, R. (1993). Autoencoders, minimum description length and helmholtz free energy. *Advances in Neural Information Processing Systems*, 6.

[Hong et al., 2020] Hong, D., Gao, L., Yao, J., Zhang, B., Plaza, A., and Chanussot, J. (2020). Graph convolutional networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7):5966–5978.

[Huang et al., 2019] Huang, Q., Li, W., Hu, T., and Tao, R. (2019). Hyperspectral image super-resolution using generative adversarial network and residual learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3012–3016. IEEE.

[Iliopoulos et al., 2015] Iliopoulos, A.-S., Liu, T., and Sun, X. (2015). Hyperspectral image classification and clutter detection via multiple structural embeddings and dimension reductions. *arXiv preprint arXiv:1506.01115*.

[in the Loop, 2020] in the Loop, H. (2020). Semantic segmentation of aerial imagery. Accessed: 2024-11-04.

[Jaiswal and Sedamkar, 2014] Jaiswal, S. and Sedamkar, R. (2014). Performance evaluation on ezw & spih image compression techniques. *International Journal of scientific and Research publications*, 4(10).

[Jayaprakash et al., 2020] Jayaprakash, C., Damodaran, B. B., Viswanathan, S., and Soman, K. P. (2020). Randomized independent component analysis and linear discriminant analysis dimensionality reduction methods for hyperspectral image classification. *Journal of Applied Remote Sensing*, 14(3):036507–036507.

[Jiang et al., 2018] Jiang, Z., Pan, W. D., and Shen, H. (2018). Lstm based adaptive filtering for reduced prediction errors of hyperspectral images. In *2018 6th IEEE international conference on wireless for space and extreme environments (WISEE)*, pages 158–162. IEEE.

[Jifara et al., 2017] Jifara, W., Jiang, F., Zhang, B., Wang, H., Li, J., Grigorev, A., and Liu, S. (2017). Hyperspectral image compression based on online learning spectral features dictionary. *Multimedia Tools and Applications*, 76(23):25003–25014.

[Joseph, 1994] Joseph, W. (1994). Automated spectral analysis: A geologic example using aviris data, north grapevine mountains, nevada. In *Proc. Tenth Thematic Con-*

ference on Geologic Remote Sensing, Environmental Research Institute of Michigan, pages 1407–1418.

[Karaca and Güllü, 2019] Karaca, A. C. and Güllü, M. K. (2019). Superpixel based recursive least-squares method for lossless compression of hyperspectral images. *Multidimensional Systems and Signal Processing*, 30:903–919.

[Karami et al., 2010] Karami, A., Yazdi, M., and Asli, A. Z. (2010). Hyperspectral image compression based on tucker decomposition and discrete cosine transform. In *2010 2nd international conference on image processing theory, Tools and Applications*, pages 122–125. IEEE.

[Karayiannis and Pai, 1995] Karayiannis, N. B. and Pai, P.-I. (1995). Fuzzy vector quantization algorithms and their application in image compression. *IEEE Transactions on Image Processing*, 4(9):1193–1201.

[Katkovsky et al., 2018] Katkovsky, L. V., Martinov, A. O., Siliuk, V. A., Ivanov, D. A., and Kokhanovsky, A. A. (2018). Fast atmospheric correction method for hyperspectral data. *Remote Sensing*, 10(11):1698.

[Khan, 2022] Khan, A. (2022). *Automated Detection and Monitoring of Vegetation Through Deep Learning*. PhD thesis, Victoria University.

[Khayam, 2003] Khayam, S. A. (2003). The discrete cosine transform (dct): theory and application. *Michigan State University*, 114(1):31.

[Kopackova and Hladikova, 2014] Kopackova, V. and Hladikova, L. (2014). Applying spectral unmixing to determine surface water parameters in a mining environment. *Remote Sensing*, 6(11):11204–11224.

[Kozhemiakin et al., 2016] Kozhemiakin, R., Abramov, S., Lukin, V., Djurović, B., Djurović, I., and Vozel, B. (2016). Lossy compression of landsat multispectral images. In *2016 5th Mediterranean Conference on Embedded Computing (MECO)*, pages 104–107. IEEE.

[Kruse and Perry, 2009] Kruse, F. A. and Perry, S. L. (2009). Improving multispectral mapping by spectral modeling with hyperspectral signatures. *Journal of Applied Remote Sensing*, 3(1):033504.

[Kumar et al., 2018] Kumar, S., Chaudhuri, S., Banerjee, B., and Ali, F. (2018). Onboard hyperspectral image compression using compressed sensing and deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.

[Kussul et al., 2017] Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.

[Kwan and Larkin, 2019] Kwan, C. and Larkin, J. (2019). New results in perceptually lossless compression of hyperspectral images. *Journal of Signal and Information Processing*, 10(3):96–124.

[Kwan et al., 2019a] Kwan, C., Larkin, J., Budavari, B., and Chou, B. (2019a). Compression algorithm selection for multispectral mastcam images. *Signal & Image Processing: An International Journal (SIPIJ)*, 10(1).

[Kwan et al., 2019b] Kwan, C., Larkin, J., et al. (2019b). Perceptually lossless compression for mastcam multispectral images: A comparative study. *Journal of Signal and Information Processing*, 10(04):139.

[La Grassa et al., 2022] La Grassa, R., Re, C., Cremonese, G., and Gallo, I. (2022). Hyperspectral data compression using fully convolutional autoencoder. *Remote Sensing*, 14(10):2472.

[Le Gall, 1991] Le Gall, D. (1991). Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58.

[Li et al., 2022] Li, H., Cui, J., Zhang, X., Han, Y., and Cao, L. (2022). Dimensionality reduction and classification of hyperspectral remote sensing image feature extraction. *Remote Sensing*, 14(18):4579.

[Li and Liu, 2019] Li, J. and Liu, Z. (2019). Multispectral transforms using convolution neural networks for remote sensing multispectral image compression. *Remote Sensing*, 11(7):759.

[Li et al., 2023] Li, L., Liu, T., Wang, C., Qiu, M., Chen, C., Gao, M., and Zhou, A. (2023). Resizing codebook of vector quantization without retraining. *Multimedia Systems*, 29(3):1499–1512.

[Li et al., 2019] Li, R., Pan, Z., and Wang, Y. (2019). The linear prediction vector quantization for hyperspectral image compression. *Multimedia Tools and Applications*, 78:11701–11718.

[Li et al., 2021] Li, R., Pan, Z., Wang, Y., and Wang, P. (2021). The correlation-based tucker decomposition for hyperspectral image compression. *Neurocomputing*, 419:357–370.

[Li et al., 2016] Li, W., Fu, H., Yu, L., Gong, P., Feng, D., Li, C., and Clinton, N. (2016). Stacked autoencoder-based deep learning for remote-sensing image classification: a case study of african land-cover mapping. *International journal of remote sensing*, 37(23):5632–5646.

[Li et al., 2014] Li, X., Ren, J., Zhao, C., Qiao, T., and Marshall, S. (2014). Novel multivariate vector quantization for effective compression of hyperspectral imagery. *Optics Communications*, 332:192–200.

[Li et al., 2017] Li, Y., Zhang, H., and Shen, Q. (2017). Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sensing*, 9(1):67.

[Licciardi et al., 2014] Licciardi, G. A., Chanussot, J., Vasile, G., and Piscini, A. (2014). Enhancing hyperspectral image quality using nonlinear pca. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5087–5091. IEEE.

[Lin and Hwang, 2010] Lin, C.-C. and Hwang, Y.-T. (2010). An efficient lossless compression scheme for hyperspectral images using two-stage prediction. *IEEE Geoscience and Remote Sensing Letters*, 7(3):558–562.

[Lin and Hwang, 2011] Lin, C.-C. and Hwang, Y.-T. (2011). Lossless compression of hyperspectral images using adaptive prediction and backward search schemes. *J. Inf. Sci. Eng.*, 27(2):419–435.

[Liu et al., 2023] Liu, J., Sun, H., and Katto, J. (2023). Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14388–14397.

[Liu et al., 2017] Liu, Y., Pu, H., and Sun, D.-W. (2017). Hyperspectral imaging technique for evaluating food quality and safety during various processes: A review of recent applications. *Trends in food science & technology*, 69:25–35.

[Lu et al., 2020] Lu, B., Dao, P. D., Liu, J., He, Y., and Shang, J. (2020). Recent advances of hyperspectral imaging technology and applications in agriculture. *Remote Sensing*, 12(16):2659.

[Macqueen, 1967] Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

[Madani and McIsaac, 2021] Madani, H. and McIsaac, K. (2021). Distance transform-based spectral-spatial feature vector for hyperspectral image classification with stacked autoencoder. *Remote Sensing*, 13(9):1732.

[Mantripragada et al., 2022] Mantripragada, K., Dao, P. D., He, Y., and Qureshi, F. Z. (2022). The effects of spectral dimensionality reduction on hyperspectral pixel classification: A case study. *Plos one*, 17(7):e0269174.

[Mantripragada and Qureshi, 2024] Mantripragada, K. and Qureshi, F. Z. (2024). Hyperspectral pixel unmixing with latent dirichlet variational autoencoder. *IEEE Transactions on Geoscience and Remote Sensing*.

[Marmanis et al., 2015] Marmanis, D., Datcu, M., Esch, T., and Stilla, U. (2015). Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109.

[Mehta et al., 2021] Mehta, I., Gharbi, M., Barnes, C., Shechtman, E., Ramamoorthi, R., and Chandraker, M. (2021). Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223.

[Mei et al., 2018] Mei, S., Khan, B. M., Zhang, Y., and Du, Q. (2018). Low-complexity hyperspectral image compression using folded pca and jpeg2000. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 4756–4759. IEEE.

[Mishra et al., 2012] Mishra, M. K., Mishra, T. K., and Pani, A. K. (2012). Parallel lempel-ziv-welch (plzw) technique for data compression. *International Journal of Computer Science and Information Technologies*, 3(3):4038–4040.

[Moffat, 2019] Moffat, A. (2019). Huffman coding. *ACM Computing Surveys (CSUR)*, 52(4):1–35.

[Mohan et al., 2023] Mohan, D., Aravinth, J., and Rajendran, S. (2023). Reconstruction of compressed hyperspectral image using squeezenet coupled dense attentional net. *Remote Sensing*, 15(11):2734.

[Monica and Widipaminto, 2020] Monica, D. and Widipaminto, A. (2020). Fuzzy transform for high-resolution satellite images compression. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(2):1130–1136.

[Nagendran et al., 2024] Nagendran, R., Ramadass, S., Thilagavathi, K., and Ravuri, A. (2024). Lossless hyperspectral image compression by combining the spectral decorrelation techniques with transform coding methods. *International Journal of Remote Sensing*, 45(18):6226–6248.

[Nalepa et al., 2019] Nalepa, J., Myller, M., and Kawulok, M. (2019). Hyperspectral data augmentation. *arXiv preprint arXiv:1903.05580*.

[Ngadiran et al., 2010] Ngadiran, R., Boussakta, S., Sharif, B., and Bouridane, A. (2010). Efficient implementation of 3d listless speck. In *International Conference on Computer and Communication Engineering (ICCCE'10)*, pages 1–4. IEEE.

[Nhaila et al., 2014] Nhaila, H., Sarhrouni, E., and Hammouch, A. (2014). A survey on fundamental concepts and practical challenges of hyperspectral images. In *2014 Second World Conference on Complex Systems (WCCS)*, pages 659–664. IEEE.

[Nian et al., 2016] Nian, Y., Liu, Y., and Ye, Z. (2016). Pairwise klt-based compression for multispectral images. *Sensing and Imaging*, 17(1):1–15.

[Ouahioune et al., 2021] Ouahioune, M., Ameur, S., and Lahdir, M. (2021). Enhancing hyperspectral image compression using learning-based super-resolution technique. *Earth Science Informatics*, 14(3):1173–1183.

[Pan et al., 2011] Pan, X., Liu, R., and Lv, X. (2011). Low-complexity compression method for hyperspectral images based on distributed source coding. *IEEE Geoscience and Remote Sensing Letters*, 9(2):224–227.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

[Paul et al., 2016] Paul, M., Xiao, R., Gao, J., and Bossomaier, T. (2016). Reflectance prediction modelling for residual-based hyperspectral image coding. *PloS one*, 11(10):e0161212.

[Perez et al., 2018] Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Qian et al., 2012] Qian, Y., Ye, M., and Zhou, J. (2012). Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4):2276–2291.

[Qiao et al., 2014] Qiao, T., Ren, J., Sun, M., Zheng, J., and Marshall, S. (2014). Effective compression of hyperspectral imagery using an improved 3d dct approach for land-cover analysis in remote-sensing applications. *International Journal of Remote Sensing*, 35(20):7316–7337.

[Rainey et al., 2003] Rainey, M. P., Tyler, A., Gilvear, D., Bryant, R. G., and McDonald, P. (2003). Mapping intertidal estuarine sediment grain size distributions through airborne remote sensing. *Remote Sensing of Environment*, 86(4):480–490.

[Raja and Suruliandi, 2011] Raja, S. and Suruliandi, A. (2011). Image compression using wdr & aswdr techniques with different wavelet codecs. *ACEEE Int. J. Inform. Technol*, 1:23–26.

[Ramamurthy et al., 2020] Ramamurthy, M., Robinson, Y. H., Vimal, S., and Suresh, A. (2020). Auto encoder based dimensionality reduction and classification using convolutional neural networks for hyperspectral images. *Microprocessors and Microsystems*, 79:103280.

[Rasti et al., 2012] Rasti, B., Sveinsson, J. R., Ulfarsson, M. O., and Benediktsson, J. A. (2012). Hyperspectral image denoising using 3d wavelets. In *Proc. IEEE International Geoscience and Remote Sensing Symposium*, pages 1349–1352. IEEE.

[Redmon, 2016] Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[Ren et al., 2022] Ren, Z., Zhai, Q., and Sun, L. (2022). A novel method for hyperspectral mineral mapping based on clustering-matching and nonnegative matrix factorization. *Remote Sensing*, 14(4):1042.

[Rezasoltani and Qureshi, 2023a] Rezasoltani, S. and Qureshi, F. Z. (2023a). Hyperspectral image compression using implicit neural representations. In *2023 20th Conference on Robots and Vision (CRV)*, pages 248–255.

[Rezasoltani and Qureshi, 2023b] Rezasoltani, S. and Qureshi, F. Z. (2023b). Hyperspectral image compression using sampling and implicit neural representations. *arXiv preprint arXiv:2312.01558*.

[Rizkinia and Okuda, 2017] Rizkinia, M. and Okuda, M. (2017). Joint local abundance sparse unmixing for hyperspectral images. *Remote Sensing*, 9(12):1224.

[Romero et al., 2015] Romero, A., Gatta, C., and Camps-Valls, G. (2015). Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362.

[Sellami and Tabbone, 2022] Sellami, A. and Tabbone, S. (2022). Deep neural networks-based relevant latent representation learning for hyperspectral image classification. *Pattern Recognition*, 121:108224.

[Shapiro, 1993a] Shapiro, J. M. (1993a). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on signal processing*, 41(12):3445–3462.

[Shapiro, 1993b] Shapiro, J. M. (1993b). Smart compression using the embedded zerotree wavelet (ezw) algorithm. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 486–490. IEEE.

[Shen et al., 2018] Shen, H., Jiang, Z., and Pan, W. D. (2018). Efficient lossless compression of multitemporal hyperspectral image data. *Journal of Imaging*, 4(12):142.

[Shen et al., 2016] Shen, H., Pan, W. D., and Wu, D. (2016). Predictive lossless compression of regions of interest in hyperspectral images with no-data regions. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):173–182.

[Shingate et al., 2010] Shingate, V., Sontakke, T., and Talbar, S. (2010). Still image compression using embedded zerotree wavelet encoding. *International Journal of Computer Science & Communication*, 1(1):21–24.

[Singh et al., 2017] Singh, G., Korange, P., and Verma, D. (2017). Analysis of ezw and spiht algorithms for compression of an image. *International Journal on Emerging Technologies*, pages 671–675.

[Sitzmann et al., 2020a] Sitzmann, V., Chan, E., Tucker, R., Snavely, N., and Wetzstein, G. (2020a). Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147.

[Sitzmann et al., 2020b] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020b). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473.

[Skodras et al., 2001] Skodras, A., Christopoulos, C., and Ebrahimi, T. (2001). The jpeg 2000 still image compression standard. *IEEE Signal processing magazine*, 18(5):36–58.

[Sommer et al., 1998] Sommer, S., Hill, J., and Megier, J. (1998). The potential of remote sensing for monitoring rural land use changes and their effects on soil conditions. *Agriculture, ecosystems & environment*, 67(2-3):197–209.

[SreeVidya et al., 2016] SreeVidya, B., Kishore, M., and Guruprasad, H. (2016). Wavelet transform analysis on image compression using spiht. *International Journal of Computer Applications*, 146(8):1–7.

[Strümpler et al., 2022] Strümpler, Y., Postels, J., Yang, R., Gool, L. V., and Tombari, F. (2022). Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer.

[Sudha and Sudhakar, 2013] Sudha, V. and Sudhakar, R. (2013). 3d listless embedded block coding algorithm for compression of volumetric medical images.

[Sullivan et al., 2012] Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668.

[Tancik et al., 2020] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547.

[Tang and Pearlman, 2006] Tang, X. and Pearlman, W. A. (2006). Three-dimensional wavelet-based compression of hyperspectral images. In *Hyperspectral data compression*, pages 273–308. Springer.

[Toderici et al., 2017] Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., and Covell, M. (2017). Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5306–5314.

[Töreyın et al., 2015] Töreyın, B. U., Yilmaz, O., Mert, Y. M., and Türk, F. (2015). Lossless hyperspectral image compression using wavelet transform based spectral decorrelation. In *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*, pages 251–254. IEEE.

[Tsai et al., 2007] Tsai, F., Lin, E.-K., and Yoshino, K. (2007). Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species. *International Journal of Remote Sensing*, 28(5):1023–1039.

[Uddin et al., 2021] Uddin, M. P., Mamun, M. A., and Hossain, M. A. (2021). Pca-based feature reduction for hyperspectral remote sensing image classification. *IETE Technical Review*, 38(4):377–396.

[UlkU and Kizgut, 2018] UlkU, I. and Kizgut, E. (2018). Large-scale hyperspectral image compression via sparse representations based on online learning. *International Journal of Applied Mathematics and Computer Science*, 28(1):197–207.

[UlkU and TOreyin, 2014] UlkU, I. and TOreyin, B. U. (2014). Lossy compression of hyperspectral images using online learning based sparse coding. In *2014 Interna-*

tional Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), pages 1–5. IEEE.

[UlkU and Toreyin, 2015] UlkU, I. and Toreyin, B. U. (2015). Sparse coding of hyperspectral imagery using online learning. *Signal, Image and Video Processing*, 9(4):959–966.

[Vali et al., 2024] Vali, A., Comai, S., and Matteucci, M. (2024). An automated machine learning framework for adaptive and optimized hyperspectral-based land cover and land-use segmentation. *Remote Sensing*, 16(14):2561.

[Valsesia and Magli, 2019] Valsesia, D. and Magli, E. (2019). High-throughput onboard hyperspectral image compression with ground-based cnn reconstruction. *IEEE transactions on geoscience and remote sensing*, 57(12):9544–9553.

[Verdú et al., 2022] Verdú, S., Serra-Sagristà, J., Bartrina-Rapesta, J., Hernández-Cabronero, M., Laparra, V., and Ballé, J. (2022). Hyperspectral image compression using convolutional neural networks with local spectral transforms and non-uniform sample normalisation. In *Proc. 8th Int. Workshop Onboard Payload Data Compress*, pages 1–8.

[Vishnu et al., 2013] Vishnu, S., Nidamanuri, R. R., and Bremananth, R. (2013). Spectral material mapping using hyperspectral imagery: a review of spectral matching and library search methods. *Geocarto international*, 28(2):171–190.

[Walker and Nguyen, 2001] Walker, J. S. and Nguyen, T. Q. (2001). Wavelet-based image compression. *Sub-chapter of CRC Press book: Transforms and Data Compression*, pages 267–312.

[Wan et al., 2021] Wan, S., Yeh, M.-L., and Ma, H.-L. (2021). An innovative intelligent system with integrated cnn and svm: Considering various crops through hyperspectral image data. *ISPRS International Journal of Geo-Information*, 10(4):242.

[Wang et al., 2004] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

[Watanabe et al., 2019] Watanabe, J.-I., Shao, Y., and Miura, N. (2019). Underwater and airborne monitoring of marine ecosystems and debris. *Journal of Applied Remote Sensing*, 13(4):044509.

[Welikanna, 2008] Welikanna, D. (2008). Analysis of the effectiveness of spectral mixture analysis and markov random field based super resolution mapping in the context of urban composition. ITC.

[Wolfe and Black, 2018] Wolfe, J. D. and Black, S. R. (2018). Hyperspectral analytics in envi target detection and spectral mapping methods. *Harris Corporation, September*, 19:2018.

[Xue et al., 2019] Xue, J., Zhao, Y., Liao, W., and Chan, J. C.-W. (2019). Nonlocal tensor sparse representation and low-rank regularization for hyperspectral image compressive sensing reconstruction. *Remote Sensing*, 11(2):193.

[Yadav and Nagmode, 2018] Yadav, R. J. and Nagmode, M. (2018). Compression of hyperspectral image using pca–dct technology. In *Innovations in Electronics and Communication Engineering: Proceedings of the Fifth ICIECE 2016*, pages 269–277. Springer.

[Yang and Wang, 2018] Yang, B. and Wang, B. (2018). Band-wise nonlinear unmixing for hyperspectral imagery using an extended multilinear mixing model. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11):6747–6762.

[Ye et al., 2023] Ye, J., Yeo, H., Park, J., and Han, D. (2023). Accelir: Task-aware image compression for accelerating neural restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18216–18226.

[Yousefi et al., 2018] Yousefi, B., Sojasi, S., Castanedo, C. I., Maldague, X. P., Beaudoin, G., and Chamberland, M. (2018). Comparison assessment of low rank sparse-pca based-clustering/classification for automatic mineral identification in long wave infrared hyperspectral imagery. *Infrared Physics & Technology*, 93:103–111.

[Yu, 2024] Yu, Y. (2024). Unmixingsr: Material-aware network with unsupervised unmixing as auxiliary task for hyperspectral image super-resolution. *arXiv preprint arXiv:2407.06525*.

[Zala and Parmar, 2013] Zala, M. and Parmar, S. (2013). 3d wavelet transform with spiht algorithm for image compression. *Int. J. Appl. Innov. Eng. Manage.*, 2:384–392.

[Zhang et al., 2018] Zhang, J., Chen, L., Zhuo, L., Liang, X., and Li, J. (2018). An efficient hyperspectral image retrieval method: Deep spectral-spatial feature extraction with dcgan and dimensionality reduction using t-sne-based nm hashing. *Remote Sensing*, 10(2):271.

[Zhang et al., 2016] Zhang, S., Li, J., Liu, K., Deng, C., Liu, L., and Plaza, A. (2016). Hyperspectral unmixing based on local collaborative sparse regression. *IEEE Geoscience and Remote Sensing Letters*, 13(5):631–635.

[Zhao et al., 2012] Zhao, C., Wang, Y., and Mei, F. (2012). Kernel ica feature extraction for anomaly detection in hyperspectral imagery. *Chinese journal of electronics*, 21(2):265–269.

[Zhao et al., 2020] Zhao, H., Deng, K., Li, N., Wang, Z., and Wei, W. (2020). Hierarchical spatial-spectral feature extraction with long short term memory (lstm) for mineral identification using hyperspectral imagery. *Sensors*, 20(23):6854.

[Zhao and Du, 2016] Zhao, W. and Du, S. (2016). Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4544–4554.

[Zheng et al., 2022] Zheng, T., Dai, Y., Xue, C., and Zhou, L. (2022). Recursive least squares for near-lossless hyperspectral data compression. *Applied Sciences*, 12(14):7172.

[Zikiou et al., 2020] Zikiou, N., Lahdir, M., and Helbert, D. (2020). Support vector regression-based 3d-wavelet texture learning for hyperspectral image compression. *The Visual Computer*, 36(7):1473–1490.

[Zintgraf et al., 2019] Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. (2019). Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR.