Meta-Learned Implicit Neural Representations for Scalable and Fast Hyperspectral Image Compression

Shima Rezasoltani[1] [0000–0002–4554–5800] and Faisal Z. Qureshi[1] [0000–0002–8992–3607]

Faculty of Science, University of Ontario Institute of Technology, Oshawa, ON L1G 0C5, Canada
shima.rezasoltani@ontariotechu.ca, faisal.qureshi@ontariotechu.ca

Abstract. Hyperspectral image (HSI) data contains high-dimensional spectral information across hundreds of channels per pixel, presenting substantial challenges for storage and transmission. In this work, we propose a meta-learning-based compression framework that transforms hyperspectral data into implicit neural representations. Unlike traditional INR-based compression schemes, which require per-image training and result in slow encoding, our method leverages a shared meta-learned base network and learns image-specific modulations through a lightweight latent code. These modulations serve as compact descriptors that enable rapid compression while preserving reconstruction quality. We extend previous work by introducing comparative evaluations against video-based compression schemes, incorporating results on a large-scale HSI dataset exceeding 28 GB, and analyzing the method's scalability. Experimental results across four standard benchmarks and a high-resolution dataset demonstrate that our method achieves significantly faster compression times and superior compression rates, while maintaining competitive PSNR at extremely low bit-per-pixel-per-band (bpppb) levels.

Keywords: Hyperspectral Image Compression · Implicit Neural Representations · Meta Learning · Large-scale Data Compression

1 INTRODUCTION

Hyperspectral imaging captures a wide range of electromagnetic spectrum values at each pixel location, in contrast to traditional grayscale or RGB images, which encode only one or three intensity values per pixel [16]. Each pixel in a hyperspectral image contains dozens or even hundreds of spectral measurements, representing the light reflectance across various frequency bands. This rich spectral information enhances the ability to recognize objects, identify materials, and interpret scenes with greater precision than is possible using standard RGB imagery. As the cost of acquiring high-resolution hyperspectral data continues to decrease, such images are being applied in an expanding array of disciplines—including remote sensing, agriculture, biotechnology, medical diagnostics, pharmaceutical manufacturing, food safety, and resource exploration in mining and the oil and gas industries [26, 4, 1, 19, 42, 33, 13, 18, 14, 7]. However, the volume of data produced by hyperspectral sensors is substantially larger than that of RGB images of comparable spatial resolution, leading to challenges in storage,

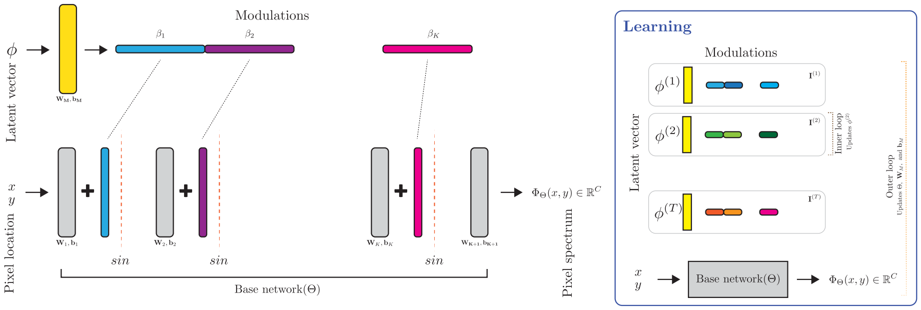


Fig. 1: The base network captures the shared structure between multiple hyperspectral images; whereas, the modulations (or latent vector) store image-specific information. Meta learning is used to learn both the shared parameters (Θ, $\mathbf{W}_\mathrm{M}$, and $\mathbf{b}_\mathrm{M}$) and the image-specific latent vectors φ. Once an image is compressed, it is sufficient to store the latent vector associated with this image [38].

processing, and transmission. These challenges highlight the critical need for efficient compression methods. In this study, we focus on hyperspectral image compression as a means to facilitate the practical use of large-scale hyperspectral datasets.

We introduce a novel compression framework for hyperspectral images that encodes each image through modulation vectors applied to a shared base network.[1] This architecture is influenced by the data-agnostic compression model proposed in [11], which we extend and adapt for the specific challenges posed by hyperspectral data. Our method departs from traditional implicit neural representation (INR) techniques by reusing a globally trained base network across multiple images. This enables the model to leverage recurring spatial and spectral patterns commonly found across hyperspectral scenes, thereby significantly reducing the time required for compression. Additionally, the proposed modulations—compact, image-specific adjustments—are far more storage-efficient than saving full neural network weights for each image. While the base network still needs to be retained, its size is shared across the dataset, making it a negligible cost in practice. The central idea is to factor out and encode the shared structure among images within the base network, while delegating image-specific content to lightweight modulation codes. In doing so, our approach improves upon earlier INR-based methods by achieving meaningful gains in both encoding speed and compression ratio [40].

To assess the effectiveness of our approach, we conduct extensive experiments on four widely used hyperspectral datasets: Indian Pines, Jasper Ridge, Pavia University, and Cuprite. The evaluation demonstrates that our method delivers substantially lower compression times than many existing techniques operating at comparable compression levels. In addition to its speed, the proposed framework achieves reconstruction qual-

[1] An initial version of this work appeared in the proceedings of ICPRAM 2025 [41]. Please see the attached *changes* document to see the list of differences between the ICPRAM 2025 conference paper and this article.

ity—measured by Peak Signal-to-Noise Ratio (PSNR)—that is on par with or better than that of established compression schemes.

The remainder of this paper is structured as follows. Section 2 reviews relevant work in hyperspectral image compression and neural representation learning. Section 3 outlines our proposed compression framework and describes the evaluation metrics used. In Section 4, we present the datasets, experimental setup, and results. Section 5 discusses the large dataset and the results related to compressing that dataset. Finally, Section 6 provides a summary of our contributions and discusses future research directions.

2 RELATED WORK

Hyperspectral image compression has been the focus of extensive research in recent years. Due to space limitations, this section focuses specifically on learning-based approaches, which have gained significant attention for their ability to model complex spectral and spatial dependencies. While this overview is not exhaustive, we refer interested readers to comprehensive surveys such as [50, 10], which catalog a broad range of compression techniques developed for hyperspectral imagery.

Model-driven learning-based approaches to hyperspectral image compression aim to optimize both bitrate and reconstruction fidelity through training. However, a common drawback of such methods is their relatively slow encoding process, which often involves computationally expensive model optimization for each input. Among these methods, autoencoders have emerged as a popular choice for compressing hyperspectral data [2]. These networks learn to project pixel spectra into compact latent spaces from which the full spectral profiles can be reconstructed. Building on this idea, several works such as [29, 30] incorporate autoregressive models to improve entropy coding. Ballé *et al.* later refined these approaches by introducing the use of *hyperpriors*, which provide additional context for more efficient and accurate compression [3].

Implicit neural representations (INRs) have recently gained attention as a promising tool for data compression [12, 11]. For instance, Davies *et al.* applied INRs to the compression of 3D mesh data, demonstrating improved performance over traditional mesh decimation techniques [8]. Similarly, implicit networks have been employed to compress 2D images [45] and videos [6], with encouraging results. Zhang *et al.* also explored INR-based methods for video compression, further validating the versatility of this approach [51]. In our prior work, we extended this idea to the domain of hyperspectral image compression, using implicit neural networks to encode high-dimensional spectral data [40]. Despite their advantages in compact representation and quality, a notable drawback of INR-based compression methods is their typically slow encoding speed, which stems from the need to fit a neural model per data instance.

Lee *et al.* [25] demonstrated that meta-learning can be used to produce sparse and parameter-efficient initializations for implicit neural networks, enabling accurate image reconstruction with significantly fewer parameters. Building on similar ideas, Strumpler *et al.* [45] showed notable gains over earlier work such as [12] by learning effective MLP initializations, and then applying quantization and entropy coding to compress the resulting network weights. Our current work draws direct inspiration from the meta-learning approach proposed in [11], which enhances the INR compression process through modulation-based parameterization. Here, we expand on our earlier hyperspectral image compression framework [40] by integrating a meta-learning strategy. The result is a system that achieves both faster encoding and more compact representations, demonstrating the practical value of combining INRs with meta-learned components in this domain.

3 METHOD

Let us denote a hyperspectral image as $\mathbf{I} \in \mathbb{R}^{W \times H \times C}$, where W and H represent the spatial dimensions and C is the number of spectral bands. Each spatial coordinate (x, y) corresponds to a spectral vector $\mathbf{I}[x, y] \in \mathbb{R}^C$, where $x \in [1, W]$ and $y \in [1, H]$. In our previous work, we showed that it is feasible to use implicit neural representations (INRs) to learn a continuous mapping from spatial locations to spectral signatures. This is achieved by training a function Φ_Θ parameterized by weights Θ such that $\Phi_\Theta(x, y) \mapsto \mathbf{I}[x, y]$. The model is optimized by minimizing the following reconstruction loss:

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

Previous research [47, 44] has established that SIREN networks—multi-layer perceptrons (MLPs) that use sine activation functions—are particularly effective at capturing high-frequency signals over structured grids. These networks have become a standard choice for constructing implicit neural representations. In our setting, we employ a SIREN-based architecture (Φ_Θ) composed of K hidden layers, each utilizing sinusoidal activations. The activations at each layer are denoted by $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K$, with the network defined as:

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

where the input $\mathbf{h}_0 \in \mathbb{R}^2$ corresponds to normalized pixel coordinates. The first layer uses weights $\mathbf{W}_1 \in \mathbb{R}^{d \times 2}$ and biases $\mathbf{b}_1 \in \mathbb{R}^d$, while subsequent layers for $i \in [2, K]$ use $\mathbf{W}_i \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_i \in \mathbb{R}^d$. The final output layer produces:

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

with $\mathbf{W}_{K+1} \in \mathbb{R}^{C \times d}$ and $\mathbf{b}_{K+1} \in \mathbb{R}^C$, yielding $\mathbf{h}_{K+1} \in \mathbb{R}^C$, the predicted spectral vector. The complete set of learnable parameters is given by $\Theta = \{\mathbf{W}_i, \mathbf{b}_i \mid i \in [1, K+1]\}$. Once training is complete, these parameters serve as a compact representation of the hyperspectral image. Compression is achieved when the storage cost of the network parameters is less than that of the original image data.

Although SIREN networks have proven effective for compressing hyperspectral images, the approach has two notable limitations. First, the encoding process is computationally intensive and slow. Second, the method does not take advantage of the common spatial or spectral patterns shared across different images—patterns that are often leveraged in traditional RGB image processing. These drawbacks arise from the need to train a separate SIREN model from scratch for each individual image. Since no parameters are reused, training must be repeated in full for every sample, typically requiring many iterations and significant computational effort.

3.1 Modulated SIREN Network

To overcome the limitations of traditional SIREN-based compression, we propose a meta-learning approach in which a shared SIREN model, referred to as the base network, is used across multiple hyperspectral images. Rather than training separate models for each image, we encode image-specific information via modulations applied to the hidden features $\mathbf{h}_\mathrm{i}$, where $\mathrm{i} \in [1, \mathrm{K}]$. These modulations consist of learned shift and scale parameters.

This idea draws inspiration from the FiLM (Feature-wise Linear Modulation) mechanism introduced by Perez *et al.* [36], where each feature map $\mathbf{h}_\mathrm{i}$ is transformed using:

$$\mathrm{FiLM}(\mathbf{h}_\mathrm{i}) = \gamma_\mathrm{i} \odot \mathbf{h}_\mathrm{i} + \beta_\mathrm{i},$$

with γ_i and β_i denoting element-wise scaling and shifting vectors, respectively. Such operations allow different behaviors to be embedded within the same underlying network by modulating its internal activations.

Chan *et al.* [5] extended this concept by incorporating modulation into SIREN networks for generative modeling. In their formulation, modulations (scale γ and shift β) are applied as follows:

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

Similarly, Mehta *et al.* [27] proposed a modulated INR framework where only multiplicative modulation is applied (scale α_i):

$$\mathbf{h}_\mathrm{i} = \alpha_\mathrm{i} \odot \sin(\mathbf{W}_\mathrm{i}\mathbf{h}_{\mathrm{i}-1} + \mathbf{b}_\mathrm{i}),$$

allowing for efficient parameterization of a family of networks using shared architecture and modulated latent codes.

These approaches demonstrate that it is feasible to generate modulation parameters from a compact latent code. For instance, Chan *et al.* [5] utilize a multilayer perceptron (MLP) to transform a latent vector into the scale (γ_i) and shift (β_i) parameters applied at each layer. In contrast, Mehta *et al.* [27] employ a recursive strategy to derive the scaling factors α_i directly from a fixed latent representation. Despite their effectiveness, both methods require storing not only the shared base network but also the parameters of the auxiliary networks used to compute the modulations. This added storage overhead makes such approaches less practical for data compression applications, where minimizing memory footprint is essential.

Dupont *et al.* [11] investigated how modulation can enhance SIREN-based networks and found that applying only additive shifts β_i to the hidden activations is sufficient to capture meaningful variations across inputs. Their findings indicate that including scale modulations offers negligible benefit, and using scale alone is not effective. Following this insight, our method exclusively uses additive modulations, applying shifts to each layer's activations in the SIREN architecture as follows:

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

where each shift vector $\beta_\mathrm{i} \in \mathbb{R}^\mathrm{d}$. From a compression standpoint, this design is highly efficient: the modulations $\beta_1, \ldots, \beta_\mathrm{K}$ require significantly fewer parameters than storing the full set of weights $\mathbf{W}_i$ and biases $\mathbf{b}_i$. Since the base network is shared across multiple images, its cost can be amortized, making the modulation-based encoding highly compact. Additional compression efficiency can be achieved by generating the modulation parameters from a compact latent code $\varphi \in \mathbb{R}^{d_{\text{latent}}}$. Dupont *et al.* [11] demonstrated that a simple linear transformation is sufficient to map this latent vector to the required modulation β_i, and that employing deeper neural networks such as multi-layer perceptrons offers minimal improvement in this context. Following their findings, we adopt a linear projection of the form:

$$\beta = \mathbf{W}_M\varphi + \mathbf{b}_M, \tag{2}$$

where $\mathbf{W}_M \in \mathbb{R}^{(d)(K)\times d_{\text{latent}}}$ and $\mathbf{b}_M \in \mathbb{R}^{(d)(K)}$ are the weight matrix and bias vector of the linear layer. This layer, which we refer to as the meta network, produces a concatenated vector $\beta = [\beta_1|\cdots|\beta_K]$ that contains the shift modulations for all hidden layers.

With this formulation, a hyperspectral image $\mathbf{I}$ can be reconstructed by evaluating the modulated base network $\Phi_\Theta(x,y;\beta_1,\ldots,\beta_K)$ at each spatial coordinate (x,y). Alternatively, we can use the latent code directly by evaluating $\Phi_\Theta(x,y;\varphi,\Theta_M)$, where $\Theta_M = \{\mathbf{W}_M, \mathbf{b}_M\}$ denotes the parameters of the meta network responsible for generating the modulations (see Figure 1) [38].

3.2 Meta Learning

Model-Agnostic Meta-Learning (MAML) [15] is a framework designed to learn a favorable initialization for model parameters Θ, allowing the model to adapt quickly to new but related tasks using only a few gradient updates. This meta-learning paradigm has been shown to improve the efficiency of training implicit neural representations by reducing the number of optimization steps needed to accurately fit new data [43]. In the context of our work, consider a collection of hyperspectral images denoted as $\mathbf{I}^{(1)},\ldots,\mathbf{I}^{(T)}$. The goal is to determine an initial set of model parameters Θ for the function Φ_Θ such that it can rapidly adapt to any image within this set. MAML employs a two-stage optimization process. In the inner loop, task-specific updates are computed by performing a gradient descent step for each image:

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

where α_{inner} is the learning rate used for adaptation. In the outer loop, the initial parameters Θ are refined based on the performance of the adapted models:

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

where α_{outer} is the meta-learning rate. This process yields a parameter initialization that generalizes well across the distribution of tasks, in this case, hyperspectral image instances. In practical implementations of MAML, the inner loop typically operates on a single randomly selected image t from the dataset. For the outer loop, it is generally sufficient to sample a subset of images rather than using the full batch. This stochastic sampling makes the meta-training process more efficient. The resulting initialization Θ

enables the model to rapidly adapt to a new, unseen hyperspectral image with minimal fine-tuning, significantly reducing the time required for encoding.

The standard MAML formulation is not directly applicable to our scenario, as our goal is to learn image-specific modulations while reusing a common base network across multiple hyperspectral images. To address this, we adopt the strategy proposed by Zintgraf *et al.* [53], which involves partitioning the model parameters into two distinct categories. The first group, known as context parameters, is specific to individual tasks (or images) and is updated during the inner loop. The second group consists of shared parameters that remain consistent across tasks and are optimized in the outer loop through meta-learning.

We incorporate this parameter separation into our framework as follows. Given a collection of hyperspectral images, we designate the base network parameters as Θ, which are shared across all images, and associate each image with its own set of modulation vectors $\beta^{(t)} = \{\beta_0^{(t)}, \dots, \beta_K^{(t)}\}$. During the inner loop, we adapt the modulation parameters for each image by performing a gradient update:

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

where α_{inner} denotes the learning rate used for the image-specific adaptation. In the outer loop, we refine the shared parameters Θ by evaluating the loss with respect to the adapted modulations across the dataset:

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

Starting value for β is fixed and [53] suggests to set the initial values for $\beta = \mathbf{0}$. $\Phi_{[\Theta|\beta]}$ denotes the modulated SIREN network (see Equation 1). To further reduce storage requirements, we generate modulation vectors using the linear mapping introduced in Equation 2, where modulations are derived from a latent code φ. As with β, the latent vector φ is initialized to zero at the beginning of training.

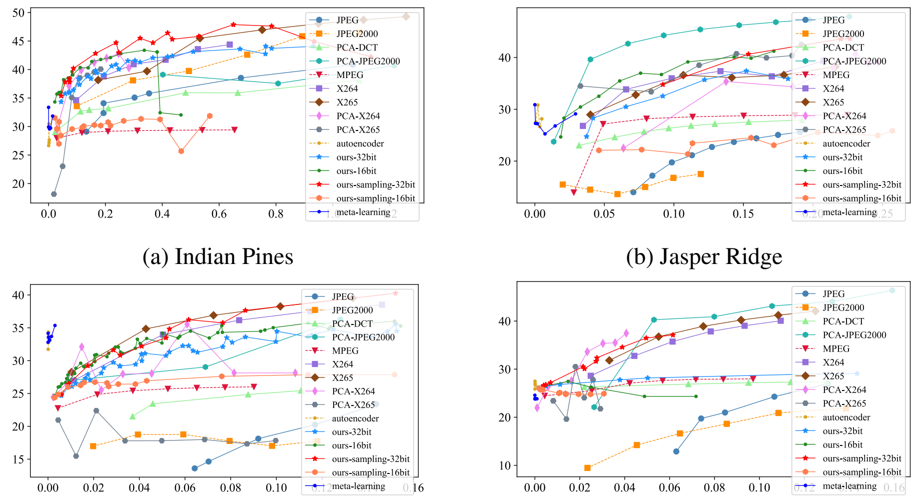
In this formulation, our objective is to learn a unique latent vector $\varphi^{(t)}$ for each image. The training procedure mirrors the earlier modulation-based approach. In the inner loop, we adapt the latent vector for image t by computing:

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

where Θ^+ includes all parameters involved in generating the modulations from the latent vector. In the outer loop, we update the shared parameters Θ^+ by aggregating the gradients computed from the loss with respect to the adapted modulations:

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

In this context, $\Theta^+ = \{\Theta, \mathbf{W}_M, \mathbf{b}_M\}$ represents the full set of shared parameters, encompassing both the weights of the base network and those of the linear projection used to generate modulations from the latent vectors. While Θ^+ remains consistent across all images, each image-specific latent vector $\varphi^{(t)}$ captures the unique content of its corresponding hyperspectral image [38].



(a) Indian Pines

(b) Jasper Ridge

(c) Pavia University

(d) Cuprite

Fig. 2: **PSNR vs.** bpppb **values**. PSNR values achieved at various bpppb for our method (meta-learning), along with those obtained by other methods. x-axis represents bpppb values and y-axis represents PSNR values.

4 RESULTS

We selected JPEG [17, 37], JPEG2000 [9] and PCA-DCT [31] schemes as baselines as these are widely adopted in the hyperspectral image analysis pipelines. In addition to these, we compare our proposed approach against both our prior work on implicit neural representations for hyperspectral image compression [40] and a broad set of techniques from the literature, including: PCA-JPEG2000 [9], FPCA-JPEG2000 [28], RPM [35], 3D SPECK [48], 3D DCT [49], 3D DWT+SVR [52], WSRC [32], MPEG [24], X264 [20], X265 [20], PCA-X264 [20], PCA-X265 [20], PCA-JPEG2000 [20, 22, 21], and autoencoder [23] methods. We also include comparisons with several variants of our own earlier approaches that employ implicit neural representations: ours-32bit, ours-16bit,m ours-sampling-32bit, and ours-sampling-16bit [39].

We conduct our experiments on four widely used hyperspectral image benchmarks: (1) Indian Pines (145 × 145 × 220), (2) Jasper Ridge (100 × 100 × 224), (3) Pavia University (610 × 340 × 103), and (4) Cuprite (614 × 512 × 224).

4.1 Metrics

To evaluate the reconstruction quality of compressed hyperspectral images, we employ two standard metrics: Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE). PSNR, measured in decibels (dB), is widely used in image compression research to quantify the fidelity of a reconstructed image relative to its original counterpart. Higher PSNR values indicate better preservation of the original image content and thus better compression performance. Complementarily, MSE captures the average squared difference between the original and compressed images. Lower MSE values correspond to more accurate reconstructions.

The MSE is defined as:

$$\mathrm{MSE} = \sum_{\mathrm{i}} \frac{||\mathrm{I[i]} - \tilde{\mathrm{I}}\mathrm{[i]}||}{\mathrm{i}}, \tag{3}$$

where I[i] and $\tilde{\mathrm{I}}$[i] refer to the original and reconstructed pixel values, respectively, and i ranges over all pixel indices. Using MSE, PSNR is computed as:

$$\mathrm{PSNR} = 10\ \log_{10}\left(\frac{\mathrm{R}^2}{\mathrm{MSE}}\right), \tag{4}$$

where R represents the maximum possible value of the input pixel intensity.

In addition to PSNR and MSE, we report the bits-per-pixel-per-band (bpppb) to quantify the compression efficiency of our method. This metric reflects the number of bits required to represent each pixel in each spectral band after compression. Lower bpppb values indicate higher compression rates, as less information is needed to encode the image.

For uncompressed hyperspectral images, bpppb typically corresponds to either 8 or 32 bits, depending on whether the data is stored in integer or floating-point format. In most cases, hyperspectral images are stored using 32-bit floating-point values per channel. The bpppb metric is calculated as:

$$\mathrm{bpppb} = \frac{\#\mathrm{parameters} \times (\mathrm{bits\ per\ parameter})}{(\mathrm{pixels\ per\ band}) \times \#\mathrm{bands}}, \tag{5}$$

where the numerator represents the total number of bits required to store the model, and the denominator corresponds to the number of data values in the original image.

4.2 Practical matters

All models in our framework are implemented using PyTorch [34]. For the inner-loop updates, we apply Stochastic Gradient Descent (SGD) with a learning rate of 1×10^{-2}, while the outer-loop optimization is carried out using the Adam optimizer with a learning rate set to either 1×10^{-6} or 3×10^{-6}. Prior to training, spatial pixel coordinates (x, y) are normalized to lie within the range $[-1, 1] \times [-1, 1]$, and spectral values at each pixel are scaled to fall between 0 and 1. In cases where the shared base network is applied to hyperspectral datasets with differing numbers of spectral channels, we exclude any unmatched channels during loss computation to ensure consistency.

4.3 **PSNR vs.** bpppb

Figure 2 shows PSNR values at various compression rates for different methods. Specifically, we compare our approach, labeled as meta-learning, with ours-32bit, ours-16bit, ours-sampling-32bit, ours-sampling-16bit, JPEG, JPEG2000, PCA-DCT, MPEG, X264,

Indian Pines					Jasper Ridge				
Method	Size (KB)	PSNR	bpppb	$\mathrm{n_h, w_h}$	Method	Size (KB)	PSNR	bpppb	$\mathrm{n_h, w_h}$
-	9251	∞	16	-,-	-	4800	∞	16	-,-
JPEG [17, 37]	115.6	34.085	0.2	-,-	JPEG [17, 37]	30	21.130	0.1	-,-
JPEG2000 [9]	115.6	35.84	0.2	-,-	JPEG2000 [9]	30	17.494	0.1	-,-
PCA-DCT [31]	115.6	33.173	0.2	-,-	PCA-DCT [31]	30	26.821	0.1	-,-
PCA-JPEG2000 [9]	115.6	39.5	0.2	-,-	PCA-JPEG2000 [9]	30	-	0.1	-,-
FPCA-JPEG2000 [28]	115.6	40.5	0.2	-.-	FPCA-JPEG2000 [28]	30	-	0.1	-,-
HEVC [46]	115.6	32	0.2	-,-	HEVC [46]	30	-	0.1	-,-
RPM [35]	115.6	38	0.2	-,-	RPM [35]	30	-	0.1	-,-
3D SPECK [48]	115.6	-	0.2	-,-	3D SPECK [48]	30	-	0.1	-,-
3D DCT [49]	115.6	-	0.2	-,-	3D DCT [49]	30	-	0.1	-,-
3D-DWT-SVR [52]	115.6	-	0.2	-,-	3D-DWT-SVR [52]	30	-	0.1	-,-
WSRC [32]	115.6	-	0.2	-,-	WSRC [32]	30	-	0.1	-,-
ours-32bit [39]	115.6	42.22	0.2	5,60	ours-32bit [39]	30	32.54	0.1	5,20
ours-16bit [39]	57.5	29.68	0.1	5,60	ours-16bit [39]	15	22.07	0.06	5,20
ours-sampling-32bit [40]	115.6	42.22	0.2	5,60	ours-sampling-32bit [40]	30	34.77	0.1	5,20
ours-sampling-16bit [40]	57.5	29.68	0.2	5,60	ours-sampling-16bit [40]	15	22.07	0.06	5,20
meta-learning	**0.0032**	33.36	6.9e-6	10,128	meta-learning	**0.0032**	30.87	1.4e-5	10,128
Pavia University					Cuprite				
Method	Size (KB)	PSNR	bpppb	$\mathrm{n_h, w_h}$	Method	Size (KB)	PSNR	bpppb	$\mathrm{n_h, w_h}$
-	42724	∞	16	-,-	-	140836	∞	16	-,-
JPEG [17, 37]	267	20.253	0.1	-,-	JPEG [17, 37]	880.2	24.274	0.1	-,-
JPEG2000 [9]	267	17.752	0.1	-,-	JPEG2000 [9]	880.2	20.889	0.1	-,-
PCA-DCT [31]	267	25.436	0.1	-,-	PCA-DCT [31]	880.2	27.302	0.1	-,-
PCA-JPEG2000 [9]	267	-	0.1	-,-	PCA-JPEG2000 [9]	880.2	40.90	0.1	-,-
FPCA-JPEG2000 [28]	267	-	0.1	-,-	FPCA-JPEG2000 [28]	880.2	-	0.1	-,-
HEVC [46]	267	-	0.1	-,-	HEVC [46]	880.2	31	0.1	-,-
RPM [35]	267	-	0.1	-,-	RPM [35]	880.2	34	0.1	-,-
3D SPECK [48]	267	-	0.1	-,-	3D SPECK [48]	880.2	27.1	0.1	-,-
3D DCT [49]	267	-	0.1	-,-	3D DCT [49]	880.2	33.4	0.1	-,-
3D-DWT-SVR [52]	267	-	0.1	-,-	3D-DWT-SVR [52]	880.2	28.20	0.1	-,-
WSRC [32]	267	-	0.1	-,-	WSRC [32]	880.2	35	0.1	-,-
ours-32bit [39]	267	34.46	0.1	10,80	ours-32bit [39]	880.2	28.954	0.1	25,100
ours-16bit [39]	133.5	34.17	0.05	10,80	ours-16bit [39]	440.1	24.334	0.06	25,100
ours-sampling-32bit [40]	267	38.08	0.1	10,80	ours-sampling-32bit [40]	880.2	36.55	0.1	25,90
ours-sampling-16bit [40]	133.5	27.49	0.05	10,80	ours-sampling-16bit [40]	440.1	24.91	0.06	25,90
meta-learning	**0.0032**	32.79	1.4e-6	10,128	meta-learning	**0.0032**	24.57	4.5e-7	10,128

Table 1: Compression rates on four benchmarks. For each benchmark, the first row lists the actual size (in KB) of the original hyperspectral image. For each method, the first column shows the size of the compressed image (in KB), the second column shows the PSNR achieved by comparing the decompressed image with the original image, and the third column shows the bpppb achieved. For approaches that rely upon implicit neural representations, the structure of the network is described by showing the number of hidden layers $\mathrm{n_h}$ and the width of these layers $\mathrm{w_h}$. Please note that previously, K is used to denote the number of hidden layers and d is used to denote the width of these layers, i.e., $\mathrm{n_h}$ = K and $\mathrm{n_w}$ = d. The size of the base network, which is shared across all datasets in the meta-learning method, is 710 KB.

Dataset	Method	bppppb	compression time (Sec)	decompression time (Sec)	PSNR ↑
Indian Pines	JPEG [17, 37]	0.1	7.353	3.27	27.47
	JPEG2000 [9]	0.1	0.1455	0.3115	33.58
	PCA-DCT [31]	0.1	1.66	0.04	32.28
	ours-32bit [39]	0.1	243.64	0	36.98
	ours-16bit [39]	0.05	243.64	0	36.95
	ours-sampling-32bit [40]	0.1	282.08	0.0005	40.1
	ours-sampling-16bit [40]	0.05	282.08	0.0005	40.1
	meta-learning	6.9e-6	**0.033**	**0.000717**	33.36
Jasper Ridge	JPEG [17, 37]	0.1	3.71	1.62	21.13
	JPEG2000 [9]	0.1	0.138	0.395	17.49
	PCA-DCT [31]	0.1	1.029	0.027	26.82
	ours-32bit [39]	0.1	312.38	0.0005	32.54
	ours-16bit [39]	0.06	312.38	0.0005	32.51
	ours-sampling-32bit [40]	0.1	75.91	0.0005	34.77
	ours-sampling-16bit [40]	0.06	75.91	0.0005	22.07
	meta-learning	1.4e-5	**0.025**	**0.0007**	30.87
Pavia University	JPEG [17, 37]	0.1	33.86	14.61	20.25
	JPEG2000 [9]	0.1	0.408	0.628	17.75
	PCA-DCT [31]	0.1	6.525	0.235	25.43
	ours-32bit [39]	0.1	780.16	0.0009	34.46
	ours-16bit [39]	0.05	780.16	0.0009	34.17
	ours-sampling-32bit [40]	0.1	72.512	0.0004	38.08
	ours-sampling-16bit [40]	0.05	72.512	0.0004	27.02
	meta-learning	1.4e-6	**0.43**	**0.0006**	32.79
Cuprite	JPEG [17, 37]	0.06	101.195	45.02	12.88
	JPEG2000 [9]	0.06	1.193	2.476	15.16
	PCA-DCT [31]	0.06	11.67	0.754	26.75
	ours-32bit [39]	0.06	1565.97	0.001	28.02
	ours-16bit [39]	0.03	1565.97	0.001	27.90
	ours-sampling-32bit [40]	0.06	664.87	0.001	37.27
	ours-sampling-16bit [40]	0.03	664.87	0.001	24.85
	meta-learning	4.5e-7	**1.11**	**0.0007**	24.57

Table 2: Compression and decompression times for various methods. The proposed method (meta-learning) achieves the fastest compression times of any method on the four benchmarks. The size of the base network, which is shared across all datasets in the meta-learning method, is 710 KB.

X265, PCA-X264, PCA-X265, PCA-JPEG2000, and autoencoder 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.

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.

4.4 Compression Results

As shown in Table 1, our proposed meta-learning-based compression method consistently produces the smallest file sizes across all evaluated benchmarks. The table provides a comparison of compression performance, including PSNR values and output sizes (in KB), for a variety of existing methods. It is important to note that some entries are missing due to unavailable results for certain benchmarks—for instance, PSNR values for the 3D_SPECK method are not reported for the Indian Pines dataset. Across all four datasets, our approach achieves the highest compression rates, significantly reducing the storage footprint of hyperspectral images. However, in terms of reconstruction quality, the PSNR achieved by meta-learning is slightly lower than that of some competing techniques on Indian Pines, Jasper Ridge, and Cuprite. On the Pavia University dataset, our method delivers PSNR performance comparable to the ours-sampling-32bit baseline. While there is room for improvement in reconstruction fidelity, it is noteworthy that our method offers these results at a fraction of the storage cost required by other compression schemes.

A primary motivation for this study was to significantly reduce the lengthy compression times typically associated with implicit neural representation-based methods for hyperspectral image compression. Table 2 summarizes the compression and decompression times, along with corresponding PSNR values, for a range of competing approaches. Notably, our meta-learning framework delivers the fastest compression times among all evaluated methods. More importantly, it outperforms prior implicit neural compression techniques by a substantial margin in terms of speed, demonstrating that our method is far more practical for real-time or large-scale deployment.

Table 3 evaluates meta-learning against seven video-based methods that treat various channels of a hyperspectral image as frames of a video and employ video coding techniques to achieve compression.[2] Our method achieves better results than all other methods on the Pavia University dataset. Additionally, our method achieves the second-best PSNR value on the Cuprite dataset.

5 Proof of Concept: Compressing a Large Hyperspectral Image

We evaluated our meta-learned compression network on a large hyperspectral image to demonstrate its ability to handle high-resolution images (Figure 3). The spatial resolution of this image is 4192×6708, and it has 270 channels. The image requires approximately 28.2 GB of storage space. The size of this image underscores the need to develop compression algorithms for hyperspectral images. The model proposed in this paper successfully compresses this image to 141 KB.

To manage the complexity of the data, we partition the image into a 7×7 grid and treat each row as an independent unit. Each row is compressed using a dedicated

[2] It is not possible to include the results presented in Table 3 in Table 1 since the bpppb values used by video-based methods do not match those used by methods listed in Table 1.

	Method	bpppb	PSNR ↑		Method	bpppb	PSNR ↑
Indian Pines	X264 [20, 22, 21]	0.1	34.61	Jasper Ridge	X264 [20, 22, 21]	0.15	37.35
	X265 [20, 22, 21]	0.1	38.5		X265 [20, 22, 21]	0.15	36.12
	PCA-X264 [20, 22, 21]	0.1	39.8		PCA-X264 [20, 22, 21]	0.15	35.35
	PCA-X265 [20, 22, 21]	0.1	38.1		PCA-X265 [20, 22, 21]	0.15	39.94
	MPEG [24]	0.1	28.9		MPEG [24]	0.15	28.75
	HEVC [46]	0.1	30		HEVC [46]	0.15	-
	RPM [35]	0.1	31		RPM [35]	0.15	-
	meta-learning	0.004151	36.46		meta-learning	0.0085	36.67
Pavia University	X264 [20, 22, 21]	0.1	37.17	Cuprite	X264 [20, 22, 21]	0.03	28.6
	X265 [20, 22, 21]	0.1	37.90		X265 [20, 22, 21]	0.03	31.8
	PCA-X264 [20, 22, 21]	0.1	28.13		PCA-X264 [20, 22, 21]	0.03	**35.5**
	PCA-X265 [20, 22, 21]	0.1	17.82		PCA-X265 [20, 22, 21]	0.03	21.7
	MPEG [24]	0.1	26.01		MPEG [24]	0.03	25.5
	HEVC [46]	0.1	-		HEVC [46]	0.03	25
	RPM [35]	0.1	-		RPM [35]	0.03	29
	meta-learning	0.0008	39.1		meta-learning	0.0001	33.64

Table 3: Comparing the proposed method (meta-learning) against video-based schemes. Our method achieves better results than all other methods on the Pavia University dataset. Additionally, our method achieves the second-best PSNR value on the Cuprite dataset. The size of the base network, which is shared across all datasets in the meta-learning method, is 710 KB.

network trained to capture its unique spatial and spectral characteristics. In total, we train seven distinct networks, one per row, ensuring that the compression is tailored to the local structure and variability of the data. Each row is divided into seven cells. The structure that is unique to each cell is captured through modulations. Each model contains

$$(2 \times 10 + 10) + 4(10 \times 10 + 10) + (10 \times 270) + (32 \times 50 + 50) = 4820$$

parameters. Latent modulations $\varphi \in \mathbb{R}^{32}$ capture the structure in each cell. Therefore, for each cell we need to store 32 parameters. The number of parameters for the seven models plus the modulations is

$$7(4820 + 7(32)) = 35,068.$$

Assume we store each parameter as a 4-byte floating point value, the disk space needed to store the compressed image is roughly 141 KB. This is significantly less than the disk space occupied by the uncompressed image.

We compare the performance of our method against several baselines, including ours-sampling-32bit, ours-sampling-16bit, JPEG, and MPEG, with a focus on the Peak Signal-to-Noise Ratio (PSNR). Due to memory constraints, we were not able to use JPEG to compress the entire image in a single step. Thus, the image was divided into smaller tiles. Next, each tile was compressed using JPEG. The PSNR value was computed by reconstructing the full image. Table 4 presents this comparison. Our meta-

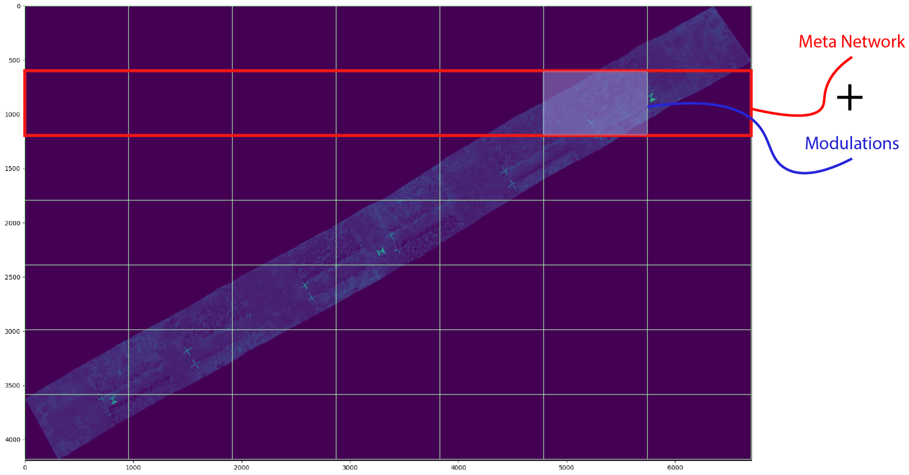


Fig. 3: Visualization of the large (28.2 GB) 4192 × 6708 hyperspectral image with 270 channels. The image is divided into a 7 × 7 grid. Each row is compressed independently. Row specific meta networks capture the overall row structure; whereas, modulations are responsible to encoding the structure present in the cell within a row. A total of 7 meta networks are used to compress this image.

learned model achieves PSNR values comparable to JPEG and MPEG, while significantly reducing the bits per pixel per band (bpppb), demonstrating both its compression efficiency and high fidelity.

6 CONCLUSIONS

In this work, we introduced a meta-learning framework for compressing hyperspectral images using implicit neural representations. Our method separates image-specific content from a shared, meta-trained base network, enabling efficient encoding through modulation vectors generated from compact latent codes. This design facilitates both faster compression and significant reductions in storage requirements. We extended previous work by conducting additional comparative studies, including video-based compression baselines, and demonstrated the scalability of our method on a large hyperspectral dataset exceeding 28 GB. Our findings reveal that the proposed method achieves state-of-the-art compression speed with competitive or superior PSNR scores, especially at extremely low compression rates. These results establish our framework as a practical and efficient solution for large-scale hyperspectral data storage and transmission. Future directions include improving reconstruction quality through entropy coding and exploring downstream task performance under compressed representations.

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 4: 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.

References

1. Afromowitz, M.A., Callis, J.B., Heimbach, D.M., DeSoto, L.A., Norton, M.K.: Multispectral imaging of burn wounds. In: Medical Imaging II. vol. 914, pp. 500–504. SPIE, Newport Beach, California, USA (June 1988)
2. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: International Conference on Learning Representations (2022)
3. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: International Conference on Learning Representations. Vancouver, Canada (April 30–May 3 2018)
4. Carrasco, O., Gomez, R.B., Chainani, A., Roper, W.E.: Hyperspectral imaging applied to medical diagnoses and food safety. In: Geo-Spatial and Temporal Image and Data Exploitation III. vol. 5097, pp. 215–221. SPIE, Orlando, Florida, USA (April 21 and 24 2003)
5. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: 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. pp. 5799–5809. Nashville, Tennessee, USA (held online) (June 19–25 2021)
6. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: Nerv: Neural representations for videos. Advances in Neural Information Processing Systems **34**, 21557–21568 (2021)
7. Clark, R.N., Swayze, G.A.: Mapping minerals, amorphous materials, environmental materials, vegetation, water, ice and snow, and other materials: the usgs tricorder algorithm. In: JPL, Summaries of the Fifth Annual JPL Airborne Earth Science Workshop. Volume 1: AVIRIS Workshop. Pasadena, California, USA (January 23–26 1995)
8. Davies, T., Nowrouzezahrai, D., Jacobson, A.: On the effectiveness of weight-encoded neural implicit 3d shapes. arXiv preprint arXiv:2009.09808 (2020)
9. Du, Q., Fowler, J.E.: Hyperspectral image compression using jpeg2000 and principal component analysis. IEEE Geoscience and Remote sensing letters **4**(2), 201–205 (2007)
10. Dua, Y., Kumar, V., Singh, R.S.: Comprehensive review of hyperspectral image compression algorithms. Optical Engineering **59**(9), 090902 (2020). https://doi.org/10.1117/1.OE.59.9.090902, https://doi.org/10.1117/1.OE.59.9.090902
11. Dupont, E., Loya, H., Alizadeh, M., Golinski, A., Teh, Y., Doucet, A.: Coin++: neural compression across modalities. Transactions on Machine Learning Research **2022**(11) (2022)
12. Dupont, E., Golinski, A., Alizadeh, M., Teh, Y.W., Doucet, A.: Coin: Compression with implicit neural representations. In: Neural Compression: From Information Theory to Applications–Workshop@ ICLR 2021. ICLR, Vienna, Austria (held online) (May 4–8 2021)

13. Edelman, G.J., Gaston, E., Van Leeuwen, T.G., Cullen, P., Aalders, M.C.: Hyperspectral imaging for non-contact analysis of forensic traces. Forensic science international **223**(1-3), 28–39 (2012)
14. Feng, Y.Z., Sun, D.W.: Application of hyperspectral imaging in food safety inspection and control: a review. Critical reviews in food science and nutrition **52**(11), 1039–1058 (2012)
15. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the International Conference on Machine Learning. pp. 1126–1135. PMLR, Sydney, Australia (August 6–11 2017)
16. Goetz, A.F., Vane, G., Solomon, J.E., Rock, B.N.: Imaging spectrometry for earth remote sensing. science **228**(4704), 1147–1153 (1985)
17. Good, W.F., Maitz, G.S., Gur, D.: Joint photographic experts group (jpeg) compatible data compression of mammograms. Journal of Digital Imaging **7**(3), 123–132 (1994)
18. Gowen, A.A., O'Donnell, C.P., Cullen, P.J., Downey, G., Frias, J.M.: Hyperspectral imaging–an emerging process analytical tool for food quality and safety control. Trends in food science & technology **18**(12), 590–598 (2007)
19. Kuula, J., Pölönen, I., Puupponen, H.H., Selander, T., Reinikainen, T., Kalenius, T., Saari, H.: Using vis/nir and ir spectral cameras for detecting and separating crime scene details. In: Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense XI. vol. 8359, p. 83590P. International Society for Optics and Photonics, Baltimore, Maryland, USA (April 23–27 2012)
20. Kwan, C., Larkin, J.: New results in perceptually lossless compression of hyperspectral images. Journal of Signal and Information Processing **10**(3), 96–124 (2019)
21. Kwan, C., Larkin, J., Budavari, B., Chou, B.: Compression algorithm selection for multispectral mastcam images. Signal & Image Processing: An International Journal (SIPIJ) **10**(1) (2019)
22. Kwan, C., Larkin, J., et al.: Perceptually lossless compression for mastcam multispectral images: A comparative study. Journal of Signal and Information Processing **10**(04), 139 (2019)
23. La Grassa, R., Re, C., Cremonese, G., Gallo, I.: Hyperspectral data compression using fully convolutional autoencoder. Remote Sensing **14**(10), 2472 (2022)
24. Le Gall, D.: Mpeg: A video compression standard for multimedia applications. Communications of the ACM **34**(4), 46–58 (1991)
25. Lee, J., Tack, J., Lee, N., Shin, J.: Meta-learning sparse implicit neural representations. Advances in Neural Information Processing Systems **34**, 11769–11780 (2021)
26. Liang, H.: Advances in multispectral and hyperspectral imaging for archaeology and art conservation. Applied Physics A **106**(2), 309–323 (2012)
27. Mehta, I., Gharbi, M., Barnes, C., Shechtman, E., Ramamoorthi, R., Chandraker, M.: Modulated periodic activations for generalizable local functional representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14214–14223. Montreal, Canada (held online) (October 11–17 2021)
28. Mei, S., Khan, B.M., Zhang, Y., Du, Q.: Low-complexity hyperspectral image compression using folded pca and jpeg2000. In: IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium. pp. 4756–4759. IEEE, Valencia, Spain (July 22–27 2018)
29. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4394–4402. Salt Lake City, Utah, USA (June 18–22 2018)
30. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. Advances in neural information processing systems **31** (2018)
31. Nian, Y., Liu, Y., Ye, Z.: Pairwise klt-based compression for multispectral images. Sensing and Imaging **17**(1), 1–15 (2016)

32. Ouahioune, M., Ameur, S., Lahdir, M.: Enhancing hyperspectral image compression using learning-based super-resolution technique. Earth Science Informatics **14**(3), 1173–1183 (2021)
33. Padoan, R., Steemers, T., Klein, M., Aalderink, B., De Bruin, G.: Quantitative hyperspectral imaging of historical documents: technique and applications. Art Proceedings pp. 25–30 (2008)
34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
35. Paul, M., Xiao, R., Gao, J., Bossomaier, T.: Reflectance prediction modelling for residual-based hyperspectral image coding. PloS one **11**(10), e0161212 (2016)
36. Perez, E., Strub, F., de Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32. New Orleans, Louisiana, USA (February 2–7 2018)
37. Qiao, T., Ren, J., Sun, M., Zheng, J., Marshall, S.: 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 (2014)
38. Qureshi, F., Rezasoltani, S.: Hyperspectral image compression using implicit neural representation and meta-learned based network. In: Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM. pp. 23–31. INSTICC, SciTePress (2025). https://doi.org/10.5220/0013121200003905
39. Rezasoltani, S., Qureshi, F.Z.: Hyperspectral image compression using implicit neural representations. In: 2023 20th Conference on Robots and Vision (CRV). pp. 248–255. Montreal, Quebec, Canada (June 6–8 2023). https://doi.org/10.1109/CRV60082.2023.00039
40. Rezasoltani, S., Qureshi, F.Z.: Hyperspectral image compression using sampling and implicit neural representations. IEEE Transactions on Geoscience and Remote Sensing pp. 1–12 (2024). https://doi.org/10.1109/TGRS.2024.3509718
41. Rezasoltani, S., Qureshi, F.Z.: Hyperspectral image compression using implicit neural representation and meta-learned based network. In: Proc. 14th International Conference on Pattern Recognition Applications and Methods. p. 9pp. Porto (February 2025)
42. Schuler, R.L., Kish, P.E., Plese, C.A.: Preliminary observations on the ability of hyperspectral imaging to provide detection and visualization of bloodstain patterns on black fabrics. Journal of forensic sciences **57**(6), 1562–1569 (2012)
43. Sitzmann, V., Chan, E., Tucker, R., Snavely, N., Wetzstein, G.: Metasdf: Meta-learning signed distance functions. Advances in Neural Information Processing Systems **33**, 10136–10147 (2020)
44. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems **33**, 7462–7473 (2020)
45. Strümpler, Y., Postels, J., Yang, R., Van Gool, L., Tombari, F.: Implicit neural representations for image compression. In: European Conference on Computer Vision. pp. 74–91. Springer, Tel Aviv, Israel (October 23–27 2022)
46. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. IEEE Transactions on circuits and systems for video technology **22**(12), 1649–1668 (2012)
47. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems **33**, 7537–7547 (2020)
48. Tang, X., Pearlman, W.A.: Three-dimensional wavelet-based compression of hyperspectral images. In: Hyperspectral data compression, pp. 273–308. Springer (2006)

49. Yadav, R.J., Nagmode, M.: Compression of hyperspectral image using pca–dct technology. In: Innovations in Electronics and Communication Engineering: Proceedings of the Fifth ICIECE 2016. pp. 269–277. Springer, Hyderabad, India (July 8–9 2018)
50. Zhang, F., Chen, C., Wan, Y.: A survey on hyperspectral remote sensing image compression. In: Proc. 2023 IEEE International Geoscience and Remote Sensing Symposium. pp. 7400–7403 (2023). https://doi.org/10.1109/IGARSS52108.2023.10281919
51. Zhang, Y., van Rozendaal, T., Brehmer, J., Nagel, M., Cohen, T.: Implicit neural video compression. In: ICLR Workshop on Deep Generative Models for Highly Structured Data (2021)
52. Zikiou, N., Lahdir, M., Helbert, D.: Support vector regression-based 3d-wavelet texture learning for hyperspectral image compression. The Visual Computer **36**(7), 1473–1490 (2020)
53. Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., Whiteson, S.: CAVIA: Fast context adaptation via meta-learning. In: Proceedings of the International Conference on Machine Learning. pp. 7693–7702. PMLR, Long Beach, California, USA (June 9–15 2019)