UNIVERSITY OF CONCEPCIÓN

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING

Supervisor:

Alejandro Rojas, PhD.

Co-Supervisor:

Hugo Garcés, PhD.

Thesis
for the degree:

Master in Electrical Engineering

Mineral Classification using SWIR Hyperspectral Imaging: From Classical Machine Learning Techniques to Transformers

Concepción, July 2025

José Ignacio Cifuentes Ramírez

University of Concepcion
Faculty of Engineering
Department of Electrical Engineering

Supervisor:
Alejandro Rojas, PhD.

Co-Supervisor:
Hugo Garcés, PhD.

Mineral Classification using SWIR Hyperspectral Imaging: From Classical Machine Learning Techniques to Transformers

José Ignacio Cifuentes Ramírez

Thesis for the degree

Master in Electrical Engineering

July 2025

Abstract

This work provides a comprehensive review of supervised models for hyperspectral imaging, from classical machine learning techniques to state-of-the-art Transformers. Additionally, this research proposes a multispectral optical sensor that harnesses capabilities from SWIR-hyperspectral to SWIR-multispectral for mineral classification using classical machine learning techniques. Furthermore, a novel 1D-RMC model was described using a solid mathematical foundation, thereby ensuring its implementation and performance viability based on previous results levering the spectral domain. Experiments were carried out using three datasets: one containing nine classes of minerals, the second containing reflectance images of 130 samples of 76 distinct minerals, and third, to ensure the 1D-RMC capabilities, the standard hyperspectral dataset "Indian Pines" and "Houston2013." The results showed that it is feasible to build an optical sensor with five channels to obtain competitive results compared with hyperspectral data using machine learning techniques. Furthermore, the proposed deep learning models outperformed the traditional machine learning algorithms by at least 3% in terms of accuracy and F1-score for the mineral dataset. Finally, 10 well-known models, including Transfomers, were compared with 1D-RMC, obtaining competitive results with state-of-the-art backbone network models, suprassing by at least 2% in terms of Overall Accuracy (OA), average accuracy (AA), and Kappa (κ) metrics.

Key words: Mineral classification, Transformers, Hyperspectral embedding, Hyperspectral Imaging.

Acknowledgment

I am grateful to express my sincere appreciation to my family and friends for their constant support; Professor Dr. Luis Arias, Dr. Alejandro Rojas, and Dr. Hugo Gracés for their invaluable guidance; and the Spectral Processing and Radiometry Laboratory team. Furthermore, Dr. Eric Pirard from the University of Liège, for providing the mineral hyperspectral dataset and for his expert feedback, which contributed significantly to the geological and insightful observations on radiometric data, strengthened the methodological foundations and scientific depth of this thesis. This research was supported by the Chilean National Research and Development Agency (ANID) through FONDECYT 1211184 and 1220903, Basal (FB008), and the Patagón Supercomputer of Universidad Austral de Chile (FONDEQUIP EQM180042).

Main Index

Figure Index

Table Index

Acronym

AA Aveerage Accuracy

AE Autoencoders

AI Artificial Intelligence

ANN Artificial Neural Networks

BLSTM Bidirectional Long Short-Term Memory

BRNN Bidirectional Recurrent Neural Networks

CCD Charge-Coupled Device

CNN Convolutional Neural Networks

DL Deep Learning

GAN Generative Adversarial Networks

HSI Hyperspectral Imaging

LDA Linear Discriminant Analysis

LP Linear Projection

LR Logistic Regression

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multilayer Perceptron

MSA Multihead Self-Attention

NLP Natural Language Processing

NN Neural Networks

OA Overall Accuracy

PCA Principal Component Analysis

PE Positional Encoding

pXRF Portable X-ray fluorescence

ResNet Residual-Network

RNN Recurrent Neural Networks

RPE Relative Positional Encoding

sPCA Sparse Principal Component Analysis

SSAN Spectral–Spatial Attention Network

SVD Singular Value Decomposition

SVM Support Vector Machine

SWIR Short-Wave Infrared Range

ViT Vision Transformer

1. Introduction

Artificial intelligence (AI) has gained significant relevance in geological studies for mineral exploration because of its ability to assist field geologists by supporting lithological and mineralogical interpretations, facilitating the planning of ground-truth campaigns, and enabling quick and reliable identification of geological materials as a complementary decision-making tool [1–3].

The integration of artificial intelligence with optical sensing technologies has revolutionized the design and implementation of specialized sensors for specific analytical tasks. AI methods, particularly machine learning (ML), enable the creation of highly optimized optical sensors that can perform complex classification, regression, and prediction tasks with unprecedented accuracy and efficiency [4, 5].

These intelligent systems leverage data-driven modeling to optimize the selection of spectral channels, improve signal interpretation, and even emulate the full-spectrum performance using reduced and cost-effective hardware configurations. By integrating AI into the sensor design process, researchers can target specific tasks such as classification, regression, and anomaly detection with enhanced precision and efficiency. This highlights the transformative role of machine learning in the development of next-generation optical sensors, making them smarter, more efficient, and better suited for domain-specific tasks in science and industry [6–8]. However, it is crucial to recognize that the ability of these models to generalize is fundamentally limited by the variety and representativeness of the datasets used for the training. If the dataset includes only a small selection of mineral species, the models may find it challenging to identify new geological samples accurately.

Meanwhile, there has been a significant advancement in the use of deep learning (DL) for several applications, including mineralogical classification using hyperspectral reflectance data in the near-infrared and short-wave infrared range. The main architectures employed for this purpose are Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Attention based networks. These models frequently use spectral-spatial techniques, which often modify the data distributions and edges associated with the sample domains [9].

One of the fundamental advantages of DL models is that, unlike feature extraction-based methods commonly employed in traditional machine learning (ML) models, such as support

vector machines (SVM), linear discriminant analysis (LDA), and decision trees. DL models can directly use and adjust the weights and feature map, often deliver better results than traditional ML algorithms [10]. However, they are strongly subject to overfitting of training data, making a diverse and large-scale dataset essential for generalizability.

Transformers have revolutionized the field of AI by delivering outstanding results in many applications and the interpretability of these results. Originally designed for natural language processing (NLP), these types of architectures present limitations in other domains such as spectral data processing. One limitation of the original model is that positional encoding (PE) is restricted to considering the absolute distance of the input data without assuming their non-symmetric and relative distance, and it does not consider multi-scale feature representations based solely on spectral features.

In this work, a comprehensive review of artificial neural networks (ANN) and spectral-based ANN, including state-of-the-art Transformers, is discussed. Classical data exploration was performed using principal component analysis, and a theoretical optical sensor was designed and simulated to classify the data. Second, to verify multi-scale feature dependency, we performed three classic ANN backbone models based on RNN, CNN-1D and CNN-2D, Transformers, and Finally, proposed a novel model named 1-D RMC.

The experiments were carried out using two datasets, one which contains 9 classes of minerals and the second, the public dataset "A 2D hyperspectral library of mineral reflectance from 900 to 2500 nm' [11] for Deep learning experiments, which contains reflectance images of 130 samples of 76 distinct minerals. Both samples were scanned using a SPECIM® SWIR camera with a linear array of 1 $\times$ 320 pixels and a spectral resolution of 256 channels (~6.25 nm) from 900 to 2500 nm.

1.1 Bibliography and Discussion

In the last decade, numerous artificial intelligence techniques have been proposed for hyperspectral imaging (HSI) classification, including classical machine learning methods and deep learning approaches such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs) [12–14]. Current approaches for HSI can be divided into three main categories: spectral, spatial, and spectral-spatial dimensions. Zhang et al. [15] have introduced a novel CNN-based model that considers 3D neighboring patches $P_{\alpha,\beta} \in \mathbb{R}^{s\times s\times b}$, a $S \times S$ window centered at (α, β), and b principal component dimensions reduced from a data cuve $X \in \mathbb{R}^{H\times W\times d}$, to join the spectral-spatial representation using a 3D-CNN followed by a 2D-CNN. This approach can achieve better spatial level representations and outperforms previous works based only on 2D or 3D representations; subsequently, multiple CNN were proposed based on this concept. Mou *et. al* [16] poposed a GRU based model for hyperspectral imaging due his better performance in sequential data.The approach proposed by Hang *et al.* employs RNN cascade networks [17], which inspired by a 2D CNNs, was extended to its spectral-spatial version.

Inspired by Attention mechanism [18, 19] and residual neural networks, Haut *et al.* [20] introduced attention mechanisms for HSI classification using the ResNet. Xi *et al.* [21] proposed a novel deep prototypical network based on hybrid residual attention. Minghao *et al.* [22], described a residual spectral–spatial attention network (RSSAN) and uses soft attention mechanism. Xue *et al.* proposed a Hierarchical residual network with attention mechanism (HResNetAM) with a double branch structure to extract spectral and spatial features [23]. Xiangrong *et al.* [24] incorporates spectral partitioning in the input for the attention residual network. Furthermore, the SSAN model incorporates a CNN-Attention module that captures spectral-spatial features filtered with an attention module to reduce interference from pixels that can alter the distributions of the data [25].

Some modern architectures consider more complex backbone, combining these architectures, Manifold *et al.* [26] present a versatile architecture based on U-Nnet [27], called U-within-U-Net capable of performing classification, segmentation and prediction in several domains in hyperspectral data. Dalal *et al.* introduced a feature selection method by increasing the variance and filtering data at the input of the entire network, called Compression and Reinforced Variation (CRV). AMM-Net, the Attention Mechanism and Multigroup Strategy [28], which combines a spatial feature extraction module with an attention mechanism and a spectral feature

extraction module with a multi-grouping strategy for spatial and spectral feature extraction, including LSTM, to explore the spectral dimension.

Vaswani et al. [29] proposed a Transformer architecture that utilizes Multihead self-attention (MSA) to relate different positions of an input and assign weights to each part of a sequence. This method has demonstrated notable success in correlating long-range dependencies in natural language processing (NLP) applications. Recently, this approach was extended to image processing by Dosovitskiy et al. [30] and further adapted for hyperspectral imaging (HSI) by Hong et al. [31]. The latter also incorporates cross-layer adaptive fusion (CAF) to facilitate cross-layer feature fusion and preserve the global characteristics. Based on this proposal, several Transformer for HSI have been proposed that consider spectral, spatial, and spectral-spatial representations [32–38].

These methods were originally based on positional encoding (PE) [29], and relative positional encoding was first proposed by Shaw *et al.* [39]. This approach focuses on the pairwise relationships between the elements in the Value and Key matrices, which are represented by the vectors ${a_{i,j}}^V, {a_{i,j}}^K \in \mathbb{R}^{d_z}$. This technique significantly improves the performance of natural language processing (NLP) applications. Cheng-Zhi *et al.* [40] based on Shaw *et al.*, presents an algorithm that reduces their intermediate memory requirement to linear in the sequence length. Subsequently, Yang *et al.* [41] introduced an additional bias for queries and used the sinusoid foundations of the original Transformer paper [29]. For Vision Transformers ViT, this approach is controversial [42]. Kan *et al.* [43] proposed a novel RPE dedicated to 2D images that considers bias and contextual modes, a trainable scalar $r_{i,j} \in \mathbb{R}$ and a trainable vector $r_{i,j} \in \mathbb{R}^{d_z}$, demonstrating that the RPE plays a valuable role in enhancing Vision Transformers. In addition, for Graph Transformers [44], RPE has been successfully proposed by considering trainable bias among Queries, Keys and Values in each part of MSA, thus outperforming previous methods. However, the performance of PE and its variations remain largely unexplored in many settings [42] and have not been explored in HSI.

In summary, since the inception of AI-based methods for hyperspectral image (HSI) classification, numerous approaches have sought to combine spectral signatures with spatial information. However, in domains such as mineral classification, where samples often lack consistent morphological structures, the contribution of spatial features remains debatable. In some cases, incorporating a spatial context may distort intrinsic spectral variability and class boundaries [9], particularly when applied to polished or fragmented mineral samples. Although convolutional neural networks (CNNs) are effective at capturing local spatial patterns, they may be insufficient

for modeling the sequential nature of spectral data. To mitigate this problem, hybrid models that combine CNNs and recurrent networks (RNNs) have been proposed. Recently, transformer architectures have emerged as powerful alternatives capable of modeling both short- and long-range spectral dependencies. Nonetheless, the relevance of spatial features in HSI classification remains an open question that must ultimately be assessed through empirical validation across diverse datasets and tasks. In this context, Transformer-based models offer a promising alternative by focusing exclusively on the spectral domain, while simultaneously capturing both long- and short-range dependencies within the data. These architectures have shown the potential to outperform traditional CNN- and RNN-based approaches and complement their intrinsic advantages, particularly in tasks where the spatial structure does not contribute to the generalization of the underlying class separability.

1.2 Hypothesis

By employing multi-scale feature representation exclusively based on spectral data, we can outperform previous hyperspectral supervised classification models and achieve competitive results compared to state-of-the-art techniques that integrate spectral or spatial-spectral information in terms of accuracy and F1-Score.

1.3 General objective

Development of a supervised artificial intelligence model for mineral identification from hyperspectral reflectance data within the shortwave infrared (SWIR) range.

1.3.1 Specific objectives

1. Design and simulate a multispectral optical sensor for mineral identification using classical machine learning techniques with multi-feature spectral descriptors.
2. Propose a deep learning model based on multifeature approach and spectral embedding levering the hyperspectral dimension for mineral classification.
3. Compare with previous state-of-the-art deep learning models based on spectral and spectral-

spatial representation for mineral classification data, and conventional database to verify its effectiveness and versatility in other domains.

2. Background

2.1 Hyperspectral imaging

Hyperspectral imaging systems are designed to assess the intensity of radiant flux for a specific surface and wavelength, as measured in watts per steradian per square meter (W/srm^2). These systems can detect light emission, reflection, and transmission from objects on a surface as a spectrum of hundreds of channels per surface unit. These channels are then assembled to form a spectral response curve, which can be used to infer material composition based on their spectral and/or spatial-spectral signatures. However, this relationship is not necessarily objective, as different materials may exhibit similar spectral features, and the acquisition conditions or surface effects can further introduce uncertainty into the identification process. The adaptability of HSI systems makes them indispensable resources for a diverse range of applications including mineral exploration, ecological monitoring, and agricultural analysis [45, 46].

A hyperspectral image that includes spectral-spatial information can be represented as a three-dimensional tensor with dimensions h (height), w (width), and d (channels), denoted as the data cube X $\in \mathbb{R}^{h \times w \times d}$. Each channel value is the result of the acquisition system and can be represented by Equation 2.1, where $I_R(\lambda)$ represents the light source over the sample target, as observed by the camera; $r(\lambda)$ is the light reflected, transmitted, or emitted by the target; $t(\lambda)$ is the optical system of the camera; $f_d(\lambda)$ is the spectral channel; $\alpha_d(\lambda)$ is the CCD array response; and ϵ_d is the additive noise (ec. 2.1). These functions were combined to form the spectral sensitivity function $\omega_d = I_R(\lambda)t(\lambda)f_d(\lambda)\alpha_d(\lambda)$ (ec. 2.2).

$$c_d = \int_{\lambda_{min}}^{\lambda_{max}} I_R(\lambda)r(\lambda)t(\lambda)f_d(\lambda)\alpha(\lambda)d\lambda + \epsilon_d \tag{2.1}$$

$$= \int_{\lambda_{min}}^{\lambda_{max}} r(\lambda)\omega_d d\lambda + \epsilon_d \tag{2.2}$$

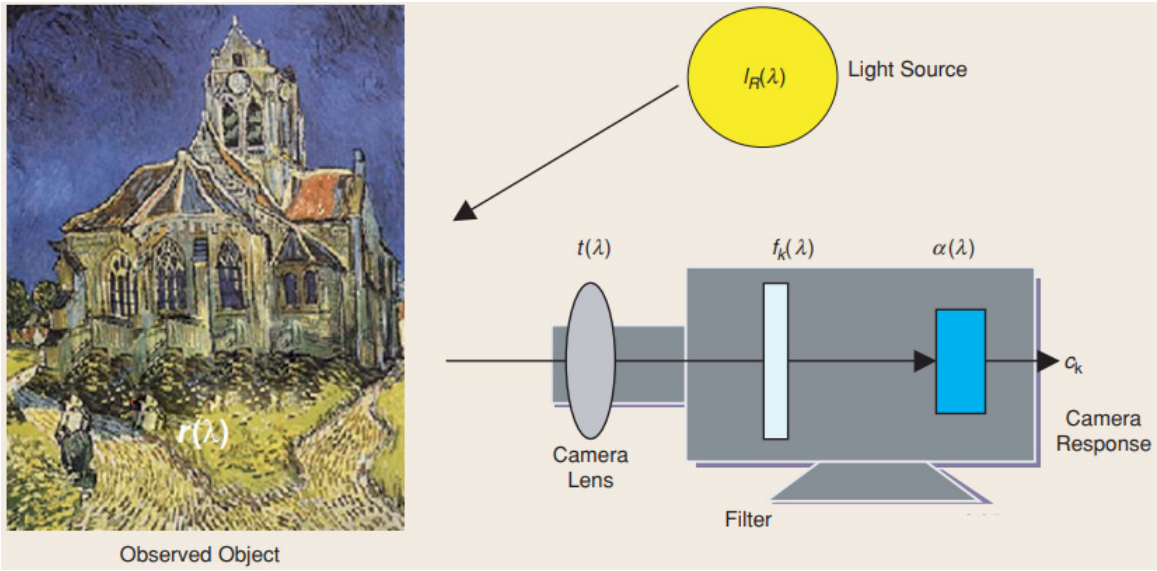

Fig. 2.1: Scheme of a optoelectronic imaging system. Light from the source $I_R(\lambda)$ interacts with the object and passes through the optical system $t(\lambda)$ and spectral filter $f_d(\lambda)$ before being recorded by the CCD sensor $\alpha_d(\lambda)$, forming the observed spectral response as described in Equation 2.2, extracted from [47].

2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a linear transformation that enables the modeling of multivariate data in a new reference system. The primary goal is to reduce the dimensionality of the dataset while preserving the maximum variance in an unsupervised manner. This is accomplished by transforming the data into a new set of orthogonal variables known as principal components (PC). These components were arranged such that the first component captured most of the variation present in the entire dataset. Note that the sum of the variances of all principal components is one.

The eigenvalue-eigenvector method and singular value decomposition (SVD) are the most commonly utilized techniques for determining the principal components. These methods can operate on either the correlation matrix **R** or the covariance matrix **C**.

The covariance matrix $C_{m\times n} \in X_{m\times n}$ can be defined as (2.4)

$$\begin{aligned} C = [c_{i,j}] &= cov(X_i, X_j) \\ i &= 1, 2, \cdots, m \\ j &= 1, 2, \cdots, m \end{aligned} \tag{2.3}$$

The correlation matrix $R_{m\times n} \in X_{m\times n}$ can be define as (2.4):

$$\begin{aligned} R = [r_{i,j}] &= \frac{cov(X_i, X_j)}{\sqrt{var(X_i)var(X_j)}} \\ i &= 1, 2, \cdots, m \\ j &= 1, 2, \cdots, m \end{aligned} \tag{2.4}$$

where $\boldsymbol{X_i}$ denotes the ith column of training matrix. The computation of the correlation matrix is equivalent to obtaining components with standardized variables, because the correlation is normalized by standard deviations, as demonstrated by equation (2.4).

The optimization problem is to find matrices $\boldsymbol{U}$ (score) and $\boldsymbol{V}$ (loadings) that minimize the Frobenius norm of the difference between the original matrix $\boldsymbol{X}$ and its approximation $\boldsymbol{UV^T}$, given a dataset $D = \{\boldsymbol{x}_i\}_{i=1}^N$. Loadings represent the weights assigned to each original variable (feature) associated with the principal component.

$$\arg\min_{U,V} ||X - UV^T||_F^2 \tag{2.5}$$

where $||\cdot||_F$ denotes the Frobenius norm, which is the square root of the sum of squares of the elements in the matrix.

2.2.1 Eigenvectors

Given a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$, $\lambda \in \mathbb{R}$ is an eigenvalue of $\boldsymbol{X}$ and $\mathbf{v} \in \mathbb{R}^n$ is the corresponding eigenvector.

$$(\mathbf{X} - \lambda_i \mathbf{I}) \cdot \mathbf{v}_i = 0 \text{ , i} = 1 \text{ ... n, } v \neq 0 \tag{2.6}$$

2.2.2 Singular Value Decomposition

In the context of Singular Value Decomposition (SVD), a data matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$, where N is the number of observations and D is the number of variables, can be factorized into three distinct matrices:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}, \tag{2.7}$$

where $\boldsymbol{U} \in \mathbb{R}^{N\times N}$ is an orthogonal matrix whose columns are known as the left singular vectors, $\boldsymbol{V} \in \mathbb{R}^{D\times D}$ is an orthogonal matrix composed of the right singular vectors, and $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times D}$ is a rectangular diagonal matrix whose nonnegative diagonal elements are singular values that are typically arranged in descending order.

In practice, the data matrix $\mathbf{X}$ is often centered (i.e., column-wise mean subtracted) and sometimes normalized by the factor $1/\sqrt{N-1}$, yielding the modified decomposition

$$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} = \frac{1}{\sqrt{N-1}}\boldsymbol{X}. \tag{2.8}$$

This formulation is particularly useful when interpreting SVD in the context of Principal Component Analysis (PCA). Specifically, the product $\boldsymbol{U}\boldsymbol{\Sigma}$ yields the principal component scores, which represent the projection of the original data onto the new orthogonal basis defined by the singular vectors. The columns of $\boldsymbol{V}$ represent the principal directions or loadings that describe the orientations of the principal axes in the original feature space.

SVD provides an efficient framework for dimensionality reduction by retaining only the first k largest singular values and their corresponding vectors. The original matrix $\mathbf{X}$ can be approximated with a reduced rank, preserving most of the variance in the data while significantly reducing the computational cost. This property makes SVD a fundamental technique for signal processing, image compression, and unsupervised learning.

2.2.3 Sparse Principal Component Analysis

Sparse Principal Component Analysis (sPCA) is a variant of classical PCA that enforces sparsity in the principal components or loadings. Whereas traditional PCA results in dense components (i.e., each component is a linear combination of all the original variables), sPCA allows the loadings to be exactly zero. This is particularly useful when working with high-dimensional datasets in which only a subset of variables is relevant for explaining the main variance directions.

Mathematically, sPCA introduces an L_1 regularization term into the PCA objective function, which is controlled by the sparsity-inducing hyperparameter $\alpha \in \mathbb{R}^{+}$. The optimization problem

can be formulated as

$$\arg\min_{U,V} \|X - UV^\top\|_F^2 + \alpha\|V\|_{1,1}, \tag{2.9}$$

where:

- $X \in \mathbb{R}^{N\times D}$ is the data matrix,
- $U \in \mathbb{R}^{N\times k}$ contains the low-dimensional representation of the data (the principal components),
- $V \in \mathbb{R}^{D\times k}$ contains the sparse loading vectors,
- $\|\cdot\|_F$ denotes the Frobenius norm,
- and $\|\cdot\|_{1,1}$ represents the sum of absolute values of all elements in V (i.e., the ℓ_1 norm).

The addition of the ℓ_1 penalty promotes sparsity in V, effectively performing variable selection during the dimensionality reduction. This allows sPCA to identify and focus on the most informative features in the dataset, which is particularly advantageous when the number of variables significantly exceeds the number of observations ($D \gg N$), which is a common scenario in hyperspectral imaging.

Furthermore, sPCA can enhance generalization in subsequent tasks such as classification or clustering by mitigating noise and redundancy through the exclusion of irrelevant variables. The hyperparameter α is selected empirically; higher values of α increase the sparsity, potentially compromising the reconstruction accuracy. Consequently, in supervised settings, α is typically determined via cross-validation, informed by ground-truth labels, to optimize performance metrics such as classification accuracy or F1-score. In unsupervised contexts, alternative criteria, such as explained variance or stability across data partitions, may be employed to guide model selection.

2.3 Artificial neural networks

Artificial neural networks (ANNs) are computational models inspired by the structure and function of biological neural systems designed for tasks in machine learning. These networks employ calculation units that emulate the behavior of human neurons, where each neuron corresponds to parameter W of the model. The parameters are iteratively adjusted using a process called backpropagation, which involves minimizing an objective function known as loss based on the input data. The fundamental architecture of an ANN is a perceptron that is mathematically represented by Equation 2.10.

$$\hat{y} = b + \sum_{j=1}^{d} \boldsymbol{w}_j \boldsymbol{x}_j \tag{2.10}$$

Where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_d]$ represents the model parameters, b denotes the bias term, d denotes the number of features, and $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_d]$ denotes the input vector of the model.

Deep neural networks exhibit distinct characteristics, including the presence of hidden layers that contribute to the increased number of parameters in the model. Moreover, each layer incorporates either a linear or a nonlinear activation function. In the realm of probability or classification, the softmax function was utilized as the activation function in the output layer (Eq. 2.11).

$$\text{softmax}(v_i) = \frac{e^{v_i}}{\sum_{j=1}^{k} e^{v_j}}, \forall i \in [1, ..., k] \tag{2.11}$$

where v_i represents the ith element of the output vector v and k denotes the total number of elements in v. Thus, we can only obtain values greater than or equal to zero in terms of the probabilities.

Another fundamental aspect of ANNs is the cost function (or objective), which must be optimized based on a specific application. In the case of categorical variables, cross-entropy is commonly employed as the cost function, defined as

$$\mathcal{L}_{ce} = -\sum_{i} y_{\text{true}}(i) \log(y_{\text{pred}}(i)) \tag{2.12}$$

where y_{true} represents the true data and y_{pred} represents the data predicted by the model. Finally, this objective function can be optimized using various algorithms, such as gradient descent, Adaptive Moment Estimation (Adam), and Adam with Weight Decay (AdamW).

2.3.1 Feed Forward Networks

A feedforward neural network is composed of multiple interconnected layers including an input layer, one or more hidden layers, and an output layer. The network operation follows a sequential process, with data passing through these layers without feedback connections.

The feedforward process in a network with l layers can be described as follows.

1. Input Layer:
$$a^{(0)} = x \tag{2.13}$$

2. Hidden Layers:
$$z^{(i)} = w^{(i)}a^{(i-1)} + b^{(i)} \tag{2.14}$$

$$a^{(l)} = g(z^{(l)}), i = 1, \ldots, l \tag{2.15}$$

where $g(\cdot)$ denotes an activation function.

3. Output layer:
$$z^{(l)} = w^{(l)}a^{(l-1)} + b^{(l)} \tag{2.16}$$

$$\hat{y} = \mathcal{FFN}(x) = a^{(l)} = g(z^{(l)}) \tag{2.17}$$

where $\hat{y}$ denotes the prediction based on input vector $\mathbf{x}$.

2.3.2 Layer Normalization

Layer normalization is an essential aspect of deep learning. Specifically, it ensures that each input within a batch is normalized across all **features**. This not only facilitated the training process but also enhanced the overall performance of the neural network. Layer normalization

plays a pivotal role in mitigating the effects of the vanishing and exploding gradients, thereby enabling the network to learn more effectively. Given l^{th} hidden layer and i^{th} hidden units [48]:

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} X_i^l \tag{2.18}$$

$$\sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(X_i^l - \mu^l)^2 + \epsilon} \tag{2.19}$$

$$\hat{X}_i = \frac{X_i^l - \mu^l}{\sigma^l} \tag{2.20}$$

$$Y_i = \mathcal{LN}(X_i) = \gamma\hat{X}_i + \beta \quad \text{(scale } (\gamma) \text{ and shift } (\beta)) \tag{2.21}$$

2.3.3 Residual Neural Networks

A Residual Neural Network (ResNet) [49], is a deep learning model that uses skip and residual functions. Specifically, the ResNet architecture involves learning residual functions relative to layer inputs, which are subsequently combined with layer outputs. This approach presents an additional advantage for mitigating the vanishing gradient problem to a certain extent, and adds global features that can vanish during training. Some variants use different types of skip blocks such as identities and CNN. Given an underlying mapping $\mathcal{H}(\boldsymbol{X})$, this can be formulated as

$$\mathcal{F}(\boldsymbol{X}) = \mathcal{H}(\boldsymbol{X}) - \boldsymbol{X} \tag{2.22}$$

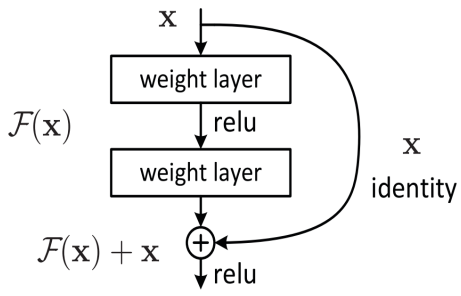

Fig. 2.2: Residual learning block from the ResNet architecture [49]. This structure enables identity mapping via shortcut connections, allowing the network to learn the residual functions $\mathcal{F}(x) + x$ instead of direct mapping. This formulation facilitates the training of very deep networks by mitigating the vanishing gradient problem.

2.4 Convolutional Neural Networks

Convolution is a mathematical operation that is extensively utilized in image processing and is a vital component of Convolutional Neural Networks (CNNs). The primary purpose is to map an input image using a set of kernels to extract features. Each filter or kernel was represented by a matrix of numerical values applied to the input image. Convolution involves element-wise multiplication between the kernel and the region of the image, followed by summation of the resulting products. This iterative process is performed across the entire image, yielding a new matrix of values, referred to as a feature map.

Mathematically, the convolution between the images I and K can be expressed as follows:

$$(I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n) \tag{2.23}$$

where (i, j) denotes the coordinates of the pixel, (m, n) represents the coordinates of the pixel in the kernel, and $I(i-m, j-n)$ and $K(m, n)$ represent the intensity values of the corresponding pixels in the image and filter, respectively, [50].

In a CNN, convolution is applied across multiple layers, each of which contains a training kernel. These kernels learn to recognize visual features, such as edges and textures, as they adjust during the training process. In addition, CNNs incorporate nonlinear activation layers such as rectified linear units (ReLUs) and pooling layers.

After obtaining the features from the CNN, they were connected to a fully connected neural network for the discriminative tasks. In the final layer, a Softmax function is typically used to predict the probability of each class.

2.5 Recurrent Neural Networks

When working with sequential input and output data, such as natural language processing, time-series analysis, and image captioning, recurrent neural networks (RNNs) are often employed as essential components. The defining characteristic of RNNs is their hidden state, which retains significant information about the sequence and the previous state. However, the initial state is often susceptible to vanishing gradients within the network. To address this issue,

bidirectional RNNs, which require initial states to rely on the final states of the network, have gained prominence. Nevertheless, this type of network can be challenging to train owing to its susceptibility to vanishing and exploding gradients, which can be attributed to its large number of parameters. Mathematically, the RNN can be described as

$$\boldsymbol{h}_t = \varphi(\boldsymbol{W}_{hx}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \tag{2.24}$$

where φ represents the activation function, the hidden state h_t is the input x_t, y_t is the output, W_{hx} is the weight matrix for the input, W_{hh} is the weight matrix for the hidden state, W_{yh} is the weight matrix for the output, b_h is the bias for the hidden state, and b_y is the output bias.

In a bidirectional recurrent neural network (BRNN), the hidden state is incorporated in parallel to retain information, and the objective is to divide the state neurons of a standard RNN into two parts: one responsible for the positive time direction $\overrightarrow{h_t}$ (forward states) and the other for the negative time direction $\overleftarrow{h_t}$ (backward states). It is important to note that outputs from the forward states should not be connected to the inputs of the backward states and vice versa.

$$\overrightarrow{\boldsymbol{h}_t} = \varphi(\overrightarrow{\boldsymbol{W}}_{hx}\boldsymbol{x}_t + \overrightarrow{\boldsymbol{W}}_{hh}\boldsymbol{h}_{t-1} + \overrightarrow{\boldsymbol{b}_h}) \tag{2.25}$$

$$\overleftarrow{\boldsymbol{h}_t} = \varphi(\overleftarrow{\boldsymbol{W}}_{hx}\boldsymbol{x}_t + \overleftarrow{\boldsymbol{W}}_{hh}\boldsymbol{h}_{t+1} + \overleftarrow{\boldsymbol{b}}_h) \tag{2.26}$$

$$\boldsymbol{h}_t = \left[\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}\right] \tag{2.27}$$

2.5.1 Gated recurrent units (GRU)

This type of RNN introduces the concept of learning when updating the hidden state using a gating unit. Given a matrix $\boldsymbol{X}_t \in \mathbb{R}^{N\times D}$, where N is the batch size and D is the vocabulary size, which is similar to the hidden state, $\boldsymbol{H}_t$ is an NXH matrix, where H is the number of

hidden states. The reset gate $\boldsymbol{R}_t \in \mathbb{R}^{N\times H}$ and update gate $\boldsymbol{Z}_t \in \mathbb{R}^{N\times H}$, and $\odot$ is the inner dot product:

$$\boldsymbol{R}_t = \boldsymbol{\sigma}(\boldsymbol{X}_t\boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1}W_{hr} + \boldsymbol{b}_r) \tag{2.28}$$

$$\boldsymbol{Z}_t = \boldsymbol{\sigma}(\boldsymbol{X}_t\boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1}W_{hz} + \boldsymbol{b}_z) \tag{2.29}$$

Given this, we define the next state vector as

$$\hat{\boldsymbol{H}}_t = \tanh\left(\boldsymbol{X}_t\boldsymbol{W}_{xh} + (\boldsymbol{R}_t \odot \boldsymbol{H}_t - 1)W_{hh} + b_h\right) \tag{2.30}$$

Finally, the current hidden state is

$$\boldsymbol{H}_t = \boldsymbol{Z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{Z}_t) \odot \hat{\boldsymbol{H}}_t \tag{2.31}$$

2.5.2 Long short term-memory LSTM

LSTM modules contain compute blocks that control the flow of information, and are composed of four steps:

1. Forget: Irrelevant part of the previous state

$$\boldsymbol{f}_t = \boldsymbol{\sigma}(\boldsymbol{W}_{xf}\boldsymbol{x}_t + \boldsymbol{b}_{if} + \boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{2.32}$$

2. Store: Relevant new information into the cell state

$$\boldsymbol{g}_t = \tanh(\boldsymbol{W}_{xg}\boldsymbol{x}_t + \boldsymbol{b}_{ig} + \boldsymbol{W}_{hg}\boldsymbol{h}_{t-1} + \boldsymbol{b}_g) \tag{2.33}$$

3. Update: Previous state

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t \tag{2.34}$$

4. Output:

$$\boldsymbol{y}_t = \boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \tag{2.35}$$

Note that the tanh activation function is used to constrain the candidate cell values g_t and c_t within the range $[-1, 1]$, allowing LSTM to model both positive and negative contributions. This bounded and centered output facilitates stable gradient propagation during training, particularly when combined with σ-based gating mechanisms in other components of the LSTM.

2.6 Transformers

The Transformer architecture was introduced by Vaswani et al. [29]. A crucial component of this architecture is the multihead self-attention and positional encoding, which are described in detail in this section. A visual representation of this architecture is shown in Fig. 2.3, and the mathematical formulation of each component will be presented in detail in the following subsections.

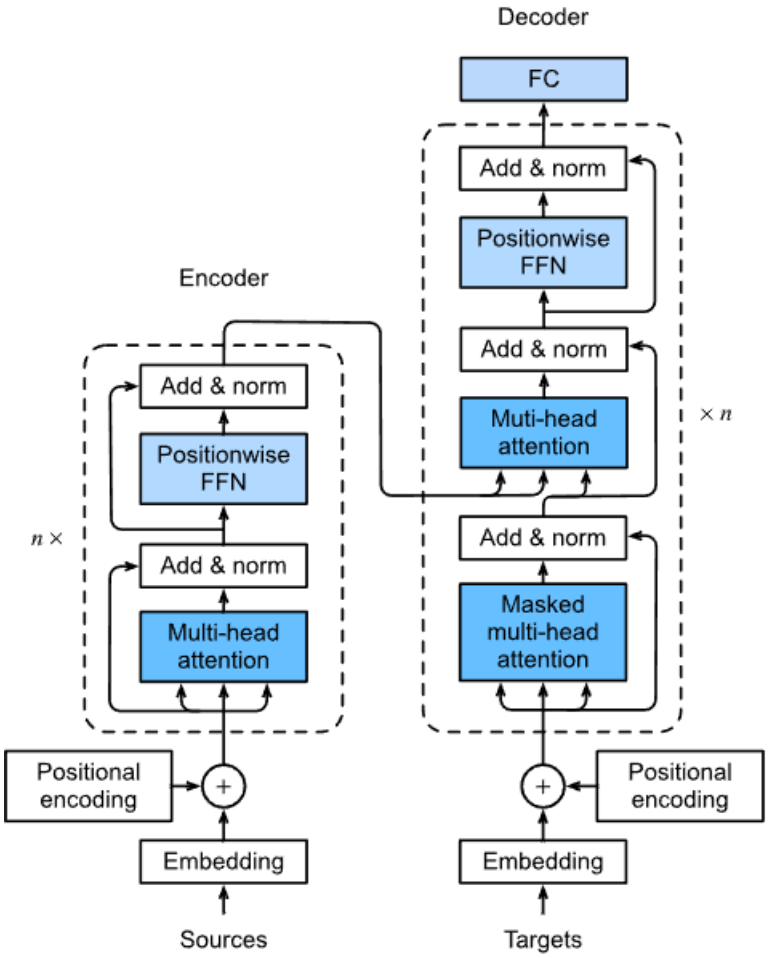

Fig. 2.3: The Transformer model, consisting of stacked encoder and decoder layers with residual connections and layer normalization. Each encoder layer includes a multi-head self-attention mechanism and a position-wise feed-forward network. The decoder incorporates an additional encoder–decoder attention sublayer, and uses masked self-attention to preserve the autoregressive property during generation, extracted from [51].

2.6.1 Scaled Dot-Product attention

The scaled dot-product attention owing to an input $\boldsymbol{x} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})$, $\boldsymbol{x_i} \in \mathbb{R}^{d_x}$, and the output $\boldsymbol{z}_i \in \mathbb{R}^{d_z}$ is given by

$$z_i = \sum_{j=1}^{L} \alpha_{ij}(X_j W^V) \tag{2.36}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{L} \exp(e_{ik})} \tag{2.37}$$

$$e_{ij} = \frac{(X_i W^Q)(X_j W^K)^T}{\sqrt{d_z}} \tag{2.38}$$

where $\boldsymbol{W^Q}, \boldsymbol{W^K}, \boldsymbol{W^V} \in \mathbb{R}^{d_x \times d_z}$ are trainable matrices that can also be represented by

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{2.39}$$

where the matrix query $\boldsymbol{Q} \in \mathbb{R}^{d_z \times d_q}$, the key $\boldsymbol{K} \in \mathbb{R}^{d_z \times d_k}$ and $\boldsymbol{V} \in \mathbb{R}^{d_z \times d_v}$.

2.6.2 Multihead Self-Attention

To capture the different notions of similarity, the i^{th} attention head can be described by

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2.40}$$

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, ..., head_h)W^0 \tag{2.41}$$

where $\boldsymbol{W}_i{}^Q \in \mathbb{R}^{d_{model} \times d_q}$, key $\boldsymbol{W}_i{}^K \in \mathbb{R}^{d_{model} \times d_k}$, and value $\boldsymbol{W}_i{}^V \in \mathbb{R}^{d_{model} \times d_v}$, and $\boldsymbol{W^0} \in \mathbb{R}^{hd_v \times d_{model}}$. Note that originally was used $h = 8$ and $d_q = d_k = d_v = \frac{d_{model}}{h} = 64$.

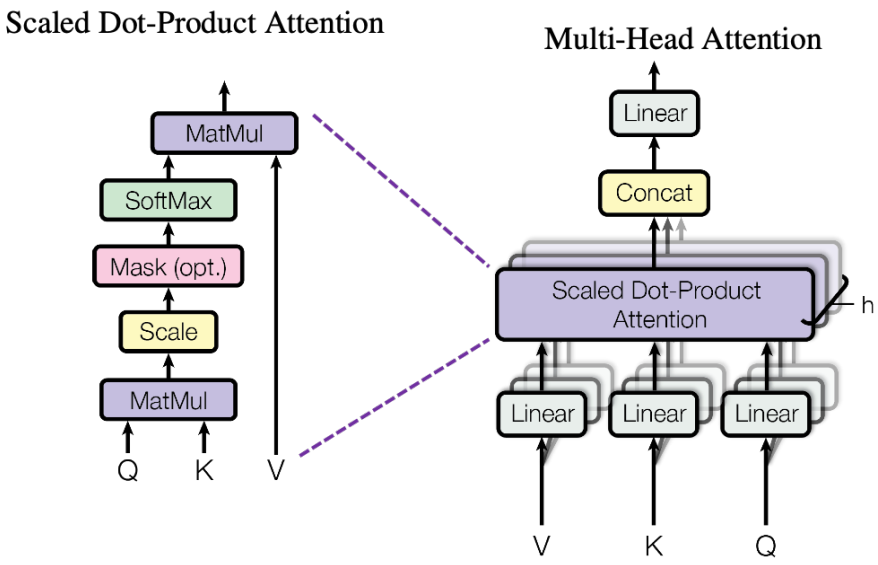

Fig. 2.4: Illustration of Multi-Head Attention, where multiple attention heads operate in parallel to capture diverse contextual relationships, followed by concatenation and a linear projection, extracted from [29].

2.6.3 Positional Encoding

It is worth noting that attention exhibits a permutation-invariant characteristic that renders it insensitive to the arrangement of input elements. To address this limitation, absolute nontrainable positional encoding has been proposed [29].

Given $\boldsymbol{x}_i \in \mathbb{R}^L$ and $PE \in \mathbb{R}^{L \times d_{pe}}$ and using a sinusoidal basis:

$$p_{i,2j} = \sin\left(\frac{i}{C^{2j/d_{pe}}}\right) \tag{2.42}$$

$$p_{i,2j+1} = \cos\left(\frac{i}{C^{2j/d_{pe}}}\right) \tag{2.43}$$

where C is the maximum sequence length and $j \in \{0, \ldots, d_{pe} - 1\}$

2.6.4 SpectralFormer: Transformer-Based Architecture for Hyperspectral Image Classification

SpectralFormer is a transformer-based architecture introduced by Danfeng Hong *et al.* [31] and specifically designed for hyperspectral image (HSI) classification. In contrast to conventional methods that treat spectral signatures as linear sequences, SpectralFormer incorporates two principal modules, *bandwise spectral embedding* (BSE), *Group-wise Spectral Embedding* (GSE) and *Cross-layer Adaptive Fusion* (CAF), to enhance spectral representation and facilitate hierarchical information flow. The architecture can be employed in many ways, such as a pixel or patch-wise spectral band, as well with Cross-Layer Adaptive Fusion.This flexibility allows the architecture to be applied to various tasks, such as considering only the spectral dimension or both the spectral and spatial dimensions.

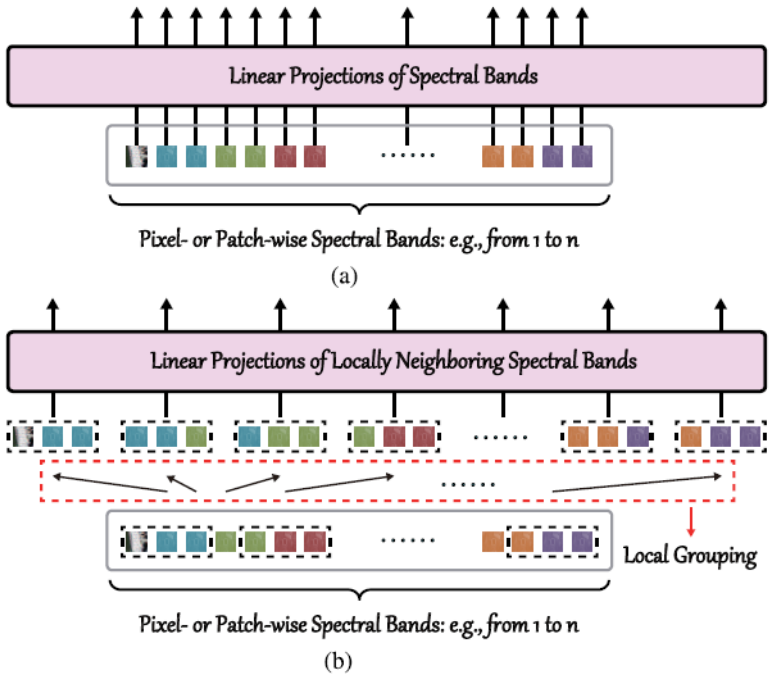

Fig. 2.5: Representation of (a) bandwise spectral embedding and (b) groupwise spectral embedding, extracted from [31].

CAF addresses the challenge of insufficient information exchange across layers of differing abstraction levels, which is a known limitation in both shallow and deep architectures when modeling complex spectral patterns. Residual connections or skip connections are widely used to mitigate information loss across layers. However, direct connections between distant layers (i.e., long SCs) may lead to semantic gaps between the shallow and deep features.

Given $\mathbf{z}^{(l-2)} \in \mathbb{R}^{1\times d_z}$ and $\mathbf{z}^{(l)} \in \mathbb{R}^{1\times d_z}$ are the feature representations extracted from the $(l-2)$th and lth layers, respectively. The fusion output $\hat{\mathbf{z}}^{(l)}$ in the l-th layer is computed as a weighted combination of both representations as follows:

$$\hat{\mathbf{z}}^{(l)} \leftarrow \ddot{\mathbf{w}} \begin{bmatrix} \mathbf{z}^{(l)} \\ \mathbf{z}^{(l-2)} \end{bmatrix} \tag{2.44}$$

where $\ddot{\mathbf{w}} \in \mathbb{R}^{1\times 2}$ denotes a trainable parameter vector that adaptively learns optimal fusion weights for each feature source. This operation enables the network to dynamically determine the relative importance of high- and low-level features during training.

In practical terms, the CAF mechanism is applied selectively; only a single fusion connection is introduced between nonadjacent encoder blocks, rather than fusing across all layers. Nevertheless, the application of multiple CAF modules may introduce unnecessary complexity without proportional benefit. Hence, a single adaptive fusion point strikes a balance between the model expressiveness and computational efficiency.

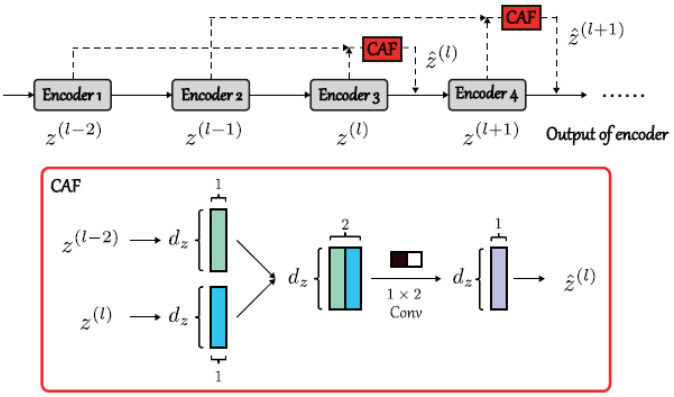

Fig. 2.6: Cross-layer adaptive fusion, extracted from [31].

$$\hat{y} = \text{softmax}(\text{MLP}(\mathbf{z}_{\text{class}}^{(L)})) \tag{2.45}$$

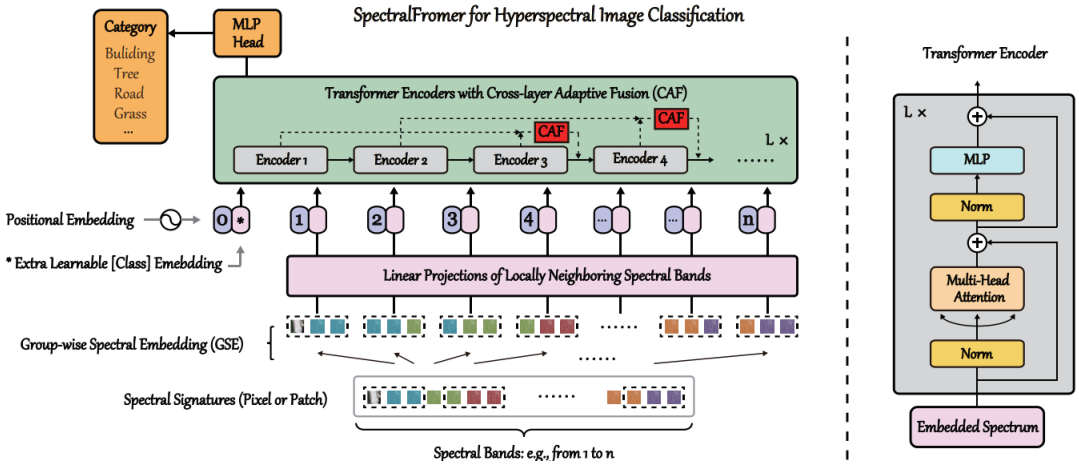

Fig. 2.7: The SpectralFormer arquitecture, combines groupwise spectral embedding (GSE), cross-layer adaptive fusion (CAF), and global context modeling via self-attention, SpectralFormer effectively captures both local and global spectral patterns. This hybrid design leads to state-of-the-art performance in hyperspectral image classification tasks, extracted from [31].

3. Database description

3.1 Lithology and mineral dataset

These two datasets are presented in this section. The first dataset was obtained from a volcanogenic massive sulfide deposit and cross-validated by field geologists and portable X-ray fluorescence (pXRF) measurements. This dataset, provided by the University of Liége, was acquired using a SPECIM®SWIR camera, which offers spatial resolutions of 320 and 256 spectral bands ranging from 900 to 2500 nm, employing the pushbroom technique. The main goal of using this dataset is only attributable to specific objective 1 owing to the limitations of computational resources and practical field constraints. The obtained data comprised nine classes, as illustrated in Fig.3.1 and summarize in Table 3.1. The lithologies detailed in Table 3.1 collectively depict a sedimentary-volcaniclastic depositional environment with a notable volcanic contribution. Such environments are typical of the development of volcanogenic massive sulfide (VMS) deposits and other zones rich in metals. The presence of pyroclastic rocks, volcaniclastics, and chemical-sedimentary facies suggests a dynamic system marked by alternating volcanic events and sediment deposition.

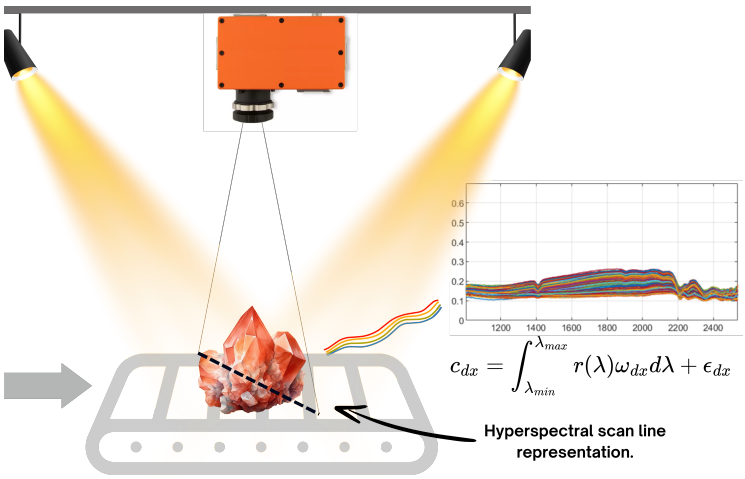

Fig. 3.1: Diagram of a pushbroom hyperspectral imaging system, which acquires data line-by-line to form a hyperspectral datacube $\boldsymbol{X} \in \mathbb{R}^{h \times w \times d_x}$, where h and w are spatial dimensions and d_x is the number of spectral bands. Each scanned line captures a full spectrum per pixel, enabling detailed characterization.

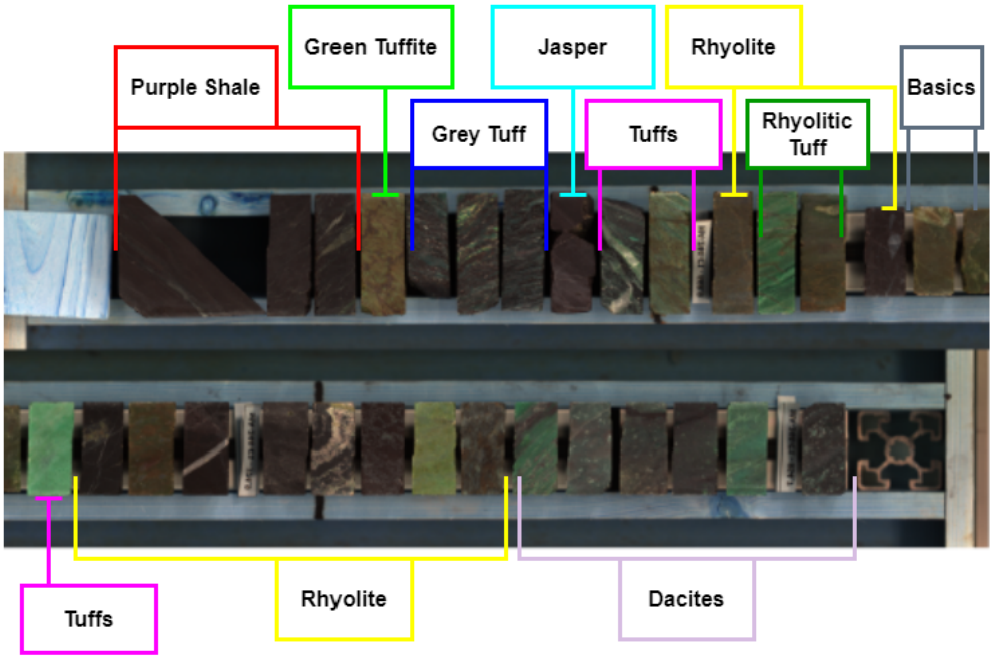

Fig. 3.2: RGB representation of HSIs polished samples of nine representative lithologies used for specular reflectance measurements under controlled conditions (dataset 1). The samples include igneous, sedimentary, and volcaniclastic rocks, such as rhyolite, purple shale, gray tuff, green tuffite, and jasper. Polishing enhances surface uniformity, minimizing diffuse scattering and allowing consistent spectral characterization of mineralogical features.

Label	Class	Train samples	Test samples
0	Purple Shale	8000	4000
1	Green Tuffite	4000	2000
2	Gray Tuff	12000	6000
3	Jasper	4000	2000
4	Tuffs	12000	6000
5	Rhyolite	40000	20000
6	Rhyolite Tuff	8000	4000
7	Basics	8000	4000
8	Dactiles	24000	12000

Table 3.1: Train and test sames of dataset 1

The second dataset used was "A 2D hyperspectral library of mineral reflectance from 900 to 2500 nm [11] for deep learning experiments using a calibrated mask dataset (dataset 2), which contains reflectance images of 130 samples of 76 pure minerals.

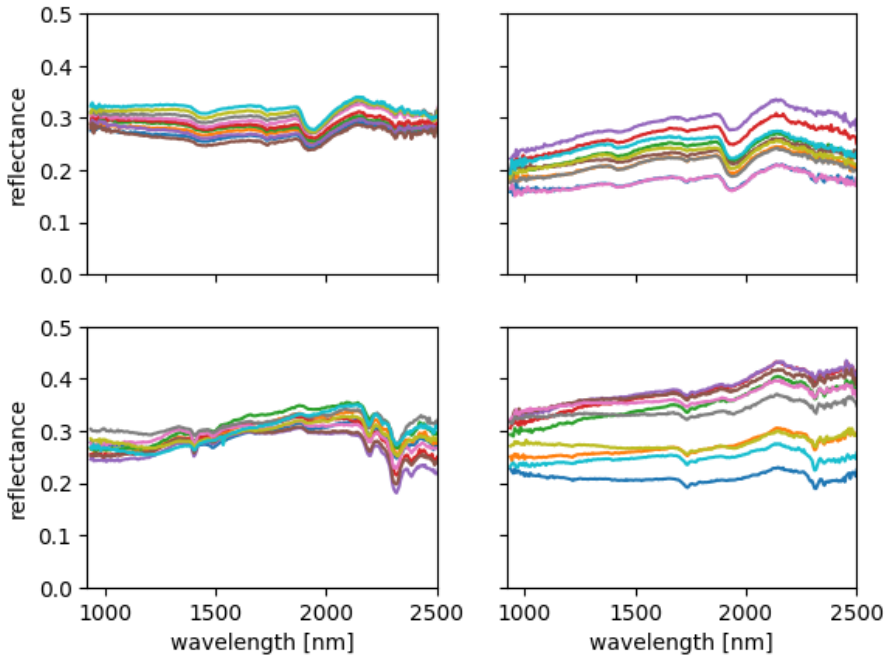

Fig. 3.3: Subset of SWIR reflectance spectra corresponding to four representative minerals extracted from Dataset 2. Celestite (top left), calcite (top right), actinolite (bottom left), and talc (bottom right). The layout displayed spectral variability within and between classes, highlighting the distinct spectral features used for discrimination.

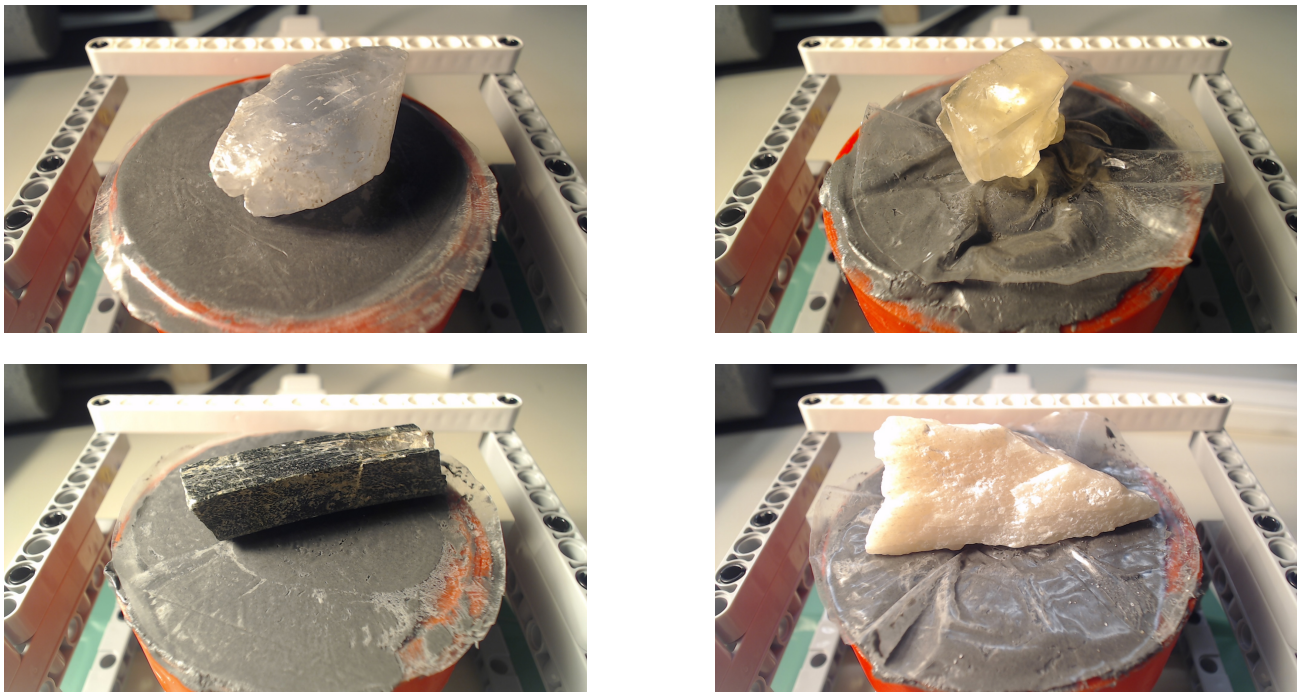

Fig. 3.4: RGB images of minerals samples: Celestite (top left), calcite (top right), actinolite (bottom left), and talc (bottom right).

3.2 Standard dataset for hyperspectral classification

In addition, the proposed method was evaluated using the widely recognized datasets, "Indian Pines" and "Houston2013," employing standard training and testing samples. These datasets were obtained from: `https://github.com/danfenghong/IEEE_TGRS_SpectralFormer`. The "Indian Pines" dataset was collected using an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor that captures agricultural and forested areas in northwestern Indiana, USA. This dataset comprises 145×145 pixels and 220 spectral bands ranging from 400 to 2500 nm. However, bands affected by water absorption and noise were typically excluded, resulting in 200 usable bands. The scene includes 16 land cover classes, primarily agricultural crops, such as corn, soybeans, and alfalfa, as well as non-vegetated areas, such as roads and buildings. In contrast, the Houston 2013 dataset was acquired using the Compact Airborne Spectrographic Imager (CASI) over the University of Houston campus and adjacent urban areas. It consists of 349×1905 pixels and 144 spectral bands covering the 380–1050 nm range. Unlike Indian Pines, Houston 2013 encompasses complex urban features, including buildings, roads, parking lots, vegetation, and water bodies, making it ideal for urban land-cover classification tasks. This dataset was released as part of the 2013 IEEE GRSS Data Fusion Contest. Detailed descriptions are provided in Tables 3.2 and 3.3. For both datasets, standard ground-truth annotations and labeled splits for training and testing, as defined in the literature, were utilized to ensure a fair performance comparison across different models. These standardized partitions facilitate reproducibility and enable direct benchmarking against state-of-the-art hyperspectral image-classification methods. These standardized datasets enhance reproducibility and facilitate equitable comparisons across models. In contrast to the lithological dataset, the Indian Pines and Houston2013 datasets exhibit more pronounced spectral and spatial contrasts, offering a complementary evaluation framework. Their inclusion highlighted the capacity of the model to generalize across diverse hyperspectral domains.

label	Class	Train samples	Test samples
0	Corn Notill	50	1384
1	Corn Mintill	50	784
2	Corn	50	184
3	Grass Pasture	50	447
4	Grass Trees	50	697
5	Hay Windrowed	50	439
6	Soybean Notill	50	918
7	Soybean Mintill	50	2418
8	Soybean Clean	50	564
9	Wheat	50	162
10	Woods	50	1244
11	Buildings Grass Trees Drives	50	330
12	Stone Steel Towers	50	45
13	Alfalfa	15	39
14	Grass Pasture Mowed	15	11
15	Oats	15	5
Total		**695**	**9671**

Table 3.2: Training and testing samples for Indian Pines dataset.

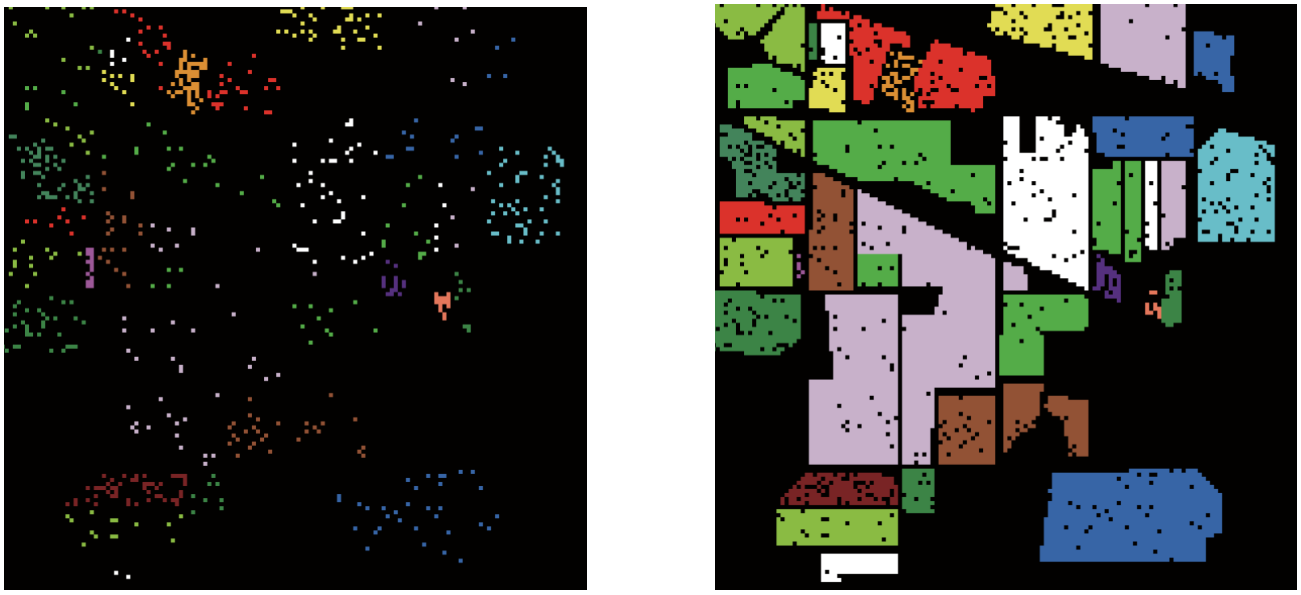

Fig. 3.5: Visual representation of labeled training and testing samples in the Indian Pines hyperspectral dataset.

Class No.	Class Name	Training	Testing
1	Healthy Grass	198	1053
2	Stressed Grass	190	1064
3	Synthetic Grass	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot1	192	1040
13	Parking Lot2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
Total		**2832**	**12197**

Table 3.3: Training and testing samples for Houston2013 dataset.

Fig. 3.6: Annotated representation of labeled training and testing samples in Houston2013 hyperspectral dataset.

4. Multispectral data-driven sensor design and classification performance

4.1 Data cleaning and prepossessing

Given a hyperspectral image cube denoted by $\boldsymbol{I} \in \mathbb{R}^{h \times w \times d_x}$, where h and w represent the spatial dimensions (height and width, respectively) and d_x corresponds to the number of spectral bands, the data can be restructured to emphasize spectral information. Specifically, the cube is reshaped into a two-dimensional matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{hw}] \in \mathbb{R}^{hw \times d_x}$, where each row vector $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ corresponds to the spectral signature of an individual pixel in the spatial domain.

Z-score normalization was applied to matrix $boldsymbol X$ to standardize the spectral signatures and mitigate the influence of scale differences among spectral bands, the Z-score normalization was applied to the matrix $\boldsymbol{X}$. This normalization procedure, formally defined in Equation 4.1, centers on each spectral band by subtracting its mean and scaling it by its standard deviation, thereby ensuring that each band exhibits a zero mean and unit variance across all spatial locations. This step is crucial for enhancing the comparability of spectral features and improving the numerical stability of subsequent machine learning models.

$$\text{Z-score}(\boldsymbol{x_i}) = \frac{\boldsymbol{x}_i - \mu_i}{\sigma_i} \tag{4.1}$$

where:

$$\mu_i = \frac{1}{hw} \sum_{j=1}^{hw} \boldsymbol{X_{ij}} \tag{4.2}$$

$$\sigma_i = \sqrt{\frac{1}{hw - 1} \sum_{j=1}^{hw} |\boldsymbol{X}_{ij} - \mu_i|^2} \tag{4.3}$$

4.2 Multispectral Sensor Design and Simulation

To develop a compact and computationally efficient multispectral sensing system, particularly tailored for the short-wave infrared (SWIR) region of the electromagnetic spectrum, we adopted and extended the methodology initially introduced by Calvini *et al.* [52]. This approach provides a principled framework for selecting a reduced subset of wavelengths that retains the most relevant spectral information while significantly reducing the data dimensionality, which is essential for real-time and embedded applications.

The methodology focuses on the application of sparse Principal Component Analysis (sPCA), as described in Equation 2.9, which is a variant of traditional PCA that incorporates an ℓ_1-norm penalty to enforce sparsity in the loading vectors. This sparsity constraint serves a dual purpose: it enhances the interpretability of the components by identifying a limited number of influential wavelengths and it facilitates the practical implementation of multispectral sensors by limiting the number of required optical filters. Through this process, the spectral bands that exhibited the highest variability and discriminative power across the dataset were effectively identified.

Once the sparse principal components were computed, wavelengths associated with the maximum absolute values of the loadings were selected. These wavelengths correspond to the most informative positions in the spectral domain and are therefore ideal candidates for multispectral filter design. These selected spectral channels were then used to simulate a virtual multispectral sensor by emulating narrowband filters centered around the chosen wavelengths.

To evaluate the performance of the simulated sensor, a feature extraction pipeline was constructed based on the reduced spectral information. These features were subsequently input into a series of well-established machine learning classifiers, including Extreme Gradient Boosting (XGBoost), Logistic Regression (LR), and Linear Discriminant Analysis (LDA). These algorithms were chosen because of their proven effectiveness in high-dimensional classification problems and their complementary strengths in handling linear and nonlinear decision boundaries.

Furthermore, the simulation of each spectral channel was modeled by introducing a synthetic filter function. Specifically, for a given filter $Z(\lambda)$ centered at a target wavelength λ_c and characterized by a specific transmittance profile T,

$$Z(\lambda) = T\frac{1}{\sigma\sqrt{2\pi}}exp\left(\frac{(\lambda-\lambda_c)^2}{2\sigma^2}\right) \tag{4.4}$$

$$\sigma = \frac{FWHM}{2\sqrt{2ln(2)}} \tag{4.5}$$

where FWHM is the full-width-at-half-maximum of each channel. To simulate the reflectance response of each channel c_k of the optical sensor, the following equation was used:

$$c_k = \sum_{i=1}^{d_x} Z_k(\lambda_i) S(\lambda_i) \tag{4.6}$$

where k is the number of channels and $S(\lambda)$ is the spectrum $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$. This formulation allows for the approximation of the sensor response, which can be achieved through real-world implementation using optical bandpass filters, effectively bridging the gap between theoretical wavelength selection and practical sensor fabrication. By simulating the filtering process on hyperspectral data, we evaluated and refined the filter configurations without the need for immediate physical prototyping, thereby offering a cost-effective and flexible design strategy. In addition to the simulated raw filter responses (Ec.4.9), we derived a set of relative descriptors to further enhance the feature representation. These descriptors include not only individual filter values but also their pairwise differences, ratios, products, and sums. This enriched feature space captures both absolute and contextual spectral information, which can enhance the discriminative power of classical machine learning models in subsequent classification tasks.

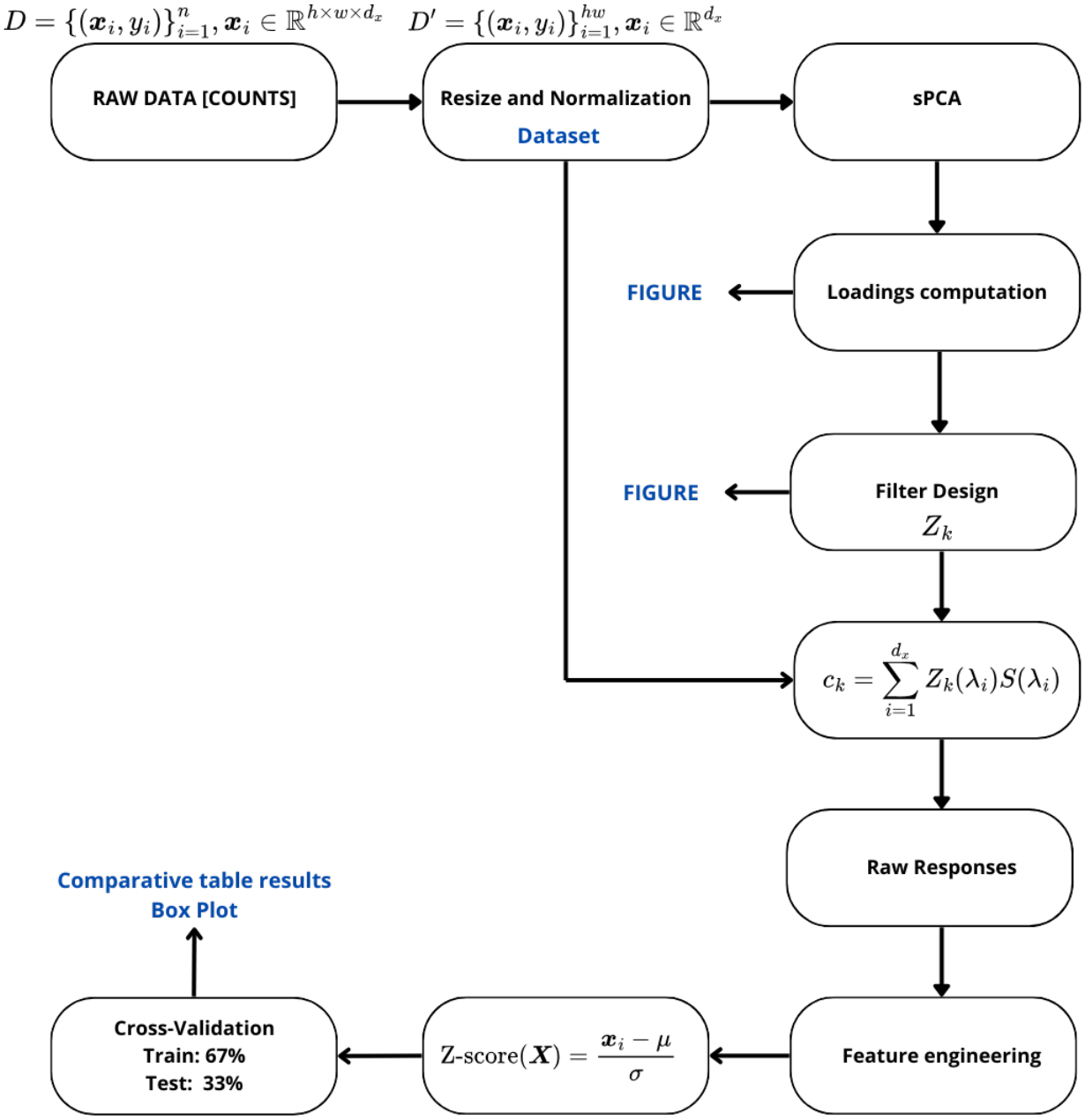

Fig. 4.1: Flowchart illustrating the methodology for specific objective 1. Raw hyperspectral data were reshaped, normalized, and processed via sparse PCA to identify informative wavelengths for filter design. Simulated filter responses were standardized using Z-score normalization and were used for feature extraction. The resulting features were evaluated using classical machine learning models. Blue-labeled elements indicate points where scientific results are generated and reported.

4.3 Prepossessing and cleaning

In satellite-based hyperspectral imaging, it is standard practice to exclude the spectral regions around 1.4 μm and 1.9 μm because of the strong atmospheric absorption primarily associated with the vibrational overtones of hydroxyl (O–H) bonds in water vapor [45,53]. Although these absorption features can provide valuable compositional information under controlled laboratory conditions, particularly for minerals containing structural water such as clays, they are highly sensitive to environmental factors in field settings. Variations in the surface moisture or hydration state can significantly alter the reflectance in these regions, introducing unwanted variability unrelated to the intrinsic mineralogy. By systematically removing these bands during preprocessing, it is possible to reduce the influence of such confounding effects and improve the stability and generalizability of classification models across diverse acquisition conditions [54,55].

Figure 4.2 illustrates the reflectance spectrum of a representative mineral sample over the wavelength range of approximately 900–2500 nm. The plot highlights undesirable absorption features centered around 1.4 μm and 1.85 μm, which are well-recognized as water absorption bands. These regions are highly sensitive to atmospheric and surface moisture content, potentially introducing variability unrelated to intrinsic mineralogical composition. To mitigate the influence of such external factors, the spectral bands corresponding to these absorption features were systematically removed from the dataset prior to model training. The excluded regions are marked in red and blue within the figure. This preprocessing step is crucial for enhancing the generalizability and interpretability of AI-based mineral classification models, as it ensures that the learning process focuses solely on mineral-specific spectral characteristics. The resulting models demonstrated enhanced robustness and reliability by eliminating confounding variability due to ambient humidity, particularly under varying environmental conditions.

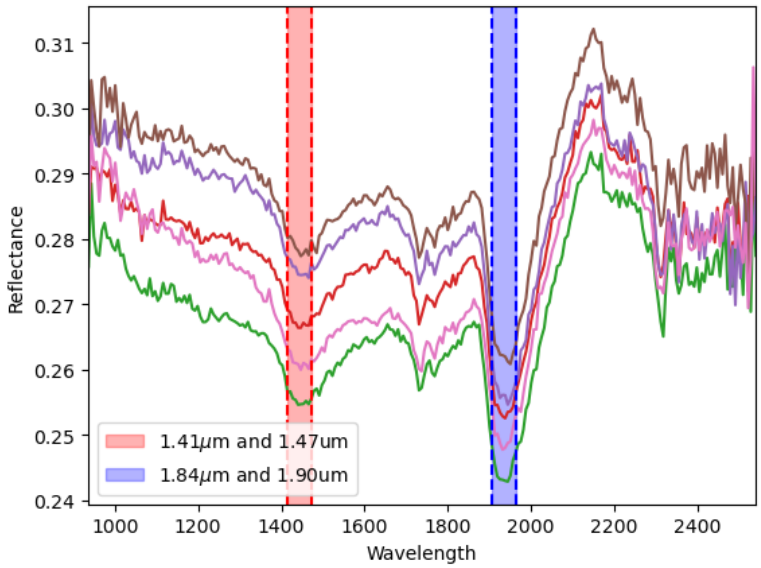

Fig. 4.2: Example of spectral profiles with shaded regions indicating strong absorption features associated with vibrational overtones of hydroxyl (O–H) bonds. The bands centered near 1400 nm and 1900 nm were excluded from the analysis due to their pronounced sensitivity to environmental moisture in mineral samples, as well as the low signal-to-noise ratio commonly observed in these regions in airborne and satellite-based hyperspectral data.

4.4 Data exploration using sPCA

As defined in Eq. 2.9, the sparse Principal Component Analysis (sPCA) framework introduces an Ł$_1$-norm penalty controlled by the hyperparameter $\alpha \in \mathbb{R}$ to promote sparsity in the principal component loadings. The inclusion of this regularization term encourages the model to retain only the most informative spectral bands, thereby improving the interpretability and enabling a more efficient filter design for multispectral sensing applications.To determine the appropriate value for α, we conducted a comparative analysis using a candidate set $\alpha \in \{0.01, 0.1, 1, 2\}$. For each value, sPCA was performed on the normalized hyperspectral dataset and the selection was guided by maximizing the cumulative explained variance captured by the first three sparse principal components (PCs). This criterion ensures that the reduced set of features retains the most salient spectral information while enforcing a level of sparsity that is compatible with the practical sensor constraints. Thus, the optimal value of α reflects the tradeoff between variance preservation and interpretability through dimensional reduction.

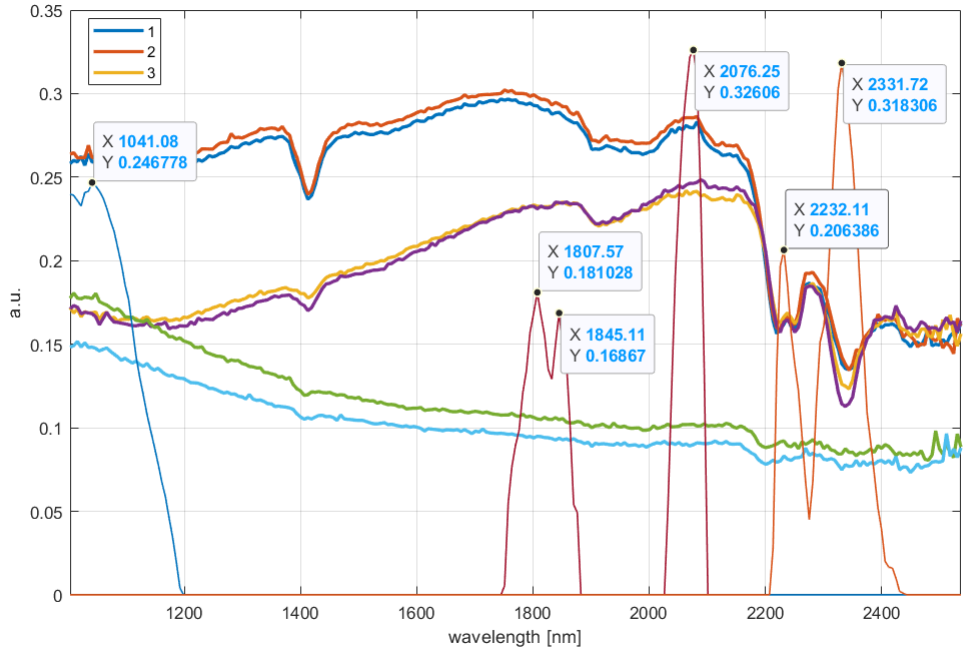

Fig. 4.3: Absolute values of the principal component loadings for Dataset 1. The curves resemble Gaussian-like profiles from which the central and peak wavelengths were extracted. These wavelengths were later used to guide the design and simulation of the optical filters for the multispectral sensor.

4.5 Real filter design and simulation

Given a filter $Z(\lambda)$ centered in λ_c with transmittance T:

$$Z(\lambda) = T\frac{1}{\sigma\sqrt{2\pi}}exp\left(\frac{(\lambda-\lambda_c)^2}{2\sigma^2}\right) \tag{4.7}$$

$$\sigma = \frac{FWHM}{2\sqrt{2ln(2)}} \tag{4.8}$$

where FWHM is the full-width-at-half-maximum of each channel. To simulate the reflectance response of each channel c_k of the optical sensor, we used the following equation:

$$c_k = \sum_{i=1}^{d_x} Z_k(\lambda_i) S(\lambda_i) \tag{4.9}$$

where k is the number of channels and $S(\lambda)$ is the spectrum $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$. Additionally, we defined relative descriptors by considering the raw values of single filters (ec.4.9), differences, ratios, products, and sums.

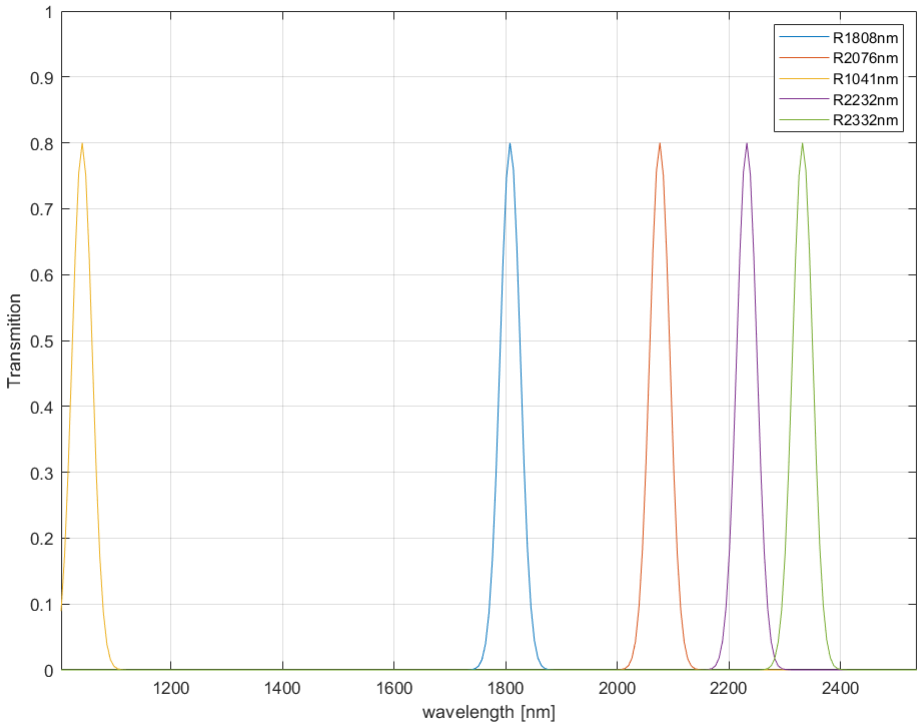

Fig. 4.4: Simulated spectral response curves of the optical filters designed for the multispectral sensor. Each filter corresponds to a selected central wavelength derived from the sparse principal component loadings, approximating Gaussian passbands tailored for optimal discrimination of the target lithologies.

Classifier	Accuracy	f1-score
XGBOOST	**94.62**%	**94.42**%
LR	80.28%	78.23%
LDA	84.85%	83.40%

Table 4.1: Comparative results using the entire spectrum (test dataset 1).

Classifier	Accuracy	f1-score
XGBOOST	**89.43**%	**88.37**%
LR	70.68%	62.49%
LDA	70.17%	65.30%

Table 4.2: Simulation results of five channel responses of the optical sensor (test dataset 1).

4.6 Machine learning models performance

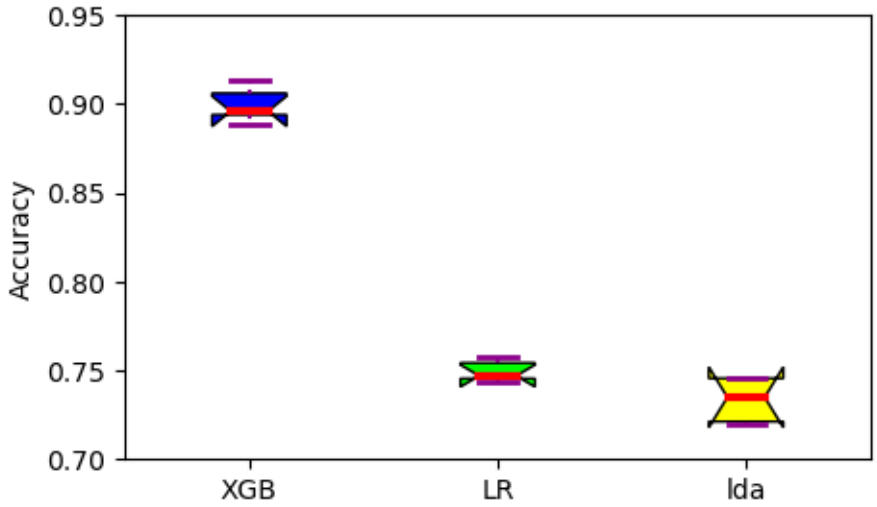

Fig. 4.5: Accuracy performance across a 5-fold cross-validation on Training Dataset 1. Box plots represent the distribution of accuracy scores, highlighting the median, interquartile range, and confidence intervals. Among the evaluated models, XGB exhibited the highest mean accuracy and lowest variance, indicating both superior predictive performance and greater stability. In contrast, LR and LDA showed lower mean accuracies and broader variability across folds.

5. Overview of Deep learning architecture proposed

5.1 Deep learning framework proposed

It is worth noting that grouping the spectral signatures can enhance the classification performance of deep learning methods in HSI. Thus, we introduce trainable spectral stride embedding (SSE). Given a hyperspectral cube $\boldsymbol{I} \in \mathbb{R}^{h \times w \times d_x}$, this can be reshaped by considering only the spectral signatures to obtain $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{hw}] \in \mathbb{R}^{hw \times d_x}$ with spectral signatures $\boldsymbol{x} \in \mathbb{R}^{d_x}$. Subsequently, by applying a grouping method $g(\cdot)$ using a sliding window with a fixed stride over the spectral dimension, we obtain a sequence representation of the spectrum. Given $g(x) = [x_1, x_2, .., x_i, ..., x_m] \in \mathbb{R}^{L \times W}$, this grouping algorithm can be described as follows:

$$\begin{aligned} x_1 &= [x_1, x_2, \ldots, x_w] \\ x_2 &= [x_{s+1}, x_{s+2}, \ldots, x_{s+w}] \\ x_i &= [x_{(i-1)s+1}, x_{(i-1)s+2}, \ldots, x_{(i-1)s+w}] \\ x_m &= [x_{d_x-w+1}, x_{d_x-w+2}, \ldots, x_{d_x}] \end{aligned} \tag{5.1}$$

where W, S, and L are the hyperparameters that define the size of the window, stride, and sequence, respectively, as defined in Eq.5.2; note that $\lfloor \cdot \rfloor$ represents round down to nearest integer.

$$\boldsymbol{L} = \left\lfloor \frac{d_x - W}{S} + 1 \right\rfloor \tag{5.2}$$

Finally, embedding is represented as a trainable matrix (Eq.5.3):

$$f = \mathcal{LN}(\boldsymbol{LP}(\boldsymbol{h}_t(\boldsymbol{g}(\boldsymbol{x})))) \tag{5.3}$$

where $\boldsymbol{LP}(h_t(g(\boldsymbol{x}))) \in \mathbb{R}^{L \times d_e}$ denotes the linear projection of the outputs of a bidirectional GRU, d_e is the embedding dimension, $h_t = \left[\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}\right] \in \mathbb{R}^{L \times 2d_h}$ represents the concatenated

hidden states from both directions, and $d_h = \frac{d_e}{2}$ is the hidden state dimension of each GRU direction. The GRU architecture is described by Eq. 2.31 and Eq. 2.27.

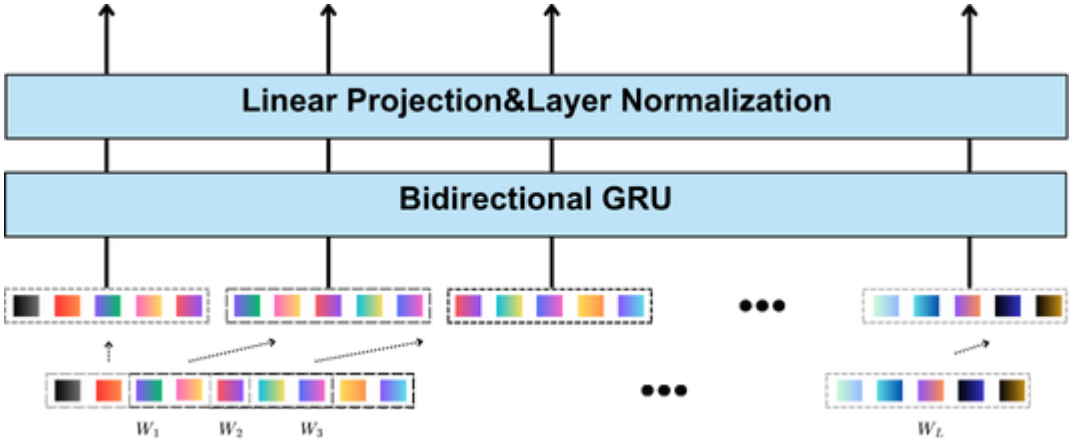

Fig. 5.1: Representation of the Spectral Sliding Embedding (SSE) module. A sequence of spectral windows was extracted from the input hyperspectral signature and processed using a Bidirectional GRU to capture contextual dependencies across adjacent spectral segments. The output is then passed through a linear projection and layer normalization to generate enhanced feature representations for subsequent classification.

Embedding is the input of a multihead self-attention mechanism to properly highlight the features of the embedding, as represented in Eq. 5.7.

$$z_i^{(h)} = \sum_{j=1}^{L} \alpha_{ij}^{(h)} (f_j W_V^{(h)}) \tag{5.4}$$

where,

$$\alpha_{ij}^{(h)} = \frac{\exp(e_{ij}^{(h)})}{\sum_{k=1}^{L} \exp(e_{ik}^{(h)})} \tag{5.5}$$

$$e_{ij}^{(h)} = \frac{(f_i W_Q^{(h)})(f_j W_K^{(h)})^T}{\sqrt{d_k}} \tag{5.6}$$

where the queries (Q), keys (K), and values (V) are trainable matrices $\boldsymbol{W_Q^{(h)}}, \boldsymbol{W_K^{(h)}}$ and $\boldsymbol{W_V^{(h)}} \in \mathbb{R}^{d_W \times d_k}$ respectively. h is the number of parallel attention heads and $d_k = d_e$ is the embedding size of the self-attention head. Finally, the output of multihead attention is given by Eq.5.7.

$$z = Concat(z^{(1)}, ..., z^{(h)})\boldsymbol{W^O} \tag{5.7}$$

where $\boldsymbol{W^O} \in \mathbb{R}^{hd_k \times d_{model}}$ is a trainable matrix and the output $z \in \mathbb{R}^{L \times d_{model}}$. This output serves as an input to convolutional neural network blocks with skip connections, allowing the model to capture dependencies at multiple levels of abstraction.

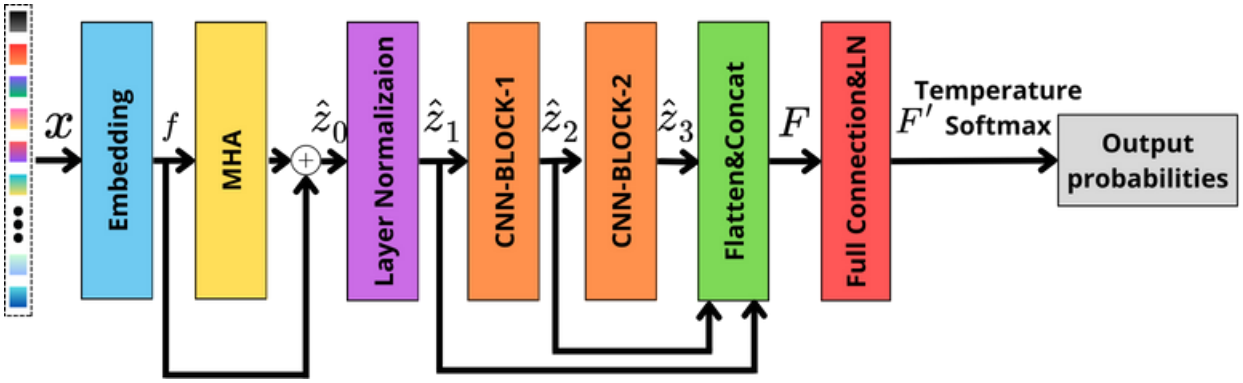

Fig. 5.2: Proposed HSI classification framework integrating an embedding, multi-head attention (MHA), and convolutional blocks with skip connections. Embedding captures sequential spectral dependencies, whereas MHA models global interactions. The CNN blocks extract hierarchical local features, and the skip connections preserve intermediate representations. The concatenated features were normalized and passed through a fully connected layer and Softmax for the final classification. This architecture effectively combines local and global spectral information, making it well-suited for hyperspectral data.

Layer	Depth	Kernel size (C_1)	Stride length	Padding	Activation
Conv1D	32	3	1	same	ReLU
Conv1D	32	3	1	same	ReLU
Conv1D	32	3	1	same	ReLU
LayerNormalization					
Dropout					

Table 5.1: CNN-BLOCK-1.

For each convolution with padding (the same), can be defined as

$$z_i^{(l)} = \sum_{j=0}^{k^{(l)}-1} \sum_{c=0}^{d^{(l-1)}-1} z_{i \cdot s^{(l)}+j-p^{(l)},c}^{(l-1)} \cdot W_{j,c}^{(l)} \tag{5.8}$$

Layer	Depth	Kernel size (C_2)	Stride length	Padding	Activation
Conv1D	128	3	1	same	ReLU
Conv1D	128	3	1	same	ReLU
Conv1D	128	3	1	same	ReLU
LayerNormalization					
Dropout					

Table 5.2: CNN-BLOCK-2.

$$p^{(l)} = \left\lfloor \frac{k^{(l)} - 1}{2} \right\rfloor \tag{5.9}$$

$$L_{\text{out}}^{(l)} = \left\lceil \frac{L^{(l-1)}}{s^{(l)}} \right\rceil \tag{5.10}$$

where p^l denotes the padding, $L^{(l)}$ denotes the sequence, and $s^{(l)}$ denotes the stride.

Each block CNN-Block$_k(\cdot)$ consists of the sequential application of three one-dimensional convolutional layers, each followed by a rectified linear unit (ReLU) activation function, layer normalization operation, and dropout regularization layer (Tables 5.1 and 5.2).

$$\hat{\boldsymbol{Z}}_0 = \text{MHA}(\boldsymbol{f}) + \boldsymbol{f} \tag{5.11}$$

$$\hat{\boldsymbol{Z}}_1 = \mathcal{LN}(\hat{\boldsymbol{Z}}_0) \tag{5.12}$$

$$\hat{\boldsymbol{Z}}_2 = \text{CNN-Block}_1(\hat{\boldsymbol{Z}}_1) \tag{5.13}$$

$$\hat{\boldsymbol{Z}}_3 = \text{CNN-Block}_2(\hat{\boldsymbol{Z}}_2) \tag{5.14}$$

Then, we define each connection to the flattened layer as $\hat{z}_1$, $\hat{z}_2$ and $\hat{z}_3$. Then, $\mathbf{F} \in \mathbb{R}^{d_e(C_1+C_2)}$ represents the *feature map* of the neural network. Considering the dimensions of each block $\mathbf{d}_e$, $\mathbf{C}_1$ and $\mathbf{C}_2$, by applying $\mathbf{LP}$, this can be represented as follows in Eq. 5.15.

$$\mathbf{F} = \text{Concat}(\text{Flatten}(\hat{z}_1),\ \text{Flatten}(\hat{z}_2),\ \text{Flatten}(\hat{z}_3)) \tag{5.15}$$

where $\hat{z}_1$, $\hat{z}_2$, and $\hat{z}_3$ denote the output feature representations of the MHA, CNN-Block$_1$, and CNN-Block$_2$ modules, respectively. Each output is flattened and concatenated along the feature dimension. The total dimensionality of $\mathbf{F}$ depends on the embedding size d_e and number of channels C_1 and C_2 from the CNN blocks. The resulting vector $\mathbf{F}$ is then processed using a linear projection layer with layer normalization, as defined in Eq. 5.16.

$$\mathbf{F}' = \mathcal{LN}(\mathbf{WF} + \mathbf{b}) \tag{5.16}$$

where $\mathbf{F}' \in \mathbb{R}^{d_p}$ denotes the final feature representation before the Softmax activation function, $\mathbf{W} \in \mathbb{R}^{d_p \times d_e(C_1+C_2)}$ denotes a trainable weight matrix, and $\mathbf{b} \in \mathbb{R}^{d_p}$ denotes the bias vector of the linear projection.

Finally, to obtain the output probabilities of the model, we define a softmax function with a trainable temperature over $\mathbf{F}'$ and introduce a temperature parameter τ (ec.5.17).

$$\text{softmax}\left(\frac{\mathbf{F}'_i}{\tau}\right) = \frac{\exp\left(\frac{\mathbf{F}'_i}{\tau}\right)}{\sum_k \exp\left(\frac{\mathbf{F}'_k}{\tau}\right)} \tag{5.17}$$

where τ denotes the trainable temperature parameter, $\mathbf{k}$ classes and $\mathbf{i}$ activation.

5.2 Results

The results of the classification analysis indicate that the proposed model demonstrates superior performance compared to conventional machine learning techniques. With an accuracy of 98.23% and an F1-score of 98.14%, our model significantly outperformed XGBoost, which achieved 89.43% accuracy and an 88.37% F1-score. Furthermore, our approach exhibits markedly enhanced performance relative to logistic regression (LR) and linear discriminant analysis (LDA), both of which demonstrate considerably lower efficacy. These findings suggest that our method effectively identifies complex patterns within the data, resulting in a more reliable and robust classification system.

Classifier	Accuracy	f1-score
XGBOOST	89.43%	88.37%
LR	70.68%	62.49%
LDA	70.17%	65.30%
SpectralFormer	93.56%	94.45%
ours	**94.73**%	**95.14**%

Table 5.3: Comparison of the simulated five channel responses of the optical sensor (test dataset 1).

5.3 Discussion

The proposed artificial intelligence model demonstrates a significant advancement over traditional machine learning techniques, as evidenced by its superior performance metrics in terms of both accuracy and F1-score. The model achieved an accuracy of 98.23% and an F1-score of 98.14%, which markedly surpassed the performances of XGBoost (89.43% accuracy, 88.37% F1-score), logistic regression (LR) (70.68% accuracy, 62.49% F1-score), and linear discriminant analysis (LDA) (70.17% accuracy, 65.30% F1-score). These results underscore the capacity of the model to effectively capture and leverage complex patterns within data, resulting in a more robust and reliable classification system.

A critical aspect of this investigation was the performance of the model on both simulated and real data (full spectrum). The simulated data, representing discrete responses from an optical sensor (equivalent to five discrete data points from a photodiode), provided a controlled environment to validate the capabilities of the model. However, the definitive assessment of the efficacy of the model lies in its application to raw spectral measurements, which encompasses the complete spectrum rather than a limited subset. The superior performance of the model on simulated data suggests strong potential for generalization to real-world scenarios, where the complexity and variability of the data are significantly higher.

The substantial disparity in performance underscores the advantages of our approach, which likely employs more sophisticated feature representations and exhibits enhanced generalization capabilities. This represents a crucial advantage over classical models that often struggle with such complexities. The sophisticated feature representations and enhanced generalization capabilities of the proposed model are likely to be key factors contributing to its performance.

6. Comparative benchmarking of state-of-the-art approaches

6.1 Comparison of hyperspectral models

6.1.1 Models description

Comparison with Leading Backbone Networks: Several notable baselines have been employed for performance comparisons, including Logistic Regression (LR), Linear Discriminant Analysis (LDA), support vector machine (SVM) [56], 1-D CNN [57], 2-D CNN, RNN [58], transformers [29], SpectralFormer [31], and the advanced 1-D RMC proposed in this study. The specific configurations of these methods are described in detail below:

1. For the LR, was mapped by using the penalties = $[l_2, l_1, elasticnet]$ with a strength $C = [0.01, 0.1, 1]$.

2. For LDA, Shrinkage of $[None, 0.1, 0.01, 1]$ was mapped and with solvers Least Squares solution, Eigenvalue decomposition-based, and Singular Value Decomposition (does not use the shrinkage parameter).

3. SVM was performed with linear basis using $C = [0.01, 0.1, 1, 10]$ which controls the trade-off between the classifier margin and the classification error, and $\gamma = [0.01, 0.1, 1, 10]$ respectively.

4. For CNN-1D, the SPEC-CNN model was used with the same parameters of the article [57], this is composed by four block of CNN and multi-feature approach.

5. 2-D CNN (two CNN blocks) and RNN (Two GRU) were implemented using the architectures in `https://github.com/AnkurDeria/HSI-Traditional-to-Deep-Models` [58].

6. For the transformers models, ViT [59] architecture was performed by using embedding proposed in [31] and furthermore the 1D and 2D version of SpectralFormer with the exactly hyperparameter published in the article.

6.1.2 Validation stage of the models

Datasets 1 and 2 were subsampled and divided into training and testing samples using a random-partitioning process. Specifically, 70% of the samples were designated for training and 30% were designated for testing using a five-fold stratified k-fold, employing cross-validation to optimize the outcomes and attain the highest level of accuracy achievable with the mapped hyperparameters. This can be represented as follows:

Algorithm 1 Stratified K-Fold Cross Validation

Require: Dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ and number of folds K.

Ensure: Shuffle training and validation indices for each fold

1: Initialize $\mathcal{D}'$ as an empty list to store the folds
2: Arrange dataset $\mathcal{D}$ according to class labels
3: Partition $\mathcal{D}$ into K equal-sized folds trying to each fold has the same posterior class distribution
4: **for** $k = 1$ to K **do**
5: $\quad train_{indices} \leftarrow \bigcup_{j \neq k}$ indices of fold j
6: $\quad test_{indices} \leftarrow$ indices of fold k
7: $\quad$ Append $(train_{indices}, test_{indices})$ to $\mathcal{D}'$
8: **end for**

6.1.3 Data Augmentation Techniques

To improve model generalization and robustness to input variability, we applied a set of stochastic data augmentation techniques directly to the input signals. These augmentations are defined as follows:

- **Random Noise:** Gaussian noise with zero mean and standard deviation σ is added element-wise to the input signal:

$$\tilde{x} = x + \mathcal{N}(0, \sigma^2)$$

  This simulates sensor noise and promotes invariance of the model to minor perturbations in the input space.

- **Random Offset:** A constant bias term is uniformly sampled from the interval $[-\delta, \delta]$ and added to the entire signal:

$$\tilde{x} = x + \mathcal{U}(-\delta, \delta)$$

This operation imitates the baseline shifts or sensor drifts that are commonly observed in real-world measurements.

- **Random Scaling:** Each element of the signal is multiplied by a scaling factor sampled uniformly from $[1 - \gamma, 1 + \gamma]$, where:

$$\tilde{x}_i = x_i \cdot s_i, \quad s_i \sim \mathcal{U}(1 - \gamma, 1 + \gamma)$$

This reflects multiplicative variations due to changes in the sensor sensitivity or environmental conditions.

6.1.4 Training details

The experiments were conducted using an NVidia A100 40GB and 128 GB RAM. All parameters were iteratively selected: spectral stride window embedding (SSE) with a fixed length of $W = 11$ and $s = 3$ and 128 hidden layers for each GRU of the bidirectional network (output of 256). The number of heads of the multihead self-attention block was set to $h = 4$, $d_k = d_e = d_{model}/h = 64$, and $d_p = 512$. To prevent overfitting, dropout with a probability of 0.25 was implemented in each stage block and 0.5, after F'. Moreover, each linear projection is regularized using an orthogonal regularizer mapped with $\alpha \in [0.01]$. Additionally, data augmentation techniques were performed using random noise ($\sigma = [0.05, 0.15]$), offset ($\delta = [0.15, 0.5]$), and scaling operations ($\gamma = [0.05, 0.15]$) in each mini-batch. The AdamW optimization algorithm [60] was employed with a weight decay parameter of 4×10^{-3} and an initial learning rate of 10^{-3}. The learning rate was reduced by a factor of 0.1 every 10 epochs over the course of 400 epochs. The trainable temperature parameter was empirically initialized as 0.35.

6.1.5 Classification performance metrics

To assess the performance of the models, the F1-score and Accuracy were employed, defined as

$$F1 = \frac{1}{K} \sum_{k=0}^{K-1} 2 \times \frac{\text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{6.1}$$

where K is the number of classes, and

$$\text{Recall}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k} \quad \text{Precision}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}. \tag{6.2}$$

$$\text{Accuracy} = \frac{\sum_{k=0}^{K-1} \text{TP}_k}{\sum_{k=0}^{K-1} (\text{TP}_k + \text{FP}_k + \text{FN}_k + \text{TN}_k)} \tag{6.3}$$

"True positive" (TP) and "true negative" (TN) pertain to instances where the model accurately forecasts a positive or negative class, respectively. Conversely, "false positive" (FP) and "false negative" (FN) refer to circumstances in which the model mispredicts the positive or negative class. The precision score measures the proportion of correctly identified predicted classes, whereas the recall reflects the number of objects in the testing set that the model can accurately detect.

Only for the standard classification challenge, the metrics of Overall Accuracy, Average Accuracy, and Kappa Coefficient were employed (**N** instances). Overall Accuracy is the proportion of correctly classified instances to the total number of instances. The average accuracy represents the average accuracy for each class. The Kappa Coefficient evaluates the degree of agreement between the observed and expected accuracies based on random chance.

$$OA = \frac{\sum_{k=1}^{K} \text{TP}_k}{N} \tag{6.4}$$

$$AA = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k} \tag{6.5}$$

$$\kappa = \frac{\text{OA} - \text{EA}}{1 - \text{EA}} \tag{6.6}$$

where:

$$EA = \frac{1}{N^2} \sum_{k=1}^{K} (\text{TP}_k + \text{FN}_k)(\text{TP}_k + \text{FP}_k) \tag{6.7}$$

6.2 Results over mineral dataset

The efficacy of our proposed 1D-RMC model for mineral classification is demonstrated by the performance comparison presented in Table 5.1. Our method outperformed both traditional

Class No.	Conventional Classifiers			Classic Backbone Networks		Transformers		Ours
	LR	LDA	SVM	1-D CNN	RNN	ViT	SpectralFormer1D	1D-RMC
OA (%)	70.68	70.17	86.87	97.50	98.53	98.14	89.91	**98.94**
AA (%)	62.49	65.30	85.86	96.11	98.12	97.12	88.33	**98.43**
κ (%)	67.95	69.27	86.11	96.11	98.13	98.10	88.33	**98.92**

Table 6.1: Performance comparison of various classifiers on pure mineral dataset.

classifiers (LR, LDA, SVM) and advanced deep learning architectures, including 1-D CNNs, RNNs, and transformer-based models (ViT, SpectralFormer1D), achieving the highest scores across all evaluation metrics. The model attained an overall accuracy (OA) of 98.94%, average accuracy (AA) of 98.43%, and kappa coefficient (κ) of 98.92%.

These exceptional results underscore the superiority of the 1D-RMC, which can be attributed to its enhanced capability to extract discriminative features from hyperspectral data. The substantial improvement over conventional techniques and transformer-based architectures suggests that our model effectively captures both local and global dependencies in spectral information. This capability positions the 1D-RMC as a promising solution for mineral classification tasks, particularly in scenarios that require high precision and robustness.

6.3 Results over standard datasets

As mentioned in Section 3.7.4, the performance metrics of the models are presented in Tables 6.2 and 6.3.

Table 6.2 presents a detailed performance comparison of various classifiers on the Indian Pines dataset, using standard training and test samples. The classifiers were categorized into four groups: Conventional Classifiers, Classic Backbone Networks, Transformers, and the proposed method (our method: 1D-RMC). The performance metrics include Overall Accuracy (OA), Average Accuracy (AA), and individual class accuracies. Conventional Classifiers: Logistic Regression (LR), Linear Discriminant Analysis (LDA), and Support Vector Machines (SVM) exhibit varying degrees of effectiveness. Notably, SVM generally outperformed LR and LDA, achieving the highest OA of 73.44%, AA of 83.06%, and κ of 69.94%. However, these classifiers exhibit limited performance in several classes, particularly in Classes 2 and 12, where the accuracy is relatively low across all conventional methods. Classic Backbone Networks, including 1-D CNN, 2-D CNN, and RNN, have demonstrated superior performance compared to conventional

Class No.	Conventional Classifiers			Classic Backbone Networks			Transformers			Ours
	LR	LDA	SVM	1-D CNN	2-D CNN	RNN	ViT	SpectralFormer1D	SpectralFormer2D	1D-RMC
1	67.20	65.61	68.71	66.40	65.90	61.78	50.43	63.15	63.80	**74.06**
2	55.61	55.99	61.99	57.27	38.52	65.43	57.02	64.92	**83.16**	73.21
3	82.61	82.07	89.67	77.17	91.30	85.33	**92.39**	86.41	82.61	90.76
4	86.13	76.73	**93.51**	87.70	91.50	88.59	87.70	83.45	87.70	92.62
5	89.53	93.11	89.38	91.82	**94.55**	89.10	88.67	87.37	83.50	90.53
6	**97.95**	98.18	94.31	94.99	96.13	93.62	87.24	93.62	95.44	**97.95**
7	70.81	67.43	70.37	74.95	75.49	76.36	81.81	80.94	82.35	**84.42**
8	52.07	44.75	57.65	67.62	58.35	69.98	70.80	66.34	69.69	**75.31**
9	71.10	70.74	76.60	81.91	**84.40**	71.28	58.69	76.06	68.62	79.08
10	98.77	**100.00**	98.77	99.38	99.38	97.53	99.38	98.15	99.38	99.38
11	84.08	84.16	87.14	86.09	79.34	94.29	87.54	**93.89**	91.16	**94.69**
12	77.88	**78.48**	72.42	77.58	**83.64**	63.03	55.76	59.70	80.91	72.73
13	95.56	86.67	95.56	93.33	88.89	95.56	97.78	**100.00**	95.56	**100.00**
14	66.67	76.92	82.05	79.49	87.18	79.49	89.74	84.62	48.72	**89.74**
15	**100.00**	**100.00**	90.91	**100.00**	**100.00**	90.91	90.91	90.91	**100.00**	90.91
16	60.00	60.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
OA (%)	70.45	67.96	73.44	75.70	72.03	76.27	72.69	75.77	78.05	**82.28**
AA (%)	78.50	77.55	83.06	83.48	83.41	82.64	80.99	83.10	83.29	**87.84**
κ	66.69	63.97	69.94	72.28	68.39	72.95	68.86	72.53	74.92	**79.76**

Table 6.2: Performance comparison of various classifiers on Indian Pines dataset using the standard train and test samples.

classifiers. The RNN model demonstrated notable performance, achieving an Overall Accuracy (OA) of 76.27%, Average Accuracy (AA) of 82.64%, and κ coefficient of 72.95%. These results underscore its efficacy in processing hyperspectral data sequences.. These models leverage spatial and spectral information more effectively, thereby contributing to an enhanced performance. Transformer-based models ViT, SpectralFormer1D, and SpectralFormer2D exhibit significant improvements over conventional classifiers and classic backbone networks. SpectralFormer2D achieved the highest OA among the transformers at 74.92%, whereas SpectralFormer1D achieved an AA of 75.81%. These models capitalize on the self-attention mechanism, which is adept at capturing long-range dependencies in the hyperspectral data. The proposed method, 1D-RMC, outperformed all other models across nearly all metrics. It achieved an exceptional OA of 82.28%, AA of 87.84%, and κ of 79.76%. This method consistently provides high accuracy across different classes, particularly in classes in which other models underperform. For instance, it achieved perfect accuracy (100.00%) in classes 13 and 16, demonstrating its robustness and reliability for hyperspectral image classification. The superiority of 1D-RMC can be attributed to its innovative architecture, which effectively integrates both spectral and spatial features. This model likely incorporates advanced feature extraction and classification strategies that address the complex nature of hyperspectral data more comprehensively than the existing methods.

Class No.	Conventional Classifiers			Classic Backbone Networks			Transformers			Ours
	LR	LDA	SVM	1-D CNN	2-D CNN	RNN	ViT	SpectralFormer1D	SpectralFormer2D	1D-RMC
1	83.29	82.24	84.33	**88.41**	84.90	86.13	83.76	84.90	84.05	**88.41**
2	95.86	98.50	96.99	95.49	96.52	93.80	97.56	96.33	**99.72**	96.62
3	99.60	**100.00**	99.80	**100.00**	99.60	99.80	99.80	99.80	98.61	**100.00**
4	96.88	94.70	98.96	97.73	96.59	**99.34**	98.77	98.39	97.82	96.02
5	97.16	96.50	98.58	98.30	96.21	97.73	97.82	97.25	**100.00**	98.67
6	97.90	86.71	94.41	95.10	95.10	93.01	**99.30**	**99.30**	97.20	95.10
7	85.26	82.18	79.85	87.22	86.01	84.42	78.08	88.34	87.69	**88.53**
8	55.46	31.91	57.83	75.02	58.21	57.74	56.69	51.47	**86.51**	73.22
9	57.88	59.96	70.16	**81.30**	80.83	76.77	68.08	72.33	75.73	78.38
10	86.58	64.96	89.85	74.61	**99.38**	66.12	66.70	83.78	51.16	88.99
11	71.63	65.28	89.85	81.59	**92.79**	77.89	68.79	87.19	84.16	87.10
12	55.72	48.32	67.05	**83.67**	69.07	63.50	50.14	61.58	74.26	79.73
13	56.14	49.82	70.53	**75.09**	68.77	71.58	62.46	67.02	74.26	76.14
14	**100.00**	97.57	**100.00**	99.60	98.79	99.60	99.19	98.79	**100.00**	**100.00**
15	97.67	97.89	97.46	98.73	98.09	97.25	98.31	98.31	**100.00**	98.73
OA (%)	80.36	74.81	84.69	87.52	85.86	82.20	78.90	83.74	85.75	**88.60**
AA (%)	82.47	77.10	86.33	88.79	87.12	84.31	81.70	85.65	87.70	**89.71**
κ	78.69	72.63	83.40	86.46	84.65	80.72	77.16	82.38	84.53	**87.64**

Table 6.3: Performance comparison of various classifiers on Houston2013 dataset using the standard train and test samples.

Table 6.3 presents a performance comparison of the same classifiers on the Houston2013 dataset using standard training and test samples. Among the Conventional classifiers, SVM consistently achieved higher accuracy than LR and LDA, with an OA of 84.60%, AA of 88.74%, and κ of 86.33%. Classic Backbone Networks exhibit the same behavior as the Indian Pine datasets. In particular, the RNN demonstrated robust performance with an OA of 87.56% and AA of 88.73%, underscoring its capability to capture temporal dependencies within the hyperspectral data. These networks effectively leverage both the spatial and spectral information, thereby contributing to their superior performance. Transformer-based models exhibit significant improvements over conventional classifiers and classic backbone networks. SpectralFormer2D stands out, with an OA of 87.65% and AA of 85.73%. Finally, the 1D-RMC method outperformed all the other models across all metrics. It achieved an impressive OA of 88.60% and AA of 89.71%. This method consistently provides high accuracy across various classes, notably excelling in classes where other models underperform. The performance of the 1D-RMC can be attributed to its novel architecture, which integrates spectral features by taking advantage of each backbone network, providing a comprehensive approach to hyperspectral data classification.

6.4 Discussion

The results presented in Tables 6.1, 6.2, and 6.3 (corresponding to the Minerals, Indian Pines, and Houston2013 datasets, respectively) demonstrate that the proposed 1D-RMC method consistently outperformed the other models in terms of OA, AA, and κ. This observation is noteworthy, and indicates the effectiveness of the proposed method. Specifically, for the Indian Pines dataset, the 1D-RMC method achieved an OA of 76.27%, an AA of 82.64%, and κ of 72.95%. In addition, the Houston2013 dataset achieved an OA of 88.60%, an AA of 89.71%, and κ of 87.64%. These results highlight the generalizability and robustness of the proposed method across various hyperspectral datasets.

Conventional classifiers fail to capture the complex spectral and/or spatial relationships inherent in hyperspectral data, as evidenced by their lower performance metrics. Classic backbone networks, particularly RNNs, exhibit marked improvements because of their ability to model temporal dependencies. Transformer-based models that leverage self-attention mechanisms further enhance the performance by capturing long-range dependencies within the data.

The proposed 1D-RMC method stands out because of its innovative approach of integrating different backbone networks and using multiple levels of abstraction based only on spectral information, which leads to superior performance across diverse classes and datasets. This consistent outperformance underscores the potential of 1D-RMC in advancing hyperspectral image classification for more accurate and reliable applications, not only in mineral applications but also in remote sensing, environmental monitoring, and agricultural assessment.

7. Conclusion

Hyperspectral images are typically characterized by a data cube that includes spatial and large spectral dimensions. In these images, the spectral signature displays remained consistent despite the changes in the spatial context. Thus, the use of spectral-spatial information can lead to an inaccurate estimation of the posterior distribution of classes owing to variations in the spatial context in practical applications.

The spectral dimension can be viewed as a sequence of data along the spectral dimension that is composed of redundant features that can lead to poor inference performance. Transformers have demonstrated a significant capability of capturing global sequential properties for this type of data. Nevertheless, transformer-based vision networks, such as the Vision Transformer (ViT) and SpectralFormer, tend to struggle to perform effectively when handling HS-like data and undergo time-consuming training. This is likely due to the inability to effectively extract local spectral variations and the poor levels of abstraction across layers.

In response to these limitations, we introduce a novel arquitecture named 1-D RMC, which is specifically designed to extract spectral information levering distinc aspect of each classic neural networks backbone. 1-D RMC relies on bidirectional recurrent units for the embedding stage, multihead self-attention to effectively penalize the sequence spectral data, and convolutional for multiple levels of abstraction, achieving state-of-the-art supervised classification performance for HSI.

A noteworthy aspect of this study is the capacity of the model to generalize across diverse hyperspectral imaging domains. In addition to its application to lithological data, where spectral differences among mineral classes tend to be subtle and often overlap, the model was also assessed using two widely adopted benchmark datasets: Indian Pines and Houston2013. These datasets are markedly different in nature, exhibiting higher spectral contrast and greater spatial variability owing to their agricultural and urban content, respectively. The ability of the proposed model to maintain strong performance across heterogeneous settings highlights its robustness and adaptability. This versatility suggests that the architecture is not only well suited for geological tasks but also holds promise for broader applications in hyperspectral image classification.

Nevertheless, it is important to recognize that the ability of any supervised model to generalize novel geological materials is inherently limited by the diversity of the training data. The limited number of mineral species included in most available datasets may constrain the applicability of the models to more complex field conditions. To address this, future work should consider the following.

- Domain adaptation techniques to transfer knowledge from labeled datasets (e.g., standard mineral libraries) to new geological domains with limited ground truth data.
- Incorporation of foundational models trained on large-scale, diverse spectral corpora, enabling zero-shot or few-shot classification of unseen minerals.
- Sensor prototyping and field validation, to evaluate the proposed multispectral configurations under realistic conditions and with direct expert supervision.

These directions aim to strengthen the collaboration between data-driven models and geological expertise, positioning AI as a tool to support and enhance field-based decision making in mineral exploration.

Bibliography

[1] Y. Liu, X. Wang, Z. Zhang, and F. Deng, "A review of deep leaning in image classification for mineral exploration," *Minerals Engineering*, vol. 204, p. 108433, 2023.

[2] F. Yang, R. Zuo, and O. P. Kreuzer, "Artificial intelligence for mineral exploration: A review and perspectives on future directions from data science," *Earth-Science Reviews*, vol. 258, p. 104941, 2024.

[3] S. Hajaj, A. El Harti, A. B. Pour, A. Jellouli, Z. Adiri, and M. Hashim, "A review on hyperspectral imagery application for lithological mapping and mineral prospecting: Machine learning techniques and future prospects," *Remote Sensing Applications: Society and Environment*, vol. 35, p. 101218, 2024.

[4] J. Cruz-Tirado, L. Honório, J. M. Amigo, L. D. Z. Cruz, D. Barbin, and R. Siche, "Portable near infrared spectrometer to predict physicochemical properties in cape gooseberry (physalis peruviana l.): An approach using hierarchical classification/regression modelling," *Journal of Food Engineering*, vol. 389, p. 112407, 2025.

[5] M. San Nicolas, A. Villate, I. Alvarez-Mora, M. Olivares, O. Aizpurua-Olaizola, A. Usobiaga, and J. M. Amigo, "Nir-hyperspectral imaging and machine learning for non-invasive chemotype classification in cannabis sativa l," *Computers and Electronics in Agriculture*, vol. 217, p. 108551, 2024.

[6] D. Peukert, C. Xu, and P. Dowd, "A review of sensor-based sorting in mineral processing: The potential benefits of sensor fusion," *Minerals*, vol. 12, no. 11, p. 1364, 2022.

[7] Y. Hatamifar, Z. Shojaeifard, and B. Hemmateenejad, "Discrimination of bottled mineral water from tap water using a dip-type colorimetric paper-based sensor array and chemometrics," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 321, p. 124719, 2024.

[8] R. Goyetche, L. Kortazar, and J. M. Amigo, "Issues with the detection and classification of microplastics in marine sediments with chemical imaging and machine learning," *TrAC Trends in Analytical Chemistry*, vol. 166, p. 117221, 2023.

[9] Y. Sun, B. Liu, X. Yu, A. Yu, P. Zhang, and Z. Xue, "Exploiting discriminative advantage of spectrum for hyperspectral image classification: Spectralformer enhanced by spectrum motion feature," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2022.

[10] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[11] L. Fasnacht, M.-L. Vogt, P. Renard, and P. Brunner, "A 2d hyperspectral library of mineral reflectance, from 900 to 2500 nm," *Scientific data*, vol. 6, no. 1, p. 268, 2019.

[12] R. Dian, S. Li, B. Sun, and A. Guo, "Recent advances and new guidelines on hyperspectral and multispectral image fusion," *Information Fusion*, vol. 69, pp. 40–51, 2021.

[13] L. Huang, R. Luo, X. Liu, and X. Hao, "Spectral imaging with deep learning," *Light: Science & Applications*, vol. 11, no. 1, p. 61, 2022.

[14] M. A. Moharram and D. M. Sundaram, "Land use and land cover classification with hyperspectral data: A comprehensive review of methods, challenges and future directions," *Neurocomputing*, 2023.

[15] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "Hybridsn: Exploring 3-d–2-d cnn feature hierarchy for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2019.

[16] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[17] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5384–5394, 2019.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[19] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[20] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE transactions on geoscience and remote sensing*, vol. 57, no. 10, pp. 8065–8080, 2019.

[21] B. Xi, J. Li, Y. Li, R. Song, Y. Shi, S. Liu, and Q. Du, "Deep prototypical networks with hybrid residual attention for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3683–3700, 2020.

[22] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral–spatial attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 449–462, 2020.

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

[24] X. Zhang, S. Shang, X. Tang, J. Feng, and L. Jiao, "Spectral partitioning residual network with spatial attention mechanism for hyperspectral image classification," *IEEE transactions on geoscience and remote sensing*, vol. 60, pp. 1–14, 2021.

[25] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral–spatial attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3232–3245, 2019.

[26] B. Manifold, S. Men, R. Hu, and D. Fu, "A versatile deep learning architecture for classification and label-free prediction of hyperspectral images," *Nature machine intelligence*, vol. 3, no. 4, pp. 306–315, 2021.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[28] J. Wang, J. Sun, E. Zhang, T. Zhang, K. Yu, and J. Peng, "Hyperspectral image classification via deep network with attention mechanism and multigroup strategy," *Expert Systems with Applications*, vol. 224, p. 119904, 2023.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[31] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "Spectralformer: Rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[32] W. Wang, L. Liu, T. Zhang, J. Shen, J. Wang, and J. Li, "Hyper-es2t: efficient spatial–spectral transformer for the classification of hyperspectral remote sensing images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 113, p. 103005, 2022.

[33] X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao, and Y. Xue, "Swin transformer embedding unet for remote sensing image semantic segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[34] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[35] L. Dang, L. Weng, Y. Hou, X. Zuo, and Y. Liu, "Double-branch feature fusion transformer for hyperspectral image classification," *Scientific Reports*, vol. 13, no. 1, p. 272, 2023.

[36] J. Yao, B. Zhang, C. Li, D. Hong, and J. Chanussot, "Extended vision transformer (exvit) for land use and land cover classification: A multimodal deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[37] Y. Xu, Y. Xie, B. Li, C. Xie, Y. Zhang, A. Wang, and L. Zhu, "Spatial-spectral 1dswin transformer with group-wise feature tokenization for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[38] S. K. Roy, A. Deria, C. Shah, J. M. Haut, Q. Du, and A. Plaza, "Spectral–spatial morphological attention transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[39] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[40] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[41] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[42] A. Kazemnejad, I. Padhi, K. Natesan Ramamurthy, P. Das, and S. Reddy, "The impact of positional encoding on length generalization in transformers," *Advances in Neural Information Processing Systems*, vol. 36, pp. 24 892–24 928, 2023.

[43] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 033–10 041.

[44] W. Park, W. Chang, D. Lee, J. Kim, and S.-w. Hwang, "Grpe: Relative positional encoding for graph transformer," in *ICLR2022 Machine Learning for Drug Discovery*, 2022.

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

[46] N. Wambugu, Y. Chen, Z. Xiao, K. Tan, M. Wei, X. Liu, and J. Li, "Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: A review," *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, p. 102603, 2021.

[47] A. Ribes and F. Schmitt, "Linear inverse problems in imaging," *IEEE Signal Processing Magazine*, vol. 25, no. 4, 2008.

[48] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[50] E. R. Dougherty, *Digital image processing methods*. CRC Press, 2020.

[51] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into deep learning*. Cambridge University Press, 2023.

[52] R. Calvini, J. M. Amigo, and A. Ulrici, "Transferring results from nir-hyperspectral to nir-multispectral imaging systems: A filter-based simulation applied to the classification of arabica and robusta green coffee," *Analytica chimica acta*, vol. 967, pp. 33–41, 2017.

[53] J. Bai, Z. Wen, Z. Xiao, F. Ye, Y. Zhu, M. Alazab, and L. Jiao, "Hyperspectral image classification based on multibranch attention transformer networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[54] J. Jiang, C. Nie, J. Deng, K. Li, L. Jia, T. Sun, and Z. Li, "Classifying iron ore with water or dust adhesion combining differential feature and random forest using hyperspectral imaging," *Minerals Engineering*, vol. 217, p. 108965, 2024.

[55] W. Langa, C. Ndou, L. Zieger, P. Harris, and N. Wagner, "Hyperspectral imaging of coal core: A focus on the visible-near-shortwave infrared (vn-swir) region," *International Journal of Coal Geology*, vol. 284, p. 104456, 2024.

[56] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[57] J. I. Cifuentes, L. E. Arias, E. Pirard, and F. Castillo, "Mineral classification using convolutional neural networks and swir hyperspectral imaging," in *AI and Optical Data Sciences V*, vol. 12903. SPIE, 2024, pp. 51–56.

[58] M. Ahmad, S. Shabbir, S. K. Roy, D. Hong, X. Wu, J. Yao, A. M. Khan, M. Mazzara, S. Distefano, and J. Chanussot, "Hyperspectral image classification–traditional to deep models: A survey for future prospects," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 968–999, 2022.

[59] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[60] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.