Deep Hyperspectral Unmixing using Transformer Network

Preetam Ghosh, Swalpa Kumar Roy, *Student Member, IEEE,* Bikram Koirala, *Member, IEEE,* Behnood Rasti, *Senior Member, IEEE,* and Paul Scheunders, *Senior Member, IEEE*

***Abstract*—Transformers have intrigued the vision research community with their state-of-the-art performance in natural language processing. With their superior performance, transformers have found their way in the field of hyperspectral image classification and achieved promising results. In this article, we harness the power of transformers to conquer the task of hyperspectral unmixing and propose a novel deep unmixing model with transformers. We aim to utilize the ability of transformers to better capture the global feature dependencies in order to enhance the quality of the endmember spectra and the abundance maps. The proposed model is a combination of a convolutional autoencoder and a transformer. The hyperspectral data is encoded by the convolutional encoder. The transformer captures long-range dependencies between the representations derived from the encoder. The data are reconstructed using a convolutional decoder. We applied the proposed unmixing model to three widely used unmixing datasets, i.e., Samson, Apex, and Washington DC mall and compared it with the state-of-the-art in terms of root mean squared error and spectral angle distance. The source code for the proposed model will be made publicly available at https://github.com/preetam22n/DeepTrans-HSU.**

***Index Terms*—Hyperspectral image, unmixing, convolutional neural network, deep learning, transformer network, abundance map, endmember extraction, blind unmixing**

I. Introduction

ADVANCES in remote sensing technology improved environmental monitoring, e.g., for tracking rapid environmental changes and take precautionary actions. In particular, hyperspectral imaging (HSI) has attracted much attention in recent years. Its tasks include but are not limited to land used and land cover classification [1]–[3], forest applications [4], [5] and target detection [6] etc. In hyperspectral remote sensing, each spectral pixel might cover several pure materials on the ground due to its limited spatial resolution. The acquired spectral reflectance is then a mixture of the pure spectra (endmembers) of the materials within the pixel [7], [8]. Spectral unmixing techniques estimate the relative proportions (fractional abundances) of the endmembers within spectral pixels. The primary goal of spectral unmixing methods is to extract/estimate endmembers and their fractional abundances in each pixel by only utilizing the observed hyperspectral image. However, this often relies on the presence of a spectral library or the estimation/extraction of endmembers, i.e., pure spectral pixels that span the abundance subspace.

In remote sensing applications, it is generally assumed that the spectra of the pure materials are mixed linearly and several linear unmixing techniques have been developed [9]. When the endmembers of the hyperspectral image are available, the fractional abundances can be estimated by minimizing the least squared errors between the actual reflectance spectra and the ones, reconstructed by the linear model. To have a physical interpretation of the estimated fractional abundances, one must assume that no endmember can have a negative abundance. This constraint is often described as the abundance non-negativity constraint (ANC). The second constraint is the abundance sum-to-one constraint (ASC), i.e., the observed reflectance spectrum is completely composed of endmember contributions. The fully constrained least squares unmixing algorithm (FCLSU) [10] obeys both ANC and ASC. The hyperspectral pixels that follow the fully constrained linear mixing model lie on a linear simplex whose corners (vertices) are given by the endmembers. As a result, many endmember extraction algorithms have been proposed to maximize the volume enclosing simplex in the hyperspectral dataset [11]–[14]. When endmembers are not available in the hyperspectral image (no pure pixel-scenario), virtual endmembers can be estimated by seeking the minimum volume linear simplex, which encloses the data points [15], [16].

Spectral unmixing techniques that can simultaneously estimate the endmembers and the abundances are referred to as blind unmixing techniques [17]–[21]. These methods formulate the unmixing problem as a nonconvex optimization problem with respect to both endmembers and abundances. A common practice is to induce a geometrical penalty term in the fully constrained least squares method. In [22], the Euclidean distances between the estimated endmembers and the center of the hyperspectral pixels were selected to form a geometrical penalty term. In [21], the Euclidean distances between the estimated endmembers and endmembers extracted by Vertex Component Analysis (VCA) were selected for the penalty term. The total variation (TV) of all estimated endmembers was considered in [23] as a geometrical penalty. The optimization equation of these methods contains a regularization parameter, which denotes the trade-off between the geometrical penalty term and the fidelity term. This parameter is data-dependent, and selecting a proper parameter for each

This work is supported by the Research Foundation-Flanders - Project G031921N.

P. Ghosh and S. K. Roy are with the Department of Computer Science and Engineering, Jalpaiguri Engineering College, West Bengal 735102, India (e-mail: pg2202@cse.jgec.ac.in; swalpa@cse.jgec.ac.in).

B. Koirala and P. Scheunders are with Imec-Visionlab, University of Antwerp (CDE) Universiteitsplein 1, B-2610 Antwerp, Belgium (e-mail: bikram.koirala@uantwerpen.be; paul.scheunders@uantwerpen.be).

B. Rasti is with Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, Machine Learning Group, Chemnitzer Straße 40, 09599 Freiberg, Germany; (e-mail: b.rasti@hzdr.de).

hyperspectral image is a highly complex problem. To tackle this challenge, in [24] an automatic parameter selection technique was proposed.

Blind unmixing methods can accurately estimate endmembers, if sufficient hyperspectral pixels are available on the facets of the data simplex. When the spectral pixels are highly mixed, the estimated endmembers are not satisfactory, which leads to poor abundance maps. To deal with highly mixed scenarios, sparse unmixing techniques have been proposed [25]–[27]. Sparse unmixing utilizes a rich and well-designed library of pure spectra and applies sparse regression for the abundance estimation. A major challenge is to correct mismatches between the real reflectance spectra and the library spectra, caused by differences in the acquisition conditions of the two data types.

Due to the success of deep learning-based networks in machine learning and computer vision applications [28], [29], recently, a variety of deep neural networks has been proposed for hyperspectral unmixing. These networks are mainly based on variations of deep encoder-decoder networks. The inputs of these networks are the reflectance spectra, while the outputs are the reconstructed spectra. The encoder transforms the input spectra to the fractional abundances while the decoder transforms the abundances to the reconstructed spectra using linear layers, with the endmembers as the weights. In [30], autoencoders that have been used for hyperspectral unmixing are grouped into five different categories: (a) Sparse nonnegative autoencoders (a stack of nonnegative sparse autoencoders (SNSA)) [31] (b) Variational autoencoders (Deep AutoEncoder Network (DAEN) [32], Deep Generative Unmixing algorithm (DeepGUn) [33] (c) Adversarial autoencoders (Adversarial autoencoder network (AAENet)) [34]–[36] (d) Denoising autoencoders (an untied Denoising Autoencoder with Sparsity (uDAS)) [37], and (e) Convolutional autoencoders [38]–[40]. Although the advantage of incorporating the spatial information for hyperspectral unmixing has been demonstrated in the literature (especially for homogeneous regions), in SNSA, DAEN, DeepGUn, and uDAS, the spatial information is ignored. Several convolutional autoencoder-based unmixing techniques have been proposed to effectively incorporate the spatial correlation between adjacent pixels. In [41], a supervised hyperspectral unmixing method (i.e., the endmembers are assumed to be known) was proposed using a 3D convolutional autoencoder. The method referred to as unmixing using deep image prior (UnDIP) [42] utilizes endmembers extracted by a simplex volume maximization (SiVM) technique. Although several deep learning-based unmixing techniques have been specifically designed for blind unmixing, most of the methods fail when pure pixels are not available in the hyperspectral image. This is because they do not exploit the geometrical properties of the linear simplex. Recently, a minimum simplex convolutional network (MiSiCNet) [43] was proposed to incorporate both the spatial correlation between adjacent pixels and the geometrical properties of the linear simplex.

A. Contributions and Novelties

HSI, being complex in nature, pose a big challenge for Convolutional Neural Networks (CNN). As a convolution operation is limited to local features determined by the dimension of the kernel size, a significant amount of contextual information present in the original HS image is lost. Most autoencoders (AEs) are purely based on CNN networks and therefore fail to preserve a substantial portion of the original information due to the limited dimensionality of the latent space. That poses an even more significant problem in the case of HSI unmixing because the final number of endmembers is considerably lower than the initial number of spectra, causing a lot of contextual information to be lost. To address this issue, a transformer [44], [45] will be utilized that can recover some of the lost information, owing to its ability to capture global contextual feature dependencies [46]. For this, the AE output is rearranged in terms of patches. Inspired by [47], we propose a new attention mechanism, called Multihead Self-Patch Attention to calculate the long-range dependencies between these patches. This leads to better quality abundance maps and an overall better unmixing result, which in turn helps the decoder to better reconstruct the HSI. Since the weights of the decoder are used to obtain the endmember spectra, a better quality of extracted endmembers is obtained. The contribution of the proposed methodology to this end is summarized below:

- We propose a new unmixing method based on a combination of a convolutional autoencoder and a transformer. The transformer is applied to the latent space of the autoencoder to enhance the feature extraction and to ensure a better estimation of abundances and endmembers. For this, the AE output is rearranged into patches.
- Inside the transformer encoder, we propose a new attention mechanism which is referred to as Multihead Self-Patch Attention. The attention modules of the multihead self-patch attention find the global contextual feature dependencies by determining the long-range relationship between the image patches.
- To estimate the endmembers, we apply a single convolution layer whose weights are initialized by VCA. The weights are learned and improved during the training of the model to obtain endmember spectra of superior quality.

The remaining of the paper is organized as follows: Section II introduces the components of the proposed method including the novel Multihead Self-Patch Attention for transformer based deep HS image unmixing. In Section III, extensive experiments are conducted with three benchmark datasets, and a hyperparameter sensitivity analysis and discussions are provided. Finally, comprehensive conclusions are drawn in Section IV.

II. Proposed Methodology

Let the HSI of spatial dimensions $H \times W$ with B spectral bands be denoted by $\mathbf{I} \in \mathbb{R}^{B \times H \times W}$. The HSI can be reshaped to produce the matrix $\mathbf{Y} \in \mathbb{R}^{B \times n}$, where $n = H \cdot W$ is the number of hyperspectral pixels. The endmember matrix will be denoted by $\mathbf{E} \in \mathbb{R}^{B \times R}$ where R represents the number of

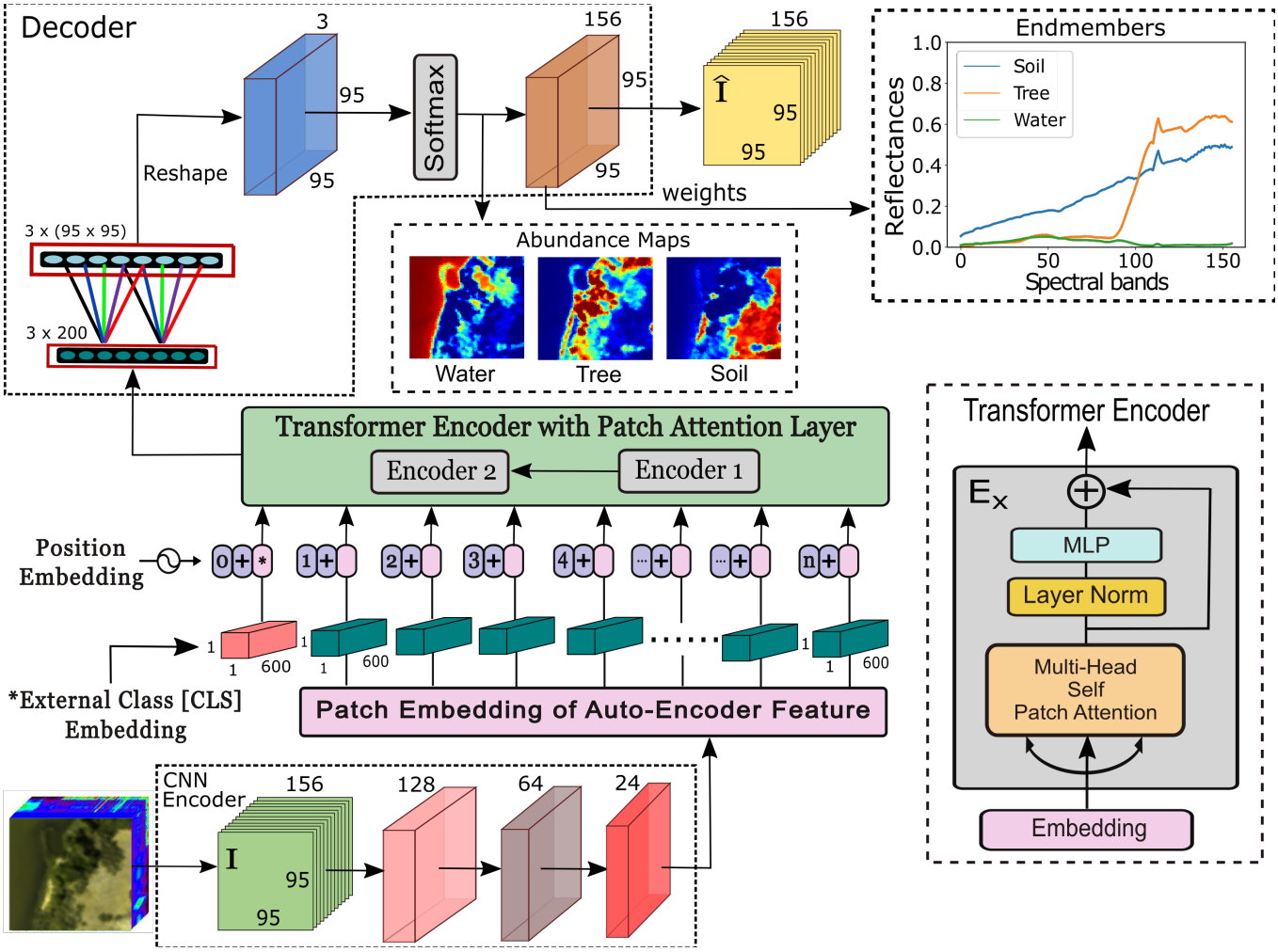


Fig. 1: Graphical representation of the proposed deep hyperspectral unmixing model.

endmembers present in the HSI. The corresponding abundance cube (i.e., the stack of R abundance maps) is represented by $\mathbf{M} \in \mathbb{R}^{R \times H \times W}$. The abundance cube can be reshaped to produce the matrix $\mathbf{A} \in \mathbb{R}^{R \times n}$.

A. Problem formulation

In the Linear Mixing Model (LMM), the observed spectral reflectance is formulated as:

$$\mathbf{Y} = \mathbf{E}\mathbf{A} + \mathbf{N} \tag{1}$$

where $\mathbf{N} \in \mathbb{R}^{B \times n}$ is the additive noise present in $\mathbf{Y}$. Generally, three physical constraints should be satisfied: 1) the endmember matrix should be non-negative $\mathbf{E} \geq 0$; 2) ANC (Eq. (2a)); and 3) ASC (Eq. (2b)):

$$\mathbf{A} \geq 0 \tag{2a}$$

$$\mathbf{1}_R^T \mathbf{A} = \mathbf{1}_n^T \tag{2b}$$

where $\mathbf{1}_n$ indicates an n-component column vector of ones.

Since spectal unmixing is a reconstruction problem, in which abundance maps are reconstructed from the given HSI, AEs can be applied. AEs are quite capable at reconstructing and extracting information from the given inputs. In this work, the performance of an AE is complemented by the use of a transformer, to significantly improve the quality of the generated abundance maps and consequently the extracted spectral signatures of the endmembers. Fig. 1 illustrates the proposed model for deep HSI unmixing. The components of the model are discussed in detail in subsections II-B through II-E.

B. Hyperspectral feature extraction using AE

AEs encode the input into a latent space with a lower dimensionality, learning only the salient features within the input image while avoiding unnecessary details. Owing to CNNs ability to extract high-level abstract features, using them in the encoder part of an AE provides a twofold benefit. Firstly, it heavily reduces the large number of spectral bands of a HSI and secondly, it extracts discriminative high-level features that form the base for the transformer in the next step.

The CNN applied in the encoder block of the proposed model contains three layers. Each layer progressively reduces the number of spectral bands of the HSI until C spectral bands remain. The value of C is a hyperparameter to be set. As the convolutional layer is primarily used to reduce the number of channels of the input HSI, a kernel size of 1×1 is used to

keep the number of parameters low and to facilitate a faster training of the model. All three layers use a 2D convolution operation followed by a batch normalization (BN). To mitigate the vanishing gradient problem of the network, the first layer uses a dropout function. To introduce non-linearity, Leaky ReLU is used in the output of the first two layers of the AE. Table I summarizes the structure of the encoder.

TABLE I: Layerwise summary of the Encoder block where B represents the number of spectral bands and C is the number of output bands.

Layers	Composition	Kernel	Bands in	Bands out
Layer 1	Conv 2D BN Dropout Leaky ReLU	(1×1)	B	128
Layer 2	Conv 2D BN Leaky ReLU		128	64
Layers 3	Conv 2D BN		64	C

In the encoder, the HSI $\mathbf{I} \in \mathbb{R}^{B \times H \times W}$ is transformed by the three consecutive layers of the encoder block into $\mathbf{I}' \in \mathbb{R}^{H \times W \times C}$:

$$\begin{aligned} \mathbf{I_1} &= f_1(\mathbf{W_1 I} + \mathbf{U_1}) \\ \mathbf{I_2} &= f_2(\mathbf{W_2 I_1} + \mathbf{U_2}) \\ \mathbf{I_3} &= f_3(\mathbf{W_3 I_2} + \mathbf{U_3}) \\ \mathbf{I}' &= \mathbf{I_3^T} \end{aligned} \tag{3}$$

where $f_1(\cdot)$, $f_2(\cdot)$ and $f_3(\cdot)$ denote the three encoder layers and $\mathbf{W_1}, \mathbf{W_2}, \mathbf{W_3}$ and $\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}$ are the weights and biases, respectively of each layer. The superscript T denotes the matrix transpose operation.

C. Patch and Position Embeddings

To efficiently capture the long range feature dependencies, the AE output is rearranged in terms of patches. The output of the AE encoder is the cube $\mathbf{I}'$ of dimension $(H \times W \times C)$ where H, W are the spatial dimensions and C represents the reduced number of bands of the output. These features are grouped in patches $((m \cdot p) \times (n \cdot p) \times C)$ where p is the patch size and $m \cdot n$ is the total number of patches. Then the cube is reshaped to a matrix $\mathbf{X_{patch}}$ of size $((m \cdot n) \times (p \cdot p \cdot C))$ = $(N' \times D)$ where N' is the total number of patches and D is the dimension of each patch embedding. As an example, for the Samson dataset (Section III-A1), with $p = 5$ and $C = 24$, the rearrangement is given as:

$$\begin{aligned} \mathbf{I}' &= (95 \times 95 \times 24) \\ &= ((19 \cdot 5) \times (19 \cdot 5) \times 24) \\ &\rightarrow \\ \mathbf{X_{patch}} &= ((19 \cdot 19) \times (5 \cdot 5 \cdot 24)) \\ &= (361 \times 600) \end{aligned}$$

In a next step, learnable class tokens $\mathbf{X_{cls}}$ of dimensions $(1 \times D)$ are defined, in which the transformer encoder will capture the long range semantic information of the patch tokens. Moreover, positional tokens $\mathbf{X_{pos}}$ of shape $(N \times D)$, with $N = N' + 1$ are generated to retain patch positional information. Rather than providing pixel and patch positional information, the positional tokens will be learned by the transformer encoder as well. Both are randomly initialized.

$\mathbf{X_{cls}}$ is appended as an extra row to the matrix $\mathbf{X_{patch}}$ and $\mathbf{X_{pos}}$ is added to the feature embedding:

$$\mathbf{X}' = (\mathbf{X_{cls}} \parallel \mathbf{X_{patch}}) + \mathbf{X_{pos}} = (\mathbf{X'_{cls}} \parallel \mathbf{X'_{patch}}) \tag{4}$$

with $\parallel$ the concatenation operation.

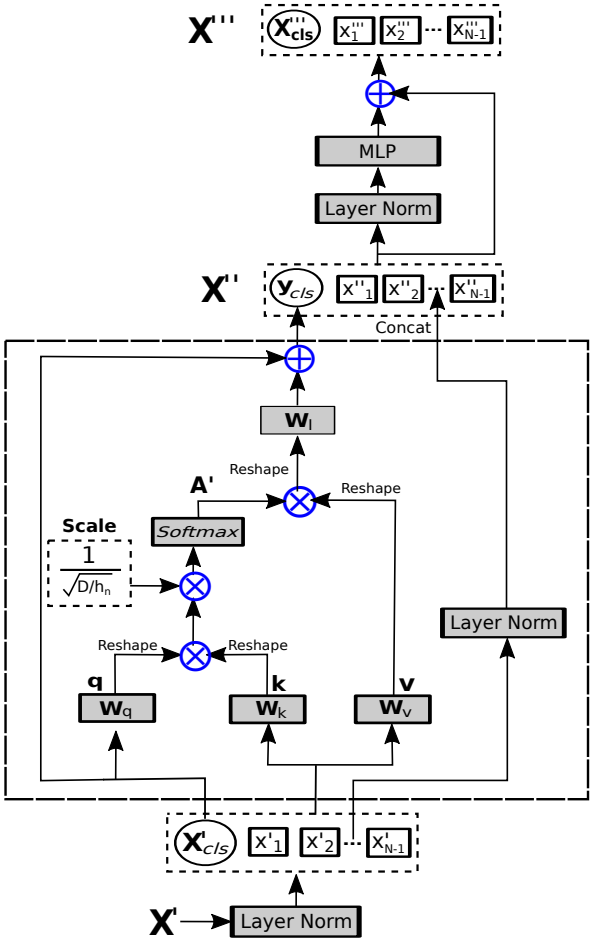


Fig. 2: Transformer Encoder with Multihead Self-Patch Attention.

D. Transformer Encoder with Multihead Self-Patch Attention

$\mathbf{X}'$ is the input of the next phase, which is composed of one or several transformer encoders. Each transformer encoder contains a Multihead Self-Patch Attention network [44]. The goal of this network is the exchange of information within the patch tokens to capture their long range contextual information and to feed this into the class token. To preserve the overall patch structure, the patch tokens are appended again to the learned class token. Fig. 2 depicts the proposed Multihead Self-Patch Attention network. A detailed description of each step is given below.

Step 1: In a fist step, the overall patch matrix $\mathbf{X}'$ enters the self attention block of the transformer after going through a layer normalisation step. Attention is calculated by three linear layers. One layer works on the class token only (weight $\mathbf{W_q}$ and output $\mathbf{q}$ of size $(1 \times D)$). The other 2 layers work on the entire patch matrix (weights $\mathbf{W_k}$ and $\mathbf{W_v}$ and outputs $\mathbf{k}$ and $\mathbf{v}$, both of size $(N \times D)$):

$$\mathbf{q} = \mathbf{W_q}\mathbf{X'_{cls}}, \quad \mathbf{k} = \mathbf{W_k}\mathbf{X}', \quad \mathbf{v} = \mathbf{W_v}\mathbf{X}'$$

Step 2: In the next step, the attention weight ($\mathbf{A}$) is calculated by computing the pairwise similarity between $\mathbf{q}$ and $\mathbf{k}$ and applying a softmax function:

$$\mathbf{A} = \texttt{softmax}(\mathbf{q}\mathbf{k}^\mathbf{T}/\sqrt{\mathbf{D}})$$

The scaling term $(1/\sqrt{D})$ counteracts the small gradients of the softmax function. The self-patch attention (PA) is then computed as:

$$\texttt{PA}(\mathbf{X}') = \mathbf{A}\mathbf{v} \tag{5}$$

To further enhance the relationships among the different patches, self-patch attention with multiple heads is applied. For this, $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ have to reshape into matrices $\mathbf{q}'$, $\mathbf{k}'$, and $\mathbf{v}'$ of size $(h_n \times D/h_n)$, $((N \cdot h_n) \times D/h_n)$, $((N \cdot h_n) \times D/h_n)$ respectively, where h_n denotes the number of heads (attention modules). Then, the attention weight becomes:

$$\mathbf{A}' = \texttt{softmax}(\mathbf{q}'\mathbf{k}'^\mathbf{T}/\sqrt{\mathbf{D}/\mathbf{h_n}})$$

The self-patch attention with multiple heads (MPA) is then computed as:

$$\texttt{MPA}(\mathbf{X}') = \mathbf{A}'\mathbf{v}' \tag{6}$$

Step 3: The output of MPA is a matrix of size $(h_n \times D/h_n)$, and is then reshaped back to a matrix of size $(1 \times D)$. This matrix is further passed through a linear layer (weights $\mathbf{W_l} \in \mathbb{R}^{D \times D}$) and added up with the original class token $\mathbf{X'_{cls}}$ to obtain the class token $\mathbf{y_{cls}}$:

$$\mathbf{y_{cls}} = \texttt{MPA}(\mathbf{X}')\mathbf{W_l} + \mathbf{X'_{cls}} \tag{7}$$

Step 4: Finally, $\mathbf{y_{cls}}$ is concatenated with the layer normalised patch tokens to obtain the output of the attention network $\mathbf{X}''$:

$$\mathbf{X}'' = \mathbf{y_{cls}} \parallel \texttt{LN}(\mathbf{X'_{patch}}) \tag{8}$$

As the output of the Multihead Self-Patch Attention network, the feature embedding $\mathbf{X}''$ is passed through a normalization layer and then fed into an Multi Layered Perceptron (MLP) block along with a residual connection to obtain the final output of the transformer encoder block (see bottom right of Fig. 1):

$$\mathbf{X}''' = \mathbf{X}'' + \texttt{MLP}(\texttt{LN}(\mathbf{X}'')) \tag{9}$$

Any number of such transformer encoders can be applied sequentially. In this work, two encoders have been applied. The output of the final block is used for further processing down the line.

The pseudo code of the Transformer Encoder with Multihead Self-Patch Attention, is shown in Algorithm 1.

Algorithm 1: Transformer Encoder with Multihead Self-Patch Attention

Input: $\mathbf{X}'$, $\mathbf{X'_{cls}}$, $\mathbf{X'_{patch}}$, D, h_n
Output: $\mathbf{X'''_{cls}}$
Multihead Self-Patch Attention (Begin)
Step 1. $\mathbf{q} = \mathbf{W_q}\mathbf{X'_{cls}}, \quad \mathbf{k} = \mathbf{W_k}\mathbf{X}', \quad \mathbf{v} = \mathbf{W_v}\mathbf{X}'$,
$\mathbf{q}' = \text{reshape}(\mathbf{q}), \mathbf{k}' = \text{reshape}(\mathbf{k})$,
$\mathbf{v}' = \text{reshape}(\mathbf{v})$
Step 2. $\mathbf{A}' = \texttt{softmax}(\mathbf{q}'\mathbf{k}'^\mathbf{T}/\sqrt{\mathbf{D}/\mathbf{h_n}})$,
$\texttt{MPA}(\mathbf{X}') = \mathbf{A}'\mathbf{v}'$ (6)
Multihead Self-Patch Attention (End)
Step 3. $\mathbf{y_{cls}} = \text{reshape}(\texttt{MPA}(\mathbf{X}'))\mathbf{W_l} + \mathbf{X'_{cls}}$ (7)
Step 4. $\mathbf{X}'' = \mathbf{y_{cls}} \parallel \texttt{LN}(\mathbf{X'_{patch}})$ (8),
$\mathbf{X}''' = \mathbf{X}'' + \texttt{MLP}(\texttt{LN}(\mathbf{X}''))$ (9),
$\mathbf{X'''_{cls}} = \mathbf{X}'''(1, :)$

E. Unmixing with decoder

The transformer produces the results $\mathbf{X}''' \in \mathbb{R}^{N \times D}$, where N is the total number of tokens and D is the dimension of each token. However, for the purpose of unmixing, only the class token $\mathbf{X'''_{cls}}$ (i.e., the first row of $\mathbf{X}'''$) of size $(1 \times D)$ is considered and forwarded to the upsampling block. To do so, we reshape $\mathbf{X'''_{cls}}$ to a matrix of size $R \times (D/R)$, and then upscale it to size $R \times (H \cdot W)$. Upscaling from a relatively small dimension of D/R to the dimensions $H \cdot W$ introduces noise in the final output. To solve this issue, a convolution operation with parameters $kernel\_size = (3 \times 3)$, $stride = 1$, $padding = 1$ is used. Finally, a reshaping operation is carried out to convert the output to the shape of the adundance cube $\mathbf{M}$ i.e., $(R \times H \times W)$. To ensure that the ANC and ASC constraints (Eqs. (2a) and (2b)) are satisfied, a softmax layer is used along the R dimension.

To calculate the endmembers, the abundance matrix $\mathbf{M}$ is passed through the decoder block of the AE which consists of a single convolutional layer. This convolution operation increases the number of bands in $\mathbf{M}$ from R to B, to obtain the reconstructed HSI $\hat{\mathbf{I}}$. The weights of the convolution layer, which are initialized with the endmembers obtained from VCA, are optimized to estimate the final endmembers $\hat{\mathbf{E}} \in \mathbb{R}^{\mathbf{B} \times \mathbf{R}}$.

F. Losses and Optimization functions

In order to train the proposed model, a combination of two different losses: *Reconstruction Error (RE) loss* and *Spectral Angle Distance (SAD) loss* were applied:

$$L_{RE}(\mathbf{I}, \hat{\mathbf{I}}) = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} (\hat{\mathbf{I}}_{\mathbf{ij}} - \mathbf{I}_{\mathbf{ij}})^2 \tag{10}$$

$$L_{SAD}(\mathbf{I}, \hat{\mathbf{I}}) = \frac{1}{R} \sum_{i=1}^{R} \arccos\left(\frac{\left\langle \mathbf{I}_i, \hat{\mathbf{I}}_i \right\rangle}{\|\mathbf{I}_i\|_2 \|\hat{\mathbf{I}}_i\|_2}\right) \tag{11}$$

The *RE* loss is calculated by the Mean Squared Error (MSE) objective function and helps the encoder part to learn only the essential features of the input HSI while discarding non-essential details. The *SAD* loss is a scale invariant objective function. MSE discriminates between endmembers, based on their absolute magnitude which is not desirable in case of HSI

unmixing. Including SAD loss helps to counter this drawback of the MSE objective function and makes the overall model converge much faster. The total loss is calculated as the weighted sum of these two losses:

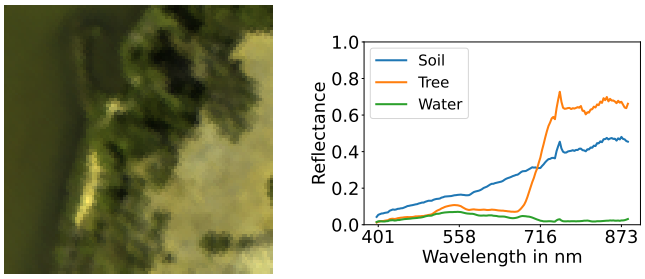
$$L = \beta L_{RE} + \gamma L_{SAD} \tag{12}$$

with regularization parameters β and γ.

III. Experimental Results

A. Hyperspectral Data Description

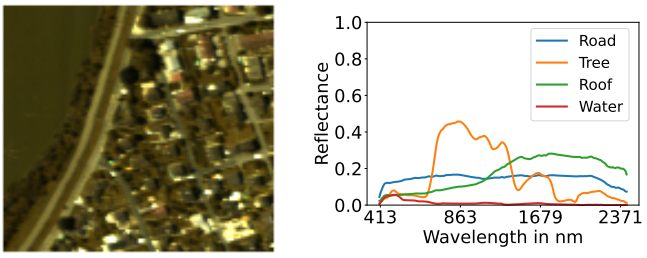
We performed experiments on three datasets. The description of the datasets are given below.



(a) (b)

Fig. 3: Samson image: (a) True-color image (Red: 571.01 nm, Green: 539.53 nm, and Blue: 432.48 nm) (b) Endmembers.

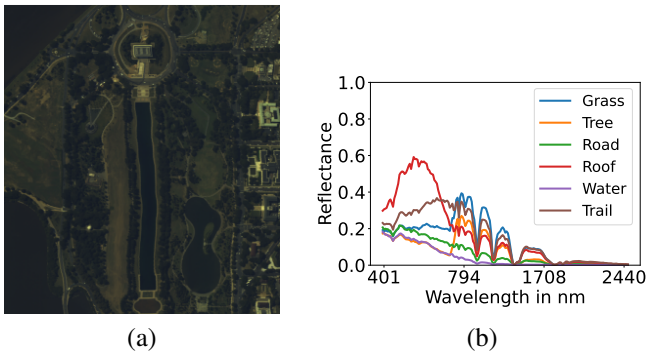
1) Samson: The Samson hyperspectral dataset ([48]) (Fig. 3(a)) utilized in this work contains 95×95 hyperspectral pixels. Each hyperspectral pixel contains reflection values from 156 bands covering the wavelength range [401–889] nm. In this hyperspectral image, there are three endmembers (i.e., Soil, Tree, and Water). The ground truth endmember spectra (see Fig. 3(b)) were manually selected from the image, and ground truth abundance maps were produced by applying FCLSU.



(a) (b)

Fig. 4: Apex image: (a) True-color image (Red: 572.2 nm, Green: 532.3 nm, Blue: 426.5 nm); (b) Endmembers.

2) Apex: Fig. 4(a) shows a cropped image of the Apex dataset ([49]), as used in this work. This image contains 110×110 hyperspectral pixels. Each hyperspectral pixel contains reflection values from 285 bands covering the wavelength range [413–2420] nm. There are four endmembers (i.e., Water, Tree, Road, and Roof) in this hyperspectral image. The ground truth endmember spectra (see Fig. 4(b)) were manually selected from the image, and ground truth abundance maps were produced by applying FCLSU.



(a) (b)

Fig. 5: Washington DC Mall image: (a) True-color image (Red: 572.7 nm, Green: 530.1 nm, Blue: 425.0 nm); (b) Endmembers.

3) Washington DC Mall: This hyperspectral image is acquired over the Washington DC Mall using the HYDICE sensor [1]. Fig. 5(a) shows the cropped data used in this paper that contains 290 × 290 pixels. Each hyperspectral pixel contains reflection values from 191 bands covering the wavelength range [400–2400] nm. There are six endmembers (i.e., Grass, Tree, Roof, Road, Water, and Trail) in this hyperspectral image. The ground truth endmember spectra (Fig. 5(b)) were manually selected from the image, and ground truth abundance maps were produced by applying FCLSU.

B. Experimental Setup

The performance of the proposed model is evaluated and compared to six different unmixing techniques from different categories: **Geometrical unmixing** method `FCLSU` [10] using VCA [12] for endmember extraction, **Geometrical and blind unmixing** method `NMF-QMV` [24], **Sparse unmixing** method Collaborative LASSO (`Collab`) [50] and **Deep unmixing** methods `uDAS` [37], `UnDIP` [42], and `CyCUNet` [40].

C. Hyperparameters

In deep unmixing models, the produced results are typically largely dependent on the hyperparameter settings. Choosing proper values for the hyperparameters can significantly improve results. Table II shows the hyperparameters used for training the proposed model, which are further discussed below:

Samson dataset: The patch size p was selected to be (5×5) and the transformer input dimensionality C was chosen to be 24. The values 4×10^3 and 5×10^{-3} were used for the regularization parameters β and γ respectively. The model was trained during 200 epochs with an initial learning rate of 6×10^{-3}, which was reduced by 20% after every 15 epochs. A weight decay rate of 4×10^{-5} was incorporated in the optimization function to keep the losses in check.

Apex dataset: The patch size p was selected to be (5×5) and the transformer input dimensionality C was chosen to be 32. Values of 4×10^3 and 5×10^{-2} were used for the regularization parameters β and γ respectively. The model

[1] https://engineering.purdue.edu/ biehl/MultiSpec/hyperspectral.html

was trained during 200 epochs with an initial learning rate of 9×10^{-3}, which was reduced by 20% after every 15 epochs. A weight decay rate of 4×10^{-5} was incorporated in the optimization function to keep the losses in check.

Washington DC Mall dataset: The patch size p was selected to be (10×10) and the transformer input dimensionality C was chosen to be 24. Values 5×10^3 and 2×10^{-3} were used for the regularization parameters β and γ respectively. The model was trained during 150 epochs with an initial learning rate of 6×10^{-3}, which was reduced by 20% after every 15 epochs. A weight decay rate of 3×10^{-5} was incorporated in the optimization function to keep the losses in check.

TABLE II: Hyperparameters used for training the proposed model.

Hyperparameters	Samson	Apex	WDC Mall
p	(5×5)	(5×5)	(10×10)
C	24	32	24
β	5×10^3	5×10^3	5×10^3
γ	3×10^{-2}	5×10^{-2}	1×10^{-4}
Epoch	200	200	150
Learning rate	6×10^{-3}	9×10^{-3}	6×10^{-3}
Weight decay	4×10^{-5}	4×10^{-5}	3×10^{-5}

D. Quantitative Performance Measures

Quantitative results are provided by the root mean squared error (RMSE) between the estimated and ground truth abundance fractions:

$$\text{RMSE}(\mathbf{M}, \hat{\mathbf{M}}) = \sqrt{\frac{1}{RHW} \sum_{k=1}^{R} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(\hat{\mathbf{M}}_{kij} - \mathbf{M}_{kij} \right)^2} \tag{13}$$

and the spectral angle distance (SAD) in degree between the estimated and ground truth endmembers:

$$\text{SAD}(\mathbf{S}, \hat{\mathbf{S}}) = \frac{1}{R} \sum_{i=1}^{R} \arccos \left(\frac{\left\langle \mathbf{s}_{(i)}, \hat{\mathbf{s}}_{(i)} \right\rangle}{\left\| \mathbf{s}_{(i)} \right\|_2 \left\| \hat{\mathbf{s}}_{(i)} \right\|_2} \right), \tag{14}$$

where $\langle . \rangle$ denotes the inner product and $\mathbf{s}_{(i)}$ indicates the ith column of the ground truth endmembers matrix $\mathbf{S}$.

E. Unmixing Experiments: Quantitative Results

Samson Dataset: Quantitative results on the Samson dataset can be found in Tables III and IV. The results confirm that the proposed model outperforms the other techniques in terms of both abundance and endmember estimation with a mean RMSE of 0.0783 showing a 48.02% improvement to the next best method and a mean SAD value of 0.0608 which amounts to a 35.93% improvement.

Apex Dataset: Quantitative results on the Apex dataset can be found in Tables V and VI. The endmember "Road" of the Apex dataset is found to be quite a challenge for the other methods, while the proposed method estimates this endmember satisfactorily. The proposed method outperforms the other unmixing techniques with a mean RMSE value of 0.1264 and a mean SAD value of 0.0867. Additionally, it provides the best endmember estimation for Road and Water in terms of SAD.

TABLE III: RMSE (Samson Dataset). The best performances are shown in bold.

	CyCU	Collab	FCLSU	NMF	UnDIP	uDAS	Proposed
Soil	0.2417	0.1506	0.1766	0.2011	0.1778	0.1799	**0.0712**
Tree	0.1386	**0.0607**	0.0653	0.1466	0.1330	0.1383	0.0683
Water	0.2654	0.1181	0.1492	0.2063	0.2096	0.2303	**0.0930**
Overall	0.2222	0.1159	0.1387	0.1866	0.1763	0.1867	**0.0783**

TABLE IV: SAD (Samson Dataset). The best performances are shown in bold.

	CyCU	Collab	NMF	SiVM	VCA	uDAS	Proposed
Soil	0.1144	0.0155	0.0391	0.0259	0.0259	0.0358	**0.0128**
Tree	0.1517	0.0832	0.1239	0.0748	0.0961	0.0960	**0.0674**
Water	0.2081	0.1402	1.5201	0.1554	0.1554	0.1527	**0.0729**
Overall	0.1581	0.0796	0.5610	0.0854	0.0925	0.0948	**0.0510**

TABLE V: RMSE (Apex Dataset). The best performances are shown in bold.

	CyCU	Collab	FCLSU	NMF	UnDIP	uDAS	Proposed
Road	0.2921	0.3078	0.2331	0.1806	**0.1737**	0.1973	0.1776
Tree	0.2020	0.1907	**0.0944**	0.2468	0.2154	0.1419	0.0993
Roof	0.1630	0.1483	0.1201	0.2359	0.2554	0.2303	**0.1200**
Water	0.1213	**0.0797**	0.1327	0.3751	0.4170	0.2887	0.0902
Overall	0.2046	0.1997	0.1543	0.2692	0.2809	0.2210	**0.1264**

TABLE VI: SAD (Apex Dataset). The best performances are shown in bold.

	CyCU	Collab	NMF	SiVM	VCA	uDAS	Proposed
Road	0.4543	0.6772	0.4003	0.0907	0.6915	0.4551	**0.0836**
Tree	**0.0850**	0.2063	0.2710	0.1339	0.2644	0.1405	0.1295
Roof	0.1298	0.1002	0.1753	**0.0689**	0.1471	0.0860	0.0903
Water	0.6223	0.5137	1.8417	0.5040	0.5176	0.2251	**0.0434**
Overall	0.3228	0.3744	0.6721	0.1994	0.4052	0.2267	**0.0867**

Washington DC Mall Dataset: Quantitative results on the Washington DC Mall dataset can be found in Tables VII and VIII. Among all the considered datasets, the similarity between the spectral signatures of its six endmembers provides the greatest challenge. The "Tree" and "Grass" endmembers have almost identical spectral signatures, and most methods struggle to find the difference. The proposed model successfully separated these two endmembers due to its ability to find long-range dependencies among the image patches, thus leading to a RMSE value of 0.1661 and 0.0963 for the "Grass" and "Tree" endmembers, respectively. In terms of overall RMSE and SAD, the proposed model outperforms the closest competitor method by 43.71% and 52.11% respectively.

Overall observations: From Tables III, V and VII, one can conclude that the overall performance of the proposed method beats the other competing methods by a significant

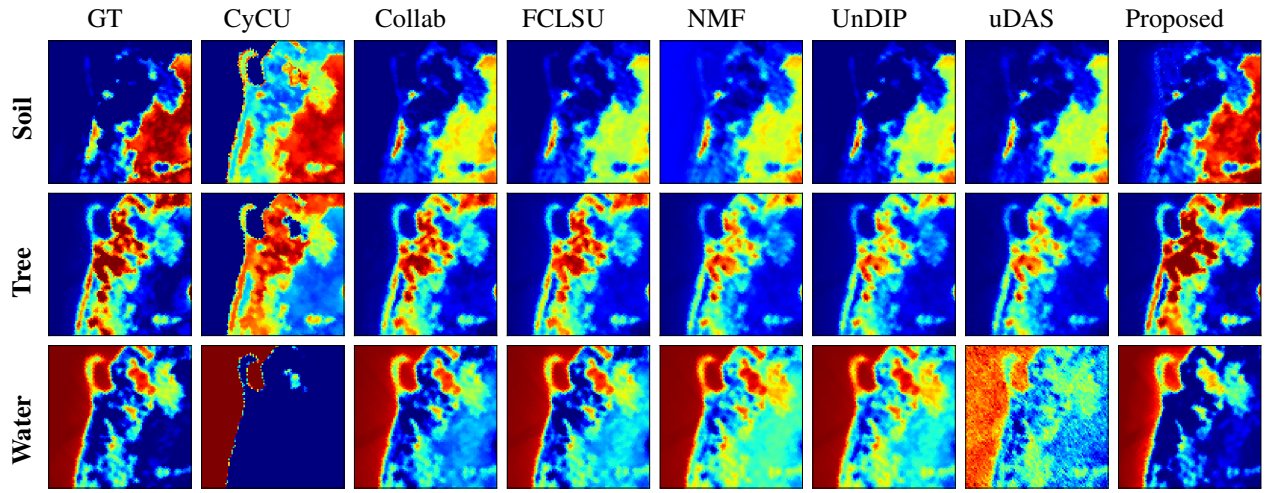


Fig. 6: Samson dataset - Visual comparison of the abundance maps obtained by the different unmixing techniques.

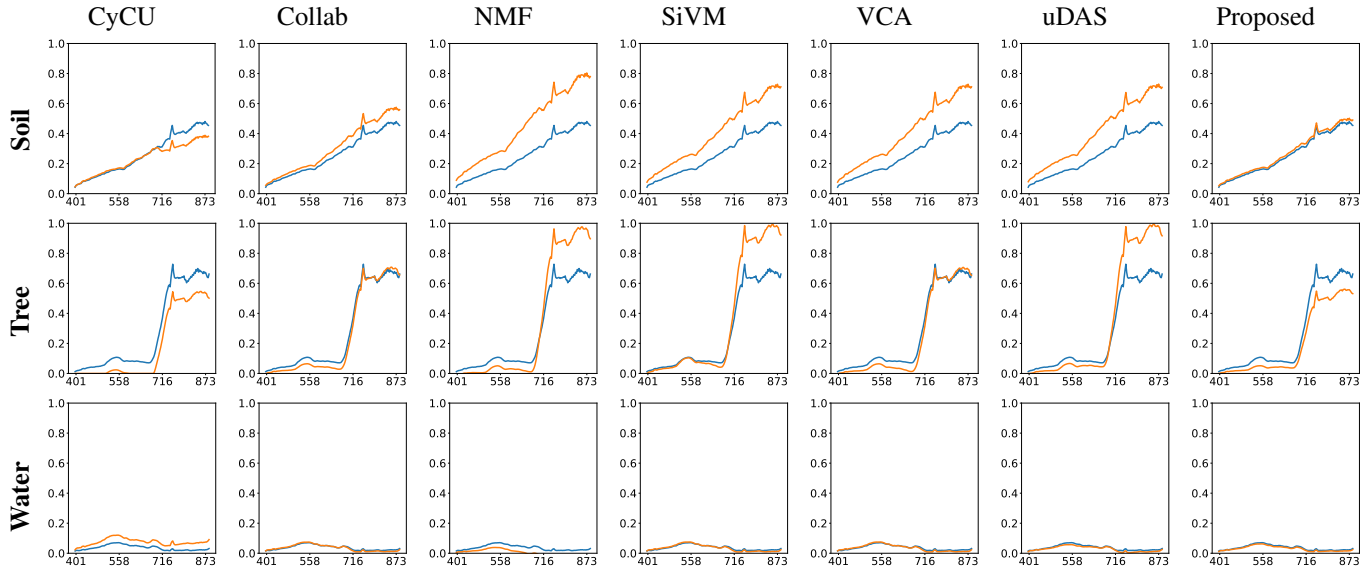


Fig. 7: Samson dataset - Visual comparison of the endmembers obtained by the different unmixing techniques. Blue: ground truth endmembers; Orange: estimated endmembers.

TABLE VII: RMSE (Washington DC Mall Dataset). The best performances are shown in bold.

	CYCU	Collab	FCLSU	NMF	UnDIP	uDAS	Proposed
Grass	0.4104	0.2901	0.3090	0.3624	0.2978	0.3780	**0.1661**
Tree	0.2824	0.4167	0.4025	0.2761	0.3514	0.3351	**0.0963**
Road	0.2545	0.2263	0.1757	0.2351	0.2436	0.2497	**0.1353**
Roof	0.4157	0.0437	**0.0380**	0.0862	0.0493	0.0463	0.0863
Water	0.3957	0.3102	0.2921	0.2076	0.3812	0.5156	**0.1326**
Trail	0.2072	0.1875	0.1230	**0.1011**	0.2360	0.1769	0.1492
Overall	0.3379	0.2715	0.2550	0.2322	0.2814	0.3206	**0.1307**

TABLE VIII: SAD (Washington DC Mall Dataset). The best performances are shown in bold.

	CYCU	Collab	NMF	SiVM	VCA	uDAS	Proposed
Grass	**0.0895**	0.3171	0.1952	0.1851	0.3170	0.1897	0.2379
Tree	0.2704	0.3335	0.4507	0.7258	0.2883	0.4251	**0.1225**
Road	0.4642	0.3439	0.2243	0.8608	0.2316	0.6585	**0.0781**
Roof	0.9500	**0.0331**	0.2078	0.2826	0.0343	0.1992	0.3352
Water	0.4205	**0.0305**	0.6736	0.9495	0.7766	0.2328	0.0533
Trail	0.7906	0.3446	**0.0615**	0.1754	0.6472	0.0940	0.0951
Overall	0.4975	0.2338	0.3022	0.5299	0.3825	0.2999	**0.1537**

margin in terms of RMSE. `Collab`, `FCLSU`, and `NMF` also shows decent performance on the Samson, Apex and WDC Mall datasets, but their performance is not consistent across the datasets. `UnDIP` and `uDAS` were unable to beat any of the methods for any given class; however, their performance was consistent throughout the different datasets used in the experiments. `CyCU` produced mixed results within a given dataset, with good performance on particular endmembers and significantly worse on other endmembers.

Tables IV, VI and VIII make it clear that obtaining good

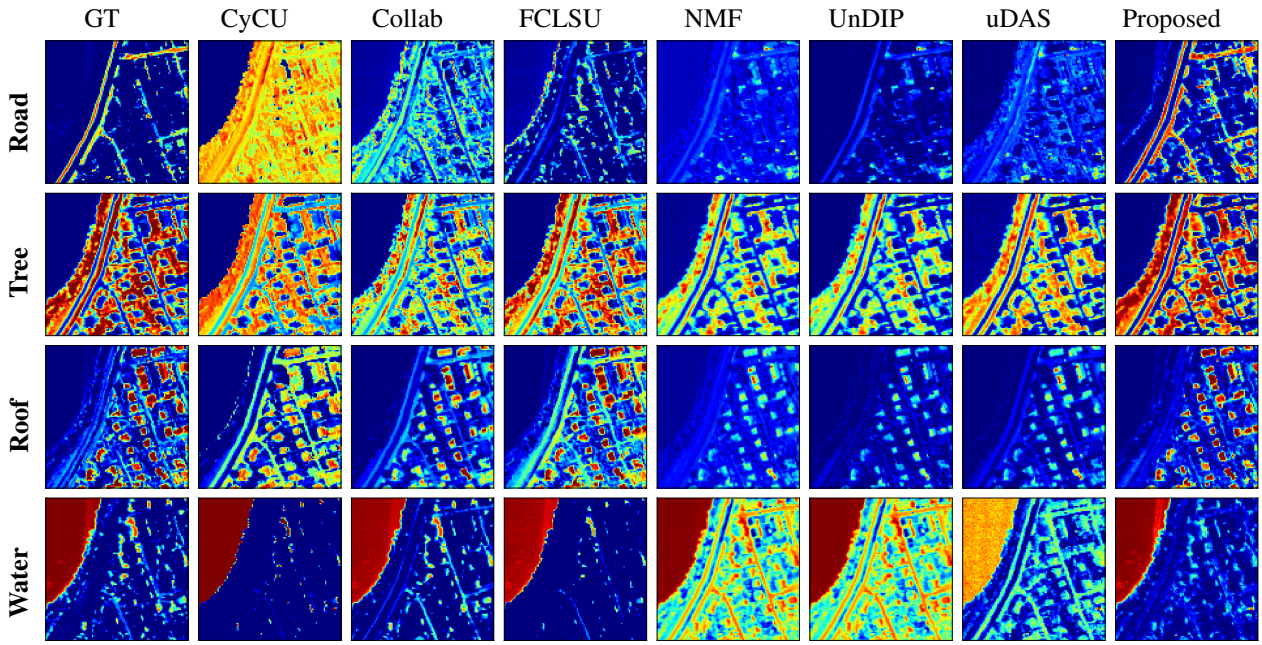


Fig. 8: Apex dataset - Visual comparison of the abundance maps obtained by the different unmixing techniques.

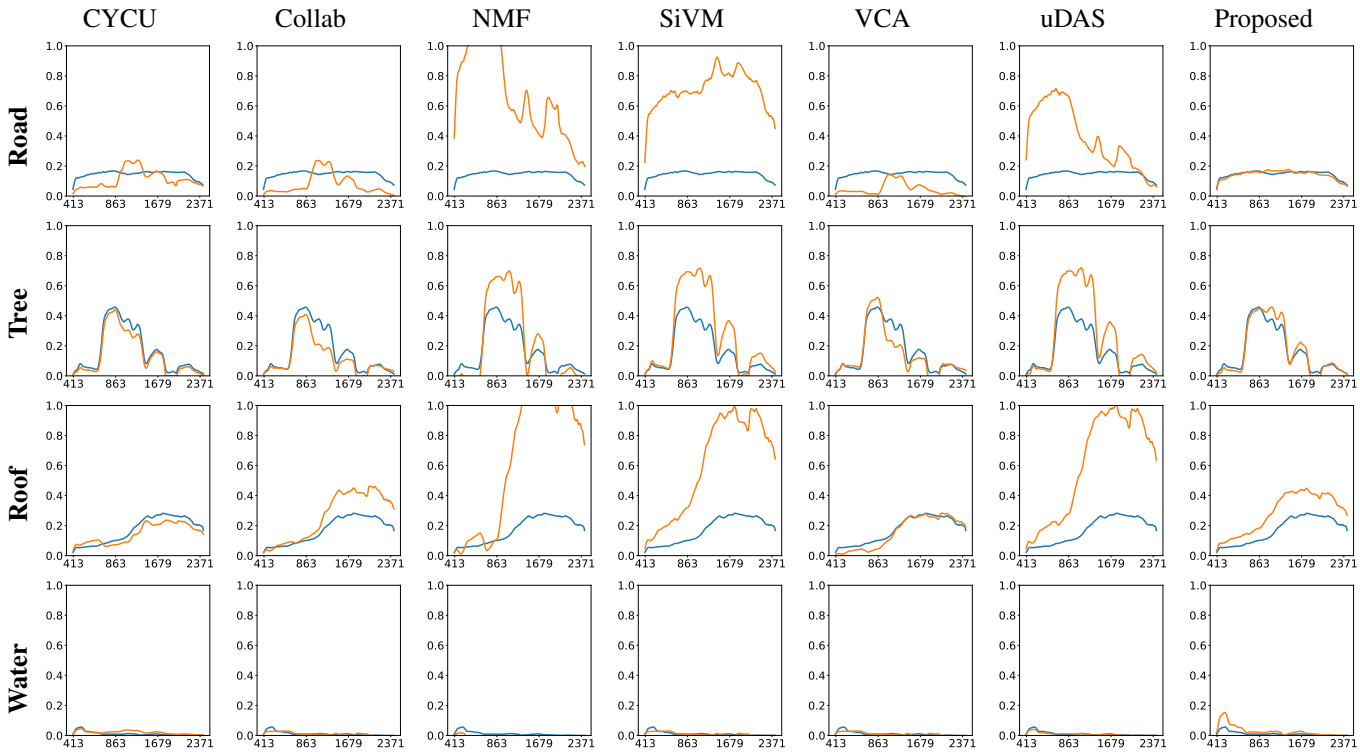


Fig. 9: Apex dataset - Visual comparison of the endmembers obtained by the different unmixing techniques. Blue: ground truth endmembers; Orange: estimated endmembers.

spectral signatures for the endmembers is more difficult than producing a good abundance map. The proposed model considerably outperforms all the other competing methods. On the Apex and WDC Mall datasets, the proposed model obtains SAD values of 0.0867 and 0.1537, respectively, about half of the next best method.

It is worth mentioning that a good SAD value does not necessarily guarantee good abundance maps, because SAD removes the norm of the endmember spectra. In other words, it ignores endmember scaling factors, caused by multiple reflections of the light and continuously variable illumination conditions in practical situations. However, such scaling fac-

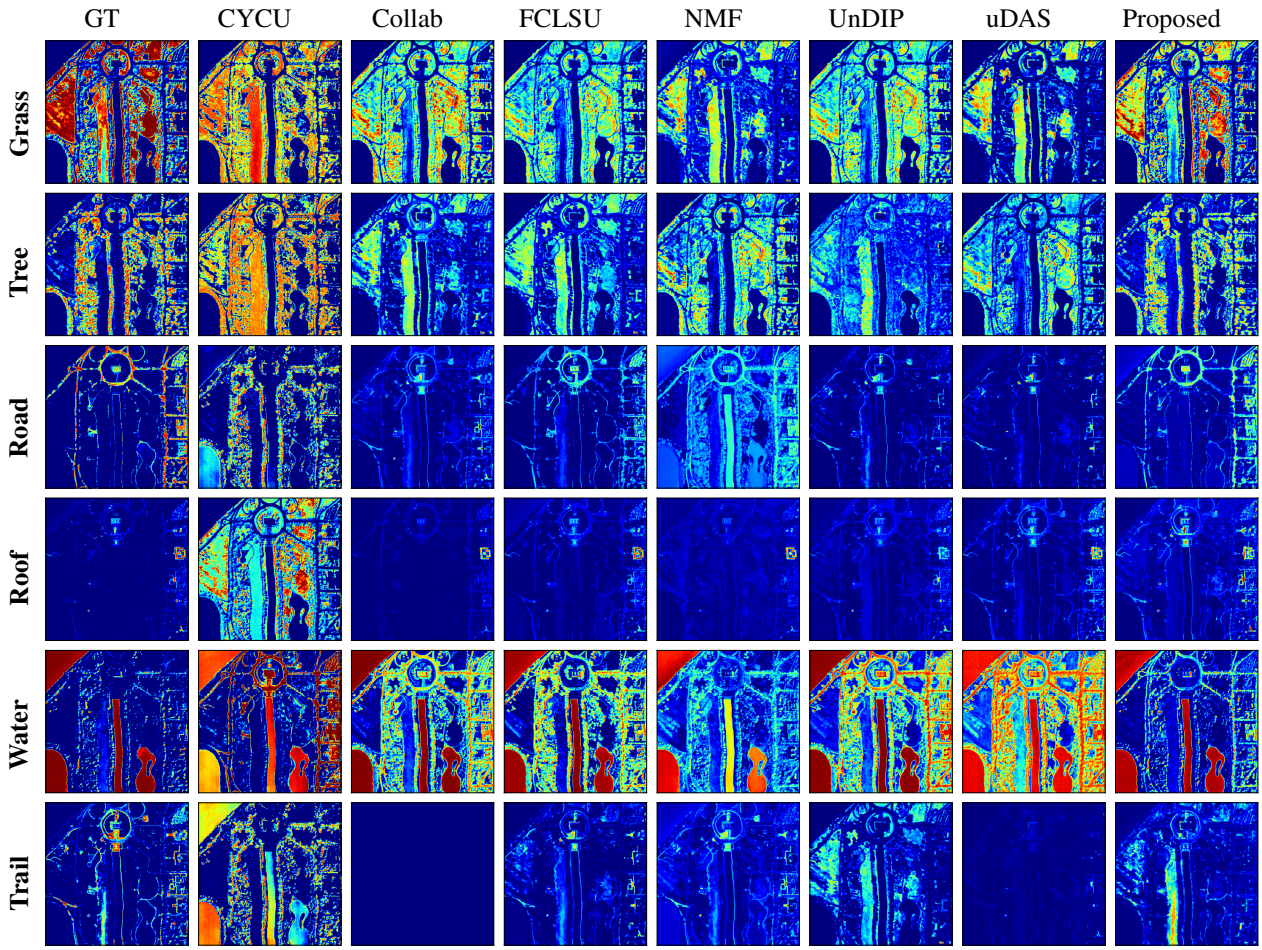


Fig. 10: Washington DC Mall dataset - Visual comparison of the abundance maps obtained by the different unmixing techniques.

tors can considerably affect the abundance estimation. As the proposed method provides the best results in both SAD and RMSE, one can conclude that it overcomes the mentioned problem at least to some degree.

F. Visual Analysis of Abundance Maps and Endmembers

The abundance maps and spectral signatures of the endmembers provide a way to visually compare the generated results of the different unmixing methods. Figs. 6, 8, and 10 show the abundance maps obtained from the various competing methods. It can be inferred that the abundance maps obtained from the proposed method are visually most similar to the ground truth abundance maps. The methods `UnDIP` and `uDAS` fail to properly represent the endmember "Water" across all the experimental datasets. Decent results are obtained by the methods `Collab`, `FCLSU`, and `NMF`, but their performance suffers from inconsistencies in RMSE values from one endmember to another. This causes the models to lose in terms of overall performance, even if they manage to obtain good results on a particular endmember. For example, none of the competing methods was able to correctly produce the "Road" endmember in the Apex dataset. The success of the proposed model on this endmember can be attributed to the ability of the transformer encoder block with self-patch attention to find the long distance feature dependencies, which are otherwise lacking in the abundance maps obtained from the output of the convolutional network.

Figs. 7, 9 and 11 depict the extracted endmembers. It was observed that the methods using `VCA` as initialization could not further improve the `VCA` extracted endmembers by much, leading to higher values of SAD later on. The proposed method however is also initialized by `VCA` analysis, but modifies the spectral signatures in a way that they more closely resemble the ground truth endmembers, with much lower SAD errors.

G. Sensitivity Analysis to Hyperparameters

The hyperparameters β and γ play essential roles in determining the model's overall performance. In order to keep the training process simple, the value of β was kept constant at 5×10^3 for all the datasets. Fig. 12 depicts the sensitivity of the proposed unmixing model to the hyperparameter γ. Both SAD and RMSE values are correlated, and changing γ affects both of them similarly in most cases. The figure suggests that γ can be set in the range 1×10^{-4} to 1×10^{-2}, with a higher number of endmembers favouring a lower γ value.

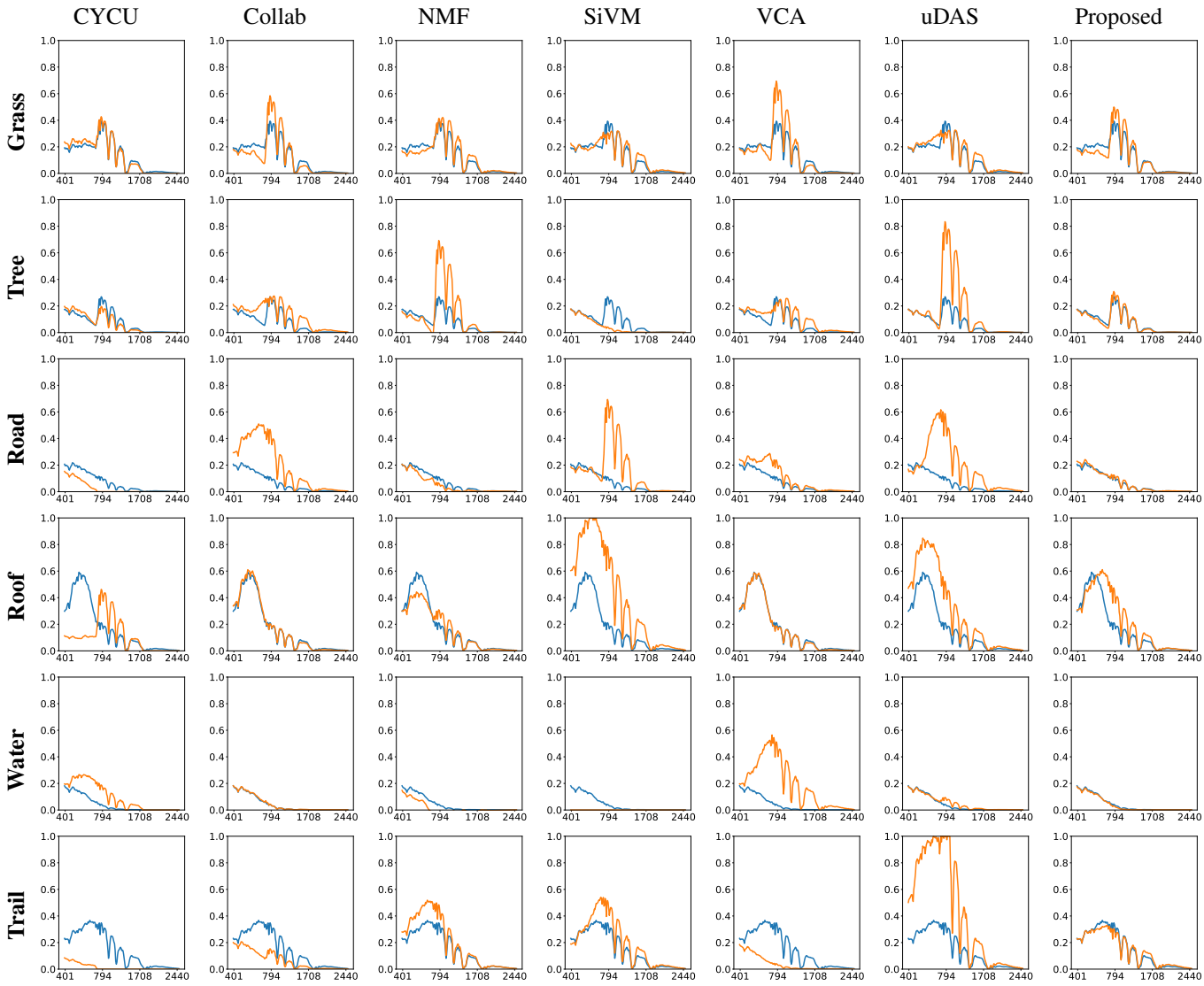


Fig. 11: Washington DC Mall dataset - Visual comparison of the endmembers obtained by the different unmixing techniques. Blue: ground truth endmembers; Orange: estimated endmembers.

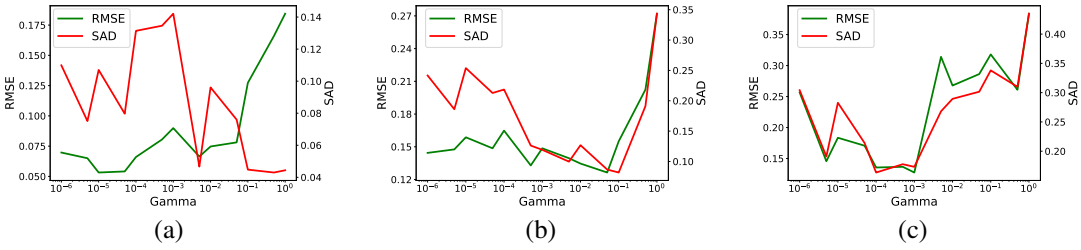


Fig. 12: Effect of the hyperparameter γ on **RMSE** and **SAD** error for (a) Samson dataset (b) Apex dataset and (c) Washington DC Mall dataset

Apart from the hyperparameters mentioned above, the learning rate and the weight decay were also found to have a significant impact on the obtained results, as can be seen in Fig. 13. Learning rates were tested in the range from 0.001 to 0.009, and the best results were obtained in the range from 0.006 to 0.009, with images having lower spatial dimensions preferring a slightly lower learning rate. The weight decay was tested in the range from 1×10^{-5} to 9×10^{-5}. Fig. 13 suggests an optimal value around 3×10^{-5}. It was observed that the quality of the abundance maps quickly deteriorates

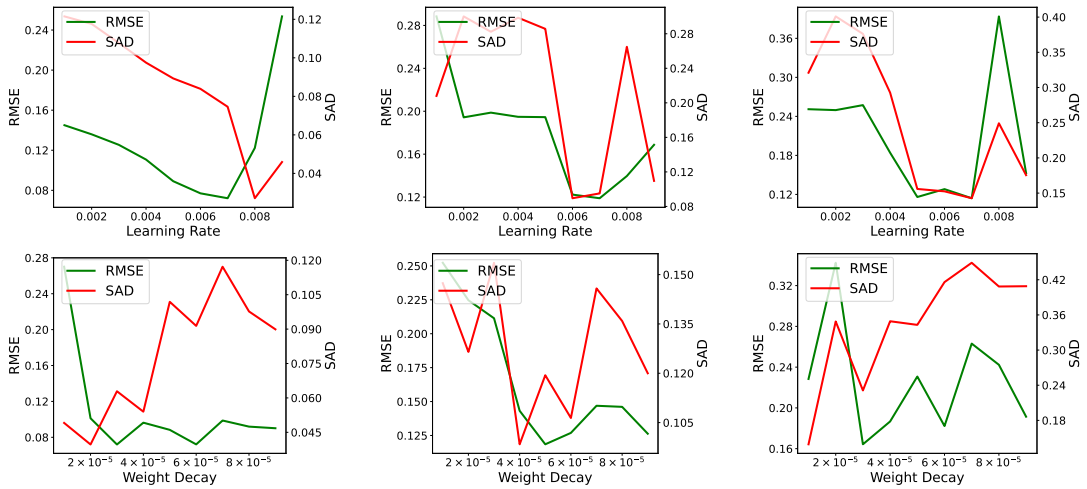


Fig. 13: Effect of learning rate and weight decay on **RMSE** and **SAD** values for (a) Samson dataset (b) Apex dataset and (c) Washington DC Mall dataset

with increasing weight decay.

The optimal parameters were selected using a grid search-based approach on the sample space [51], and the combination of parameter values which resulted in the minimal value of the loss function in Eq. (12) was finally applied to obtain the reported results.

IV. Conclusion

In this article we proposed a novel HSI unmixing model that uses a convolutional autoencoder combined with a transformer. We demonstrated the viability of the novel Multihead Self-Patch Attention mechanism used in the encoder block of the transformer. The experiments were carried out on three real datasets, each with its unique set of challenges, and were successfully handled by the proposed model with consistent performance across the range of endmembers. The accuracy and consistency of the proposed model can be credited to the use of the transformer block which captures the long range feature dependencies that are otherwise not reachable by a CNN based architecture. This enables our model to achieve superior unmixing results, which are significantly better than the competing methods.

Acknowledgment

The authors thank Ganesan Narayanasamy who is leading IBM OpenPOWER/POWER enablement and ecosystem worldwide for his support to get the IBM AC922 system's access. B. Koirala is funded by the Research Foundation-Flanders - Project G031921N.

References

[1] 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, 2020.

[2] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 60–88, 2020.

[3] S. K. Roy, R. Mondal, M. E. Paoletti, J. M. Haut, and A. Plaza, "Morphological convolutional neural networks for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8689–8702, 2021.

[4] B. Koetz, F. Morsdorf, S. Van der Linden, T. Curt, and B. Allgöwer, "Multi-source land cover classification for forest fire management based on imaging spectrometry and lidar data," *Forest Ecology and Management*, vol. 256, no. 3, pp. 263–271, 2008.

[5] 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.

[6] W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognition*, vol. 48, no. 12, pp. 3904–3916, 2015.

[7] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[8] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 37–78, 2017.

[9] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, April 2012.

[10] D. C. Heinz and Chein-I-Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529–545, 2001.

[11] R. Heylen, D. Burazerovic, and P. Scheunders, "Fully constrained least squares spectral unmixing by simplex projection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4112–4122, Nov 2011.

[12] J. Nascimento and J. Bioucas-Dias, "Vertex component analysis: A fast algorithm to extract endmembers spectra from hyperspectral data," in *Pattern Recognition and Image Analysis*, F. J. Perales, A. J. C. Campilho, N. P. de la Blanca, and A. Sanfeliu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 626–635.

[13] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis-

based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4418–4432, 2009.

[14] M. E. Winter, "N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Imaging Spectrometry V*, M. R. Descour and S. S. Shen, Eds., vol. 3753, International Society for Optics and Photonics. SPIE, 1999, pp. 266 – 275. [Online]. Available: https://doi.org/10.1117/12.366289

[15] W. Full, R. Ehrlich, and J. Klovan, "Extended qmodel—objective definition of external end members in the analysis of mixtures," *Journal of the International Association for Mathematical Geology*, vol. 13, pp. 331–344, 08 1981.

[16] M. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 3, pp. 542–552, 1994.

[17] J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, 2008, pp. III – 250–III – 253.

[18] J. M. Bioucas-Dias, "A variable splitting augmented lagrangian approach to linear spectral unmixing," in *2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2009, pp. 1–4.

[19] N. Dobigeon, S. Moussaoui, M. Coulon, J. Tourneret, and A. O. Hero, "Joint bayesian endmember extraction and linear unmixing for hyperspectral imagery," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4355–4368, 2009.

[20] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 3, pp. 765–777, 2007.

[21] J. Li, J. M. Bioucas-Dias, A. Plaza, and L. Liu, "Robust collaborative nonnegative matrix factorization for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6076–6090, 2016.

[22] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Collaborative nonnegative matrix factorization for remotely sensed hyperspectral unmixing," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2012, pp. 3078–3081.

[23] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. Huntington, "Ice: a statistical approach to identifying endmembers in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 10, pp. 2085–2095, 2004.

[24] L. Zhuang, C. Lin, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 9858–9877, 2019.

[25] M. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2014–2039, 2011.

[26] ——, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4484–4502, 2012.

[27] B. Rasti and B. Koirala, "Suncnn: Sparse unmixing using unsupervised convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2021.

[28] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[29] S. K. Roy, P. Kar, D. Hong, X. Wu, A. Plaza, and J. Chanussot, "Revisiting deep hyperspectral feature extraction networks via gradient centralized convolution," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[30] B. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "Blind hyperspectral unmixing using autoencoders: A critical comparison," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–1, 2022.

[31] Y. Su, A. Marinoni, J. Li, J. Plaza, and P. Gamba, "Stacked nonnegative sparse autoencoders for robust hyperspectral unmixing," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1427–1431, 2018.

[32] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "Daen: Deep autoencoder networks for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4309–4321, 2019.

[33] R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez, "Deep generative endmember modeling: An application to unsupervised spectral unmixing," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 374–384, 2020.

[34] Q. Jin, Y. Ma, F. Fan, J. Huang, X. Mei, and J. Ma, "Adversarial autoencoder network for hyperspectral unmixing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[35] M. Tang, Y. Qu, and H. Qi, "Hyperspectral nonlinear unmixing via generative adversarial network," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 2404–2407.

[36] S. K. Roy, J. M. Haut, M. E. Paoletti, S. R. Dubey, and A. Plaza, "Generative adversarial minority oversampling for spectral–spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[37] Y. Qu and H. Qi, "udas: An untied denoising autoencoder with sparsity for spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1698–1712, 2019.

[38] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao, "Hyperspectral unmixing via deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 11, pp. 1755–1759, 2018.

[39] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Convolutional autoencoder for spectral-spatial hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2020.

[40] L. Gao, Z. Han, D. Hong, B. Zhang, and J. Chanussot, "Cycu-net: Cycle-consistency unmixing network by learning cascaded autoencoders," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–14, 2021.

[41] F. Khajehrayeni and H. Ghassemian, "Hyperspectral unmixing using deep convolutional autoencoders in a supervised scenario," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 567–576, 2020.

[42] B. Rasti, B. Koirala, P. Scheunders, and P. Ghamisi, "UnDIP: Hyperspectral unmixing using deep image prior," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2021.

[43] B. Rasti, B. Koirala, P. Scheunders, and J. Chanussot, "Misicnet: Minimum simplex convolutional network for deep hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–1, 2022.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[46] J. Guo, K. Han, H. Wu, C. Xu, Y. Tang, C. Xu, and Y. Wang, "Cmt: Convolutional neural networks meet vision transformers," *arXiv preprint arXiv:2107.06263*, 2021.

[47] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 357–366.

[48] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5412–5427, 2014.

[49] M. E. Schaepman, M. Jehle, A. Hueni, P. D'Odorico, A. Damm, J. Weyermann, F. D. Schneider, V. Laurent, C. Popp, F. C. Seidel, K. Lenhard, P. Gege, C. Küchler, J. Brazile, P. Kohler, L. De Vos, K. Meuleman, R. Meynart, D. Schläpfer, M. Kneubühler, and K. I. Itten, "Advanced radiometry measurements and earth science applications with the airborne prism experiment (apex)," *Remote Sensing of Environment*, vol. 158, pp. 207–219, 2015.

[50] L. Drumetz, T. R. Meyer, J. Chanussot, A. L. Bertozzi, and C. Jutten, "Hyperspectral image unmixing with endmember bundles and group sparsity inducing mixed norms," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3435–3450, 2019.

[51] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.