Baltic J. Modern Computing, Vol. 13 (2025), No. 3, pp. 624–640
https://doi.org/10.22364/bjmc.2025.13.3.04

Hyperspectral Unmixing of Agricultural Images Taken from UAV Using Adapted U-Net Architecture

Vytautas PAURA, Virginijus MARCINKEVIČIUS

Institute of Data Science and Digital Technologies, Vilnius University, Vilnius, Lithuania

vytautas.paura@mif.stud.vu.lt, virginijus.marcinkevicius@mif.vu.lt

ORCID 0000-0002-4315-729X, ORCID 0000-0002-2281-4035

Abstract. The hyperspectral unmixing method is an algorithm that extracts material (usually referred to as endmembers) data from hyperspectral data, along with their corresponding abundances. Due to the lower spatial resolution of hyperspectral sensor data compared to conventional cameras, each pixel is more likely to contain mixed information from multiple endmembers. In turn, mixed hyperspectral data is less valuable for use in research or predictive models. One of the problems in hyperspectral unmixing is the lack of openly available, field-collected datasets, particularly those from agricultural and other UAV-gathered sources. In turn, hyperspectral unmixing algorithms are rarely tested on this data type. This paper proposes a hyperspectral unmixing algorithm based on the U-Net network architecture to achieve more accurate unmixing results on existing and newly created hyperspectral unmixing datasets. The proposed model is fully unsupervised and is not limited by data shape and size. We also developed and shared a hyperspectral unmixing dataset derived from blueberry field data collected using a hyperspectral camera mounted on a Unmanned Aerial Vehicle (UAV). Compared to the state-of-the-art transformer-based unmixing model, our proposed algorithm achieved approximately 20% lower endmember RMSE and more than 50% lower reconstruction error values.

Keywords: Hyperspectral Unmixing, Remote Sensing, Deep Neural Networks, Hyperspectral Dataset.

1 Introduction

The growing popularity of remote sensing systems and advancements in hyperspectral imaging technologies have increased interest in using these technologies in various agricultural applications. Most commonly used near-infrared hyperspectral cameras enable the simultaneous collection of large amounts of spatial and spectral information. A large amount of spectral data comes at the cost of spatial resolution compared to RGB or multispectral cameras. Smaller spatial resolution results in mixed material data within each pixel of hyperspectral data, particularly when using satellite hyperspectral sensors. To address the issue of mixed data, hyperspectral unmixing algorithms are employed, which typically perform three tasks simultaneously: identifying material (or endmember) counts, determining material spectral signatures, and determining mixture amounts (or abundances) in each hyperspectral pixel. In this paper, we focus on expanding hyperspectral unmixing research using agricultural hyperspectral data and compare our proposed algorithm with a transformer-based algorithm. This paper is structured into several chapters, including a literature review on hyperspectral unmixing algorithms and their applications to agricultural data.

2 Related Work

Works related to hyperspectral unmixing use cases focusing on agricultural data and algorithms used for hyperspectral unmixing that are based on sparse regression, non-negative matrix factorisation, and neural networks.

2.1 Hyperspectral Unmixing Algorithms

A few of the most common types of hyperspectral unmixing algorithms were reviewed in our previous work (Paura and Marcinkevičius, 2023). These three types of algorithms are:

- Semi-supervised sparse regression modelling.
- Unsupervised non-negative matrix factorisation methods.
- Unsupervised deep learning autoencoder neural networks.

Sparse regression. Sparse regression algorithms are used because, in a hyperspectral image, most pixels contain only a few material data values mixed in, compared to the total number of endmembers in the image, creating a sparse abundance matrix. Popular algorithms in this category are:

The sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) and *constrained SUnSAL* (C-SUnSAL) algorithms (Bioucas-Dias and Figueiredo, 2010) are based on the Alternating Direction Method of Multipliers (ADMM) (Gabay and Mercier, 1976). The ADMM algorithm breaks down a complex problem into a series of more straightforward issues. The results provided by the authors are in dB values of reconstruction signal-to-noise ratio (RSNR) metric, and both algorithms were tested using 50 dB of artificial noise.

SUnSAL and total variation (SUnSAL-TV) (Iordache and Bioucas-Dias, 2012) is a variation of the SUnSAL algorithm with an added total variation regularisation, which provides spatial information for better spectral unmixing results. The created TV regularizer accounts for spatial homogeneity because it is very likely that neighbouring pixels will have quite a similar abundance fraction of the same endmembers.

Spectral–spatial weighted sparse unmixing (Zhang and Li, 2018) is a hyperspectral unmixing framework that computes a sparse solution constrained by both spectral and spatial domains simultaneously. It implements ADMM for parameter and coefficient optimisation purposes.

Non-negative matrix factorisation. Non-negative matrix factorisation algorithms are used because the information gathered by the hyperspectral sensors can never be harmful, and the hyperspectral cube can be factored into abundance, endmember and residual noise matrices. Popular algorithms in this category are:

Coupled Non-negative Matrix Factorisation (CNMF) (Yokoya and Yairi, 2012) is an algorithm that simultaneously unmixes high spatial resolution multispectral data and high spectral resolution hyperspectral data, thereby achieving fusion of hyperspectral and multispectral data. The algorithms utilise a vertex component analysis (VCA) algorithm to calculate the initial endmember matrix from the spectral data and a user-defined number of endmembers to identify.

Spatial group sparsity regularised NMF (SGSNMF) (Wang and Zhong, 2017) is a blind unmixing method that incorporates a spatial group sparsity regularizer constraint, which takes into account the pixel location (spatial data) and the fact that abundance matrices are sparse.

Neural networks. The last type of algorithm is neural networks, specifically autoencoder-type networks, that create an artificial neuron bottleneck to compress the data into a latent space, extracting spatial and spectral features from hyperspectral images. Deep neural network algorithms are gaining popularity due to their ability to learn from complex, non-linear data relations. Due to the base model architecture design, these types of neural networks can be trained using the difference between the original image and its reconstruction, making the algorithm unsupervised. Popular algorithms and packages in this category are:

HySUPP - An open source *Python* package created by (Rasti and Zouaoui, 2023) for hyperspectral unmixing as a framework to run experiments on different hyperspectral data and the implemented unmixing algorithms. The framework was not included in the experiments conducted in this paper because the algorithms implemented by the framework are supervised or semi-supervised.

The Deep Generative Unmixing algorithm (DeepGUn) (Borsoi and Imbiriba, 2020) is a spectral unmixing algorithm based on Generative models, such as generative adversarial networks (GANs) and variational autoencoders (VAEs). According to the authors, their proposed strategy yields more accurate abundance estimation at a minimal cost in computational resources. Their proposed autoencoder architecture consists of three hidden encoder layers with rectified linear unit (ReLU) activation functions, three hidden decoder layers with ReLU activation functions, and an input and output layer with sigmoid activation functions.

The Deep Half-Siamese network (Deep HSNet) (Dong and Yuan, 2020) is a hyperspectral unmixing algorithm that consists of two distinct networks: an endmember-guided network and a reconstruction network. The first network maps extracted endmembers to the abundances, while the reconstruction network is an autoencoder architecture that recreates hyperspectral pixels.

2.2 Agricultural Hyperspectral Data Unmixing

A comprehensive and extensive review paper on the usage and unmixing of hyperspectral data in agriculture was written by (Guerri and Distante, 2024). The authors explore the use cases of hyperspectral data in agriculture and the available algorithms used to unmix this hyperspectral data. They examine these types of algorithms in their papers: Autoencoder denoising (Deng and Zhou, 2023), Convolutional Neural Networks (Mou and Ghamisi, 2018), Recurrent Neural Networks for classification (Zhou and Hang, 2019), Deep Belief Networks (Li and Huang, 2022), Generative Adversarial Networks for denoising (Zhang and Cai, 2022) and super-resolution (Shi and Han, 2022), Transfer Learning for classification tasks (Li and Liu, 2022), Semi-Supervised Learning classification (Zheng and Jia, 2022), Unsupervised learning classification (Mou and Ghamisi, 2018). From their paper, it can be concluded that the most popular unmixing methods recently are all within the domain of deep learning algorithms.

A paper by (Annam and Singla, 2021) employs supervised and unsupervised machine learning methods to detect heavy metals (arsenic (As), cadmium (Cd), and lead (Pb)) in soil using hyperspectral data, with the unsupervised k-means algorithm achieving the best resulting accuracy of approximately 98%.

3 Hyperspectral Dataset of Blueberry Fields

In this section, we describe the field data gathering process and the creation of a hyperspectral unmixing dataset from hyperspectral data gathered by a UAV flying over a blueberry field. Building on the review paper by (Guerri and Distante, 2024) and our previous work (Paura and Marcinkevičius, 2023), it was concluded that the amount of openly available hyperspectral data, especially in agricultural areas, is limited. In this paper, we create a hyperspectral unmixing dataset from UAV data gathered in blueberry fields. Blueberry field dataset is an expansion on our previous work (Paura and Marcinkevičius, 2024) with these key differences:

- Further research on classification accuracy using the Vertex Component Analysis (VCA) (Nascimento and Dias, 2005) algorithm, changing the variation threshold in class data sampling to 1.5 σ from 2. This, in turn, extends the variability of the data, preserving more of the original data and keeping the dataset as close to the raw data as possible.
- Expansion of the dataset from a single hyperspectral cube to 3 cubes, for more extensive experimentation and data variety. This improves the usability of the dataset for machine learning algorithm training, validation, and testing, as each of the cubes can be used for separate tasks.
- Experimentation on classification, with the best results achieved by keeping the same class distribution in all hyperspectral data cubes. Keeping the same number of classes across all three data cubes allows for simpler training, testing and validation of machine learning models that have number of classes as a hyperparameter.

3.1 Data gathering

Raw hyperspectral data was gathered using an *Aurelia* X6 drone (Aurelia Technologies Inc, 2024) with *SPECIM* hyperspectral push broom camera (Specim, 2024) flying over a blueberry field. Push broom hyperspectral camera records the data of all (in case of this data gathering mission) 224 spectral bands in lines 1024 pixels wide. In turn, the final hyperspectral cube size gathered depends on the flight length, but will always have a similar size of 1024 pixels wide with 224 spectral bands gathered. A data-gathering flight was conducted at 70 meters above ground, and the required drone speed was calculated to maintain a square pixel resolution. The final pixel size of gathered UAV hyperspectral cubes is 5 x 5 centimetres.

To maintain the accuracy of the data, recording must be done along straight flight paths, known as flight lines. Each flight line creates a separate hyperspectral data cube of size $1024 * x * 224$, where x depends on the line length and camera recording speed. An exposure time of 6 ms and the camera fps (or lines per second in push broom camera case) set to 100.

3.2 Calibration

To keep the data consistent and comparable between flights, a set of calibration carpets was deployed in the field with their laboratory-calibrated reflectance values of 5 %, 10 %, and 40 %, and a data cube with the camera lens closed to gather fully dark data or sensor noise. Calibration was performed using the reference/reflectance carpets placed in the field, with one of the drone flight lines intersecting the carpets. The primary methodology used was that described in the article by (Burger and Geladi, 2005).

3.3 Raw Hyperspectral Data

From the multiple hyperspectral data cubes gathered during the UAV mission, three data cubes were selected as the base of the unmixing dataset. Three cubes were chosen to create train, test, and validation data cubes. All data cubes share the same set of end-members (e.g., blueberries, grass, soil, water and wet soil, areas obscured by shadows, and any other data), but the data were collected over the field at different times and in various locations within the same blueberry field. Three cubes are used to increase data variety and, in turn, check algorithm robustness to changes in field data. The three data cubes have these parameters:

- Cube 1 shape: 1024 pixels wide, 3177 pixels long, with 224 spectral bands.
- Cube 2 shape: 1024 pixels wide, 3047 pixels long, with 224 spectral bands.
- Cube 3 shape: 1024 pixels wide, 2815 pixels long, with 224 spectral bands.
- All cubes have the same spectral data collected from 400 to 1000 nm with an average distance between bands of approximately 2.5 nm.

Hyperspectral cube RGB representations, created by data integration over the International Commission on Illumination (CIE) 1931 XYZ colour matching functions and conversion from XYZ to RGB, are given below in Figure 1.

3.4 Hyperspectral dataset ground truth creation

Gathered field hyperspectral data has the disadvantage of not having completely accurate classification and ground truth data, which are challenging to create. To apply

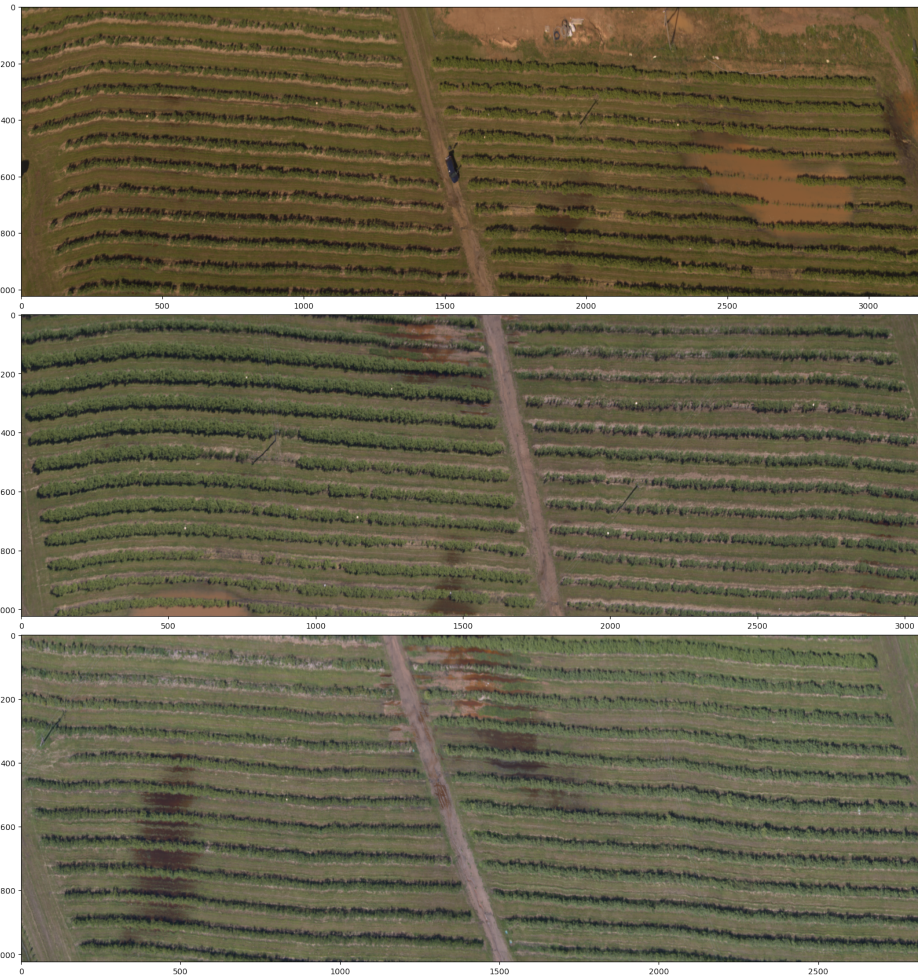

Fig. 1: RGB representation of the hyperspectral blueberry data cubes, from top to bottom, Cube 1, Cube 2 and Cube 3 are shown. The three selected hyperspectral cubes were collected in the same blueberry field at different locations along the field and over several days to increase data variety.

as accurate a data classification as possible, a collection of classes was extracted from the data cubes using an unsupervised method called VCA (Nascimento and Dias, 2005). The suggested endmembers extracted using the VCA algorithm were used as the ground truth classes for this hyperspectral unmixing dataset. By selecting multiple endmember counts, the algorithm extracted possible endmembers from the hyperspectral data cube (Cube 1 was used for extraction). A higher number of classes resulted in a smaller sample size for each class and an increase in the difficulty of unmixing down the line. The VCA selected six classes to maintain a high-class representation of the hyperspectral image and a reasonable calculation difficulty. Raw class data is represented in Figure 2. Classes represent blueberry crops, bare soil, grass, data in shadow, water, and other data types that are distinct from different classes.

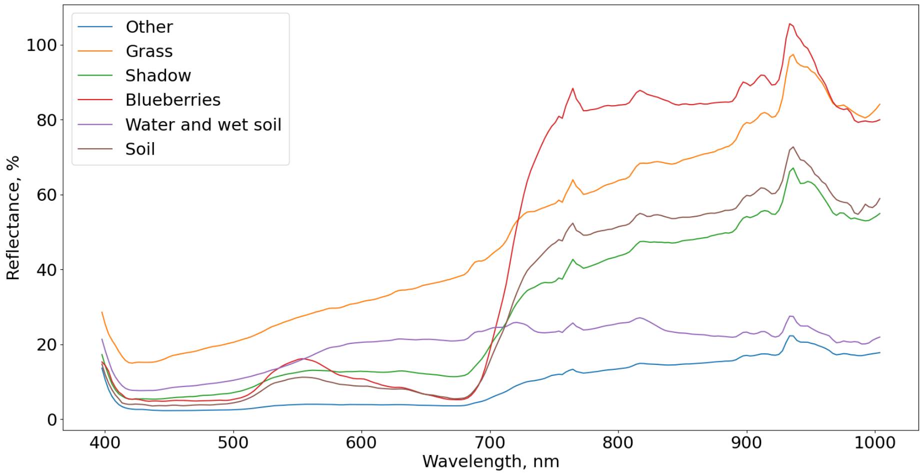

Fig. 2: Averages, for each of the six classes, of extracted endmembers used as the ground truth for the created hyperspectral unmixing dataset.

The whole data cube classification was performed using the extracted possible endmembers. Each pixel was checked against the endmembers to classify the data, and RMSE values (Equation 1) were calculated for each one. The class selected for each pixel had the lowest RMSE value among the six extracted endmembers. To maintain a variation of data in the hyperspectral data cube, each pixel was left as the original data if its variation was within 1.5 σ of its respective class. In other words, the pixels close to the extracted endmember were left unchanged and only given a class number. Pixels outside of a 1.5 σ variation were replaced with random pixel data from within the class data distribution, to keep computation times shorter and maintain higher spatial variation.

Other endmember extraction and cube classification methods may be used to balance the dataset's resource requirements, labelling time, and data variation. Complete raw and classified data are published as open data for other experiments and hyperspectral unmixing and classification tasks. Data is published on the Zenodo platform with open access: `https://doi.org/10.5281/zenodo.13856357`.

Class distribution on the hyperspectral data Cube 1 is shown in Figure 3.

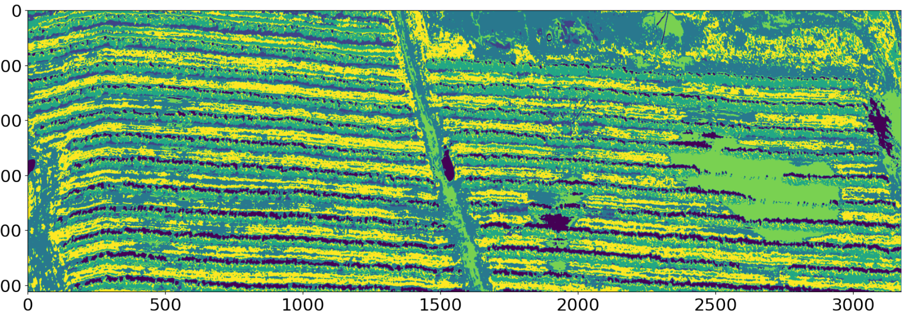

Fig. 3: Class distribution in the hyperspectral Cube 1. Color class representation: Yellow - bare soil; Green - Blueberries; Blue - Grass; Dark blue - Shadows; Light green - Water and wet soil; Black - Other data

3.5 Ground truth data mixing for hyperspectral unmixing dataset creation

Classified hyperspectral data cubes were mixed using a sliding window and linear mixing. Multiple sliding window sizes were considered, and experiments were conducted on window kernel sizes of 2, 3, and 4. The primary kernel size used was 3, to balance the amount of data mixed (9 pixels with this kernel mixed into 1) and the cube sizes. As this is a continuation of our previous work (Paura and Marcinkevičius, 2023), a smaller dataset was targeted due to the numerous hyperspectral unmixing algorithm resources and computational requirements associated with large datasets.

With the selected sliding windows of size 3x3, 9 pixels were linearly mixed into 1 pixel, including the classes of those pixels. Created the dataset cubes of sizes:

- Mixed cube 1 shape: 341 pixels wide, 1059 pixels long, with 224 spectral bands.
- Mixed cube 2 shape: 341 pixels wide, 1015 pixels long, with 224 spectral bands.
- Mixed cube 3 shape: 341 pixels wide, 938 pixels long, with 224 spectral bands.

In previous work (Paura and Marcinkevičius, 2024), six classes were used to classify the hyperspectral cubes. Using the same sliding window over the class array, an abundance matrix was generated of the same size as the hyperspectral data, but with a third dimension of size 6.

An example RGB representation of data Cube 1 is given in Figure 4. The RGB representation is computed using data scaled across the entire hyperspectral cube; differences in minimum and maximum values between images distort the final colours. The generated RGB image is a false-colour image used to visualise the hyperspectral data cube conveniently.

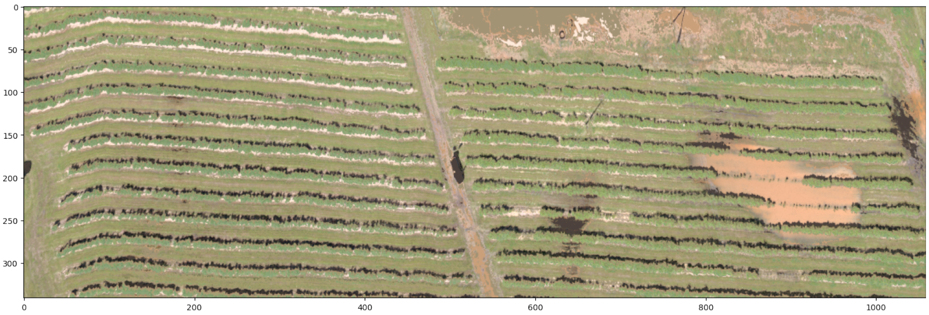

Fig. 4: Mixed hyperspectral cube RGB representation that was used in training.

4 Hyperspectral Unmixing Using U-Net-based Architecture

This section describes our proposed hyperspectral unmixing algorithm based on the U-Net model architecture and the datasets used in experimentation and validation. With the growing popularity of deep learning neural networks and based on existing works in hyperspectral unmixing of agricultural data, an unsupervised deep learning model is a viable solution already employed in hyperspectral unmixing.

4.1 Metrics

Multiple metrics are used in the hyperspectral unmixing problems. The most common are Root Mean Squared Error (RMSE), Signal Reconstruction Error (SRE), Spectral Angle Distance (SAD), Spectral Angle Mapping (SAM), and Reconstruction Error (RE). RMSE and Signal Reconstruction Error (SRE) metrics were selected due to their popularity in hyperspectral unmixing algorithm performance evaluation and their overall simplicity in describing the differences between evaluated and real spectra:

- RMSE (Equation 1) shows the difference between the predicted spectra and the ground truth. Different authors employed several variations of RMSE to evaluate various aspects of the developed algorithms, including the average RMSE between all endmembers, reconstruction RMSE, and abundance RMSE. Average RMSE is calculated by taking the computed RMSE for each endmember separately and calculating the average value. Reconstruction RMSE is computed between the ground truth signal and the signal reconstructed by a predictive model. Abundance RMSE is computed between ground truth abundances and predicted ones. This metric was selected due to its widespread use in papers on hyperspectral unmixing algorithms and its versatility in various error computations.
- SRE (Equation 2) is used to determine the quality of the spectral mixture reconstruction generated by the algorithms. A higher SRE value means a better reconstruction quality. This metric was selected as a modification of the RE metric and used explicitly for signal data.
- SAD (Equation 3) - measures the angles between two vectors in multidimensional space.

- Cosine similarity (Equation 5) - calculates the dot product of the vectors divided by the product of their lengths.
- RE (Equation 4) - measures the average difference between model-generated data and actual ground truth data.

Metrics are calculated using these formulas:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{1}$$

, where N is the number of values in the tested vector, x_i is the i-th actual value, and $\hat{x}_i$ is the i-th predicted value.

$$SRE = 10log_{10}\left(\frac{E[\|x\|_2^2]}{E[\|x-\hat{x}\|_2^2]}\right), \tag{2}$$

where x is the true value, $\hat{x}$ is the predicted value, and E is the average value of the expression inside.

$$L_{SAD}(I, \widehat{I}) = \frac{1}{R}\sum_{i=1}^{R}\arccos\left(\frac{\left\langle I_i, \widehat{I_i}\right\rangle}{\|I_i\|_2\|\widehat{I_i}\|_2}\right), \tag{3}$$

where R is the number of pixels, I_i - actual data, $\widehat{I_i}$ - predicted data. When applied to hyperspectral data, the two spatial dimensions of the hyperspectral data cube are combined into a single dimension of size R, and the angle is calculated for each pixel independently.

$$L_{RE}(I, \widehat{I}) = \frac{1}{W \cdot H}\sum_{i=1}^{H}\sum_{j=1}^{W}(\widehat{I}_{ij} - I_{ij})^2, \tag{4}$$

where W is the image width, H image height, $\widehat{I}_{ij}$ - predicted spectral data in pixel ij, I_{ij} - actual spectral data in pixel ij. The computation is performed for each pixel in the image separately.

$$\cos(\theta) = \frac{Y \cdot \widehat{Y}}{\|Y\| \left\|\widehat{Y}\right\|}, \tag{5}$$

where Y and $\widehat{Y}$ are the two input vectors to be measured, in the case of hyperspectral data, the vectors are the spectral values of each pixel.

4.2 Proposed model architecture

The original U-Net model created by (Ronneberger and Fischer, 2015) was used for biomedical image segmentation. Our previous work demonstrates that autoencoder networks are a widely used method in deep learning for hyperspectral unmixing. Due to this, a base autoencoder architecture from the U-Net model was employed. The autoencoder compresses data into a small latent space, allowing it to extract features at various scales during the compression process.

To adapt the U-Net model architecture for the hyperspectral unmixing task, a set of changes was made:

- Splitting the hyperspectral image into smaller, same-size photos to reduce the overall size of the model, enabling usage of augmentations (e.g., mirroring and rotations) on the input data and training the model by selecting these image patches in random order.
- Addition of cosine similarity loss 5 was used to encourage the model to extract less similar endmembers.
- Splitting the compressed data into endmember and abundance extraction sub-networks.
- Fully unsupervised unmixing model is trained on the hyperspectral image reconstruction loss.
- Ability to provide reference endmembers for more accurate unmixing or faster convergence speeds.

Compared to our previous work, these model changes were made:

- Addition of dropout layers after convolutional layers and the optimisation of dropout parameters.
- Addition of data augmentations during training.
- Optimisation of starting hyperparameter set, which increased stability across different datasets. The main changes were a variable learning rate.
- General code optimisation to reduce data reading overhead since large datasets do not fit in the graphics card memory during training.
- Improved and expanded testing methodology of the model and extended research on new and improved hyperspectral datasets.

Figure 5 shows the proposed model architecture. With the selected batch size and patch size, the model is constructed based on the feature extraction encoder layers, split into two parts. The first part extracts the endmembers by compressing the spatial data, and the second part extracts abundances from the spectral data. The data cube is reconstructed in the last layer to maintain the model's unsupervised nature. The reconstruction utilises matrix multiplication, rather than decoder layers, to learn the abundances and endmembers. The input and output data have the same shape, and the model is trained on the reconstruction accuracy.

The complete algorithm and experimentation code are open and provided in the repository at `https://github.com/VytautasPau/UAVHyperspectral`.

4.3 Datasets used for experimentation

In this section, we analyse freely available hyperspectral datasets selected for algorithm performance experimentation and our newly created hyperspectral unmixing dataset.

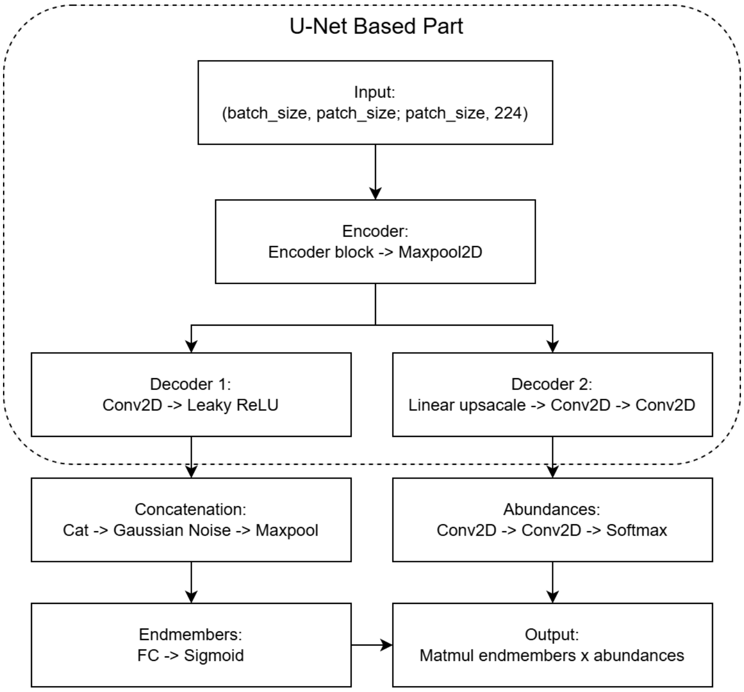
U-Net Based Part
Input:
(batch_size, patch_size; patch_size, 224)
Encoder:
Encoder block -> Maxpool2D
Decoder 1:
Conv2D -> Leaky ReLU
Decoder 2:
Linear upsacale -> Conv2D -> Conv2D
Concatenation:
Cat -> Gaussian Noise -> Maxpool
Abundances:
Conv2D -> Conv2D -> Softmax
Endmembers:
FC -> Sigmoid
Output:
Matmul endmembers x abundances

Fig. 5: The model architecture outline.

- *DC Mall* (Nanjing University, 2024) (MultiSpec, 2020). An area in Washington, DC, with a size of 1208 x 307 pixels and 191 spectral bands. Created ground truth for classification with these classes: Roofs, Streets, Paths, Grass, Trees, Water, and Shadows.
- *Samson* (Nanjing University, 2024) Hyperspectral data cube cut to the size of 95 * 95 pixels with 156 spectral bands and three different classes.
- *Apex* (Nanjing University, 2024) Hyperspectral data of 110 * 110 pixels with 285 spectral bands and four different classes.
- Cube 1 shape: 1024 pixels wide, 3177 pixels long, with 224 spectral bands.
- Cube 2 shape: 1024 pixels wide, 3047 pixels long, with 224 spectral bands.
- Cube 3 shape: 1024 pixels wide, 2815 pixels long, with 224 spectral bands.

The three open datasets selected were from the original transformer-based hyperspectral unmixing model created by (Ghosh and Roy, 2022). Their code included tuned hyperparameters for these datasets, along with evidence that the model performs well on them.

4.4 Evaluation of proposed method

Model performance was measured using this methodology:

- For each dataset, the model was trained until the change in reconstruction loss was almost zero; in turn, the number of training epochs and training time differed for each dataset. Not provided in Table 1, training times mainly depended on the input data size.
- For each dataset, the hyperparameter set was manually based on existing sets provided by the authors of the transformer-based hyperspectral unmixing model or determined after a few experimental runs. This was done to ensure that the algorithms produced a proper result, as not all hyperparameter sets yield a usable result.
- For model training, RE, SAD and an additional cosine similarity loss were used. Cosine similarity loss was added to encourage the model to learn less similar endmembers. This loss should not be used if the dataset is suspected to have similar classes.
- For testing the results, RMSE and SAD metrics were averaged over all classes. In addition, the hyperspectral cube RE is also provided.
- For each model and dataset combination, the models were entirely trained from scratch with weights initialised randomly on model creation.

Experimentation results for each dataset are provided in Table 1. Our proposed model results were compared to the transformer-based hyperspectral unmixing model created by (Ghosh and Roy, 2022). This model was selected due to the availability of code and novelty in hyperspectral unmixing algorithms using transformers, which showed higher accuracy than other deep neural network model papers available at the time of research initiation.

Table 1: Proposed method and transformer networks comparison results on selected datasets and metrics.

Dataset	Proposed model				Transformer model			
	mRMSE	mSAD	RE	Epochs	mRMSE	mSAD	RE	Epochs
Apex	**0.4705**	**0.1737**	**0.0990**	*1001*	0.5555	0.2025	0.1048	*1000*
DC	0.3971	0.3764	0.0480	*1001*	**0.3918**	**0.3009**	**0.0232**	*1000*
Samson	**0.4301**	**0.1507**	**0.0526**	*1001*	0.6031	0.2400	0.1675	*1000*
Blueberry Cube 1	**0.3112**	**0.2737**	**0.0752**	*3001*	0.4845	0.3951	0.3012	*1000*
Blueberry Cube 2	**0.3740**	**0.2591**	**0.1263**	*3001*	0.4511	0.4012	0.2860	*1000*
Blueberry Cube 3	**0.3088**	**0.2214**	**0.0978**	*3001*	0.4232	0.3852	0.2645	*1000*

Figure 6 shows the loss changes for both models to show the differences in training loss during one of the experiment runs.

The proposed model, created and trained, was used to extract the endmember data from Hyperspectral Data Cube 3. The resulting endmembers, compared to the ground truths, are given in Figure 7.

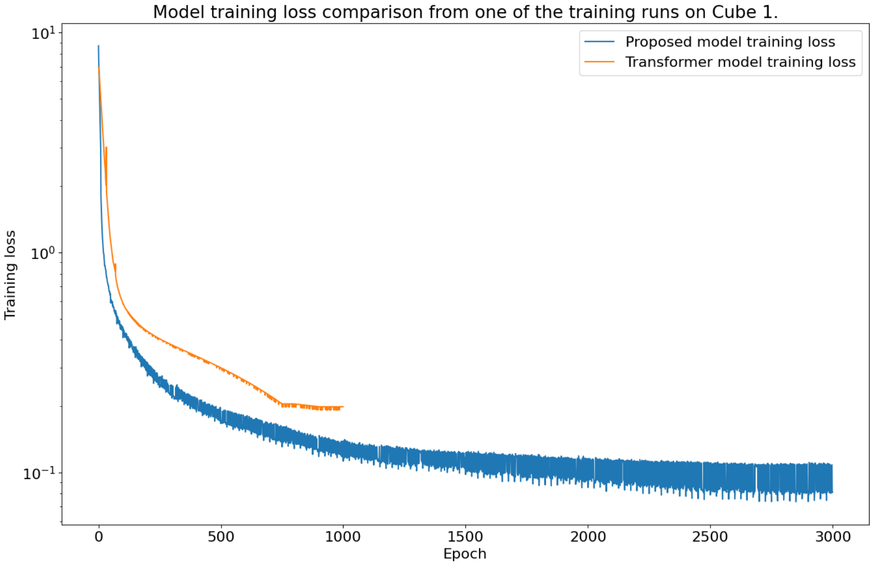

Fig. 6: The model training loss comparison on a Cube 1 experiment run.

5 Conclusions

- Mean RMSE values achieved with our proposed model were lower by about 27% than those of the comparison model on almost all datasets except the DC Mall dataset.
- RE and mean SAD values achieved by our proposed model were lower than those of the comparison model by about 187% and 41%, respectively, on average.
- Our proposed model architecture reduces the number of required hyperparameters by removing the need for latent space dimensions and patch size parameters used by transformer-based networks.

As explained previously, the given results are only the best achieved with manual parameter tuning and, consequently, may not be the optimal results for each of the datasets tested. For future works, an automated hyperparameter search is planned to minimise reconstruction error as much as possible.

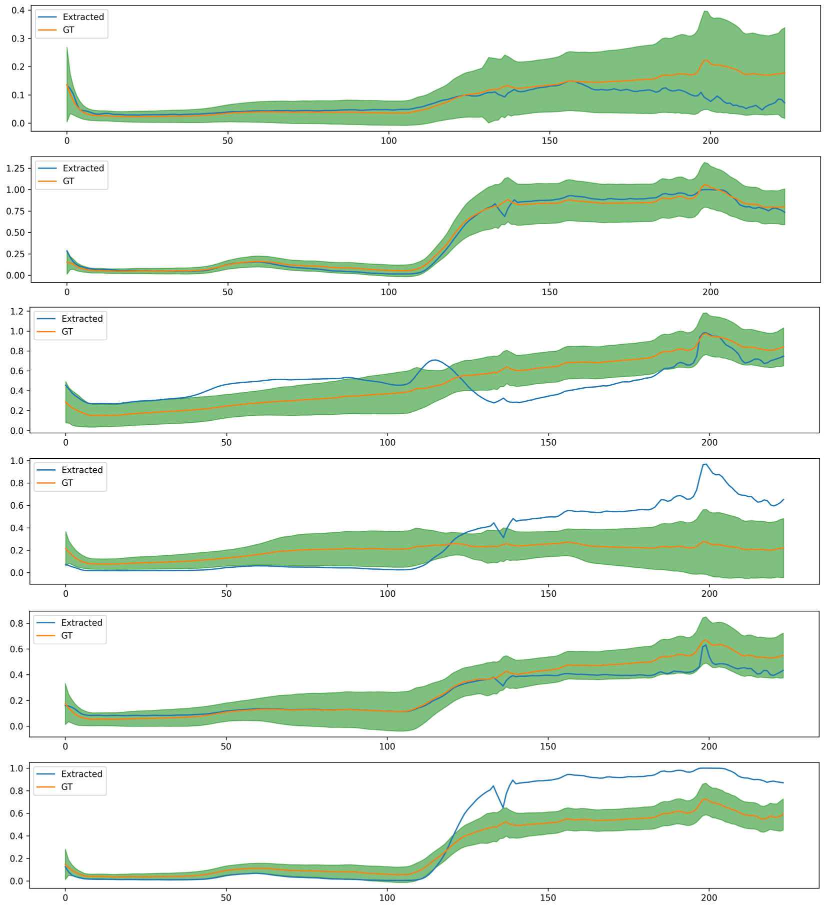

Fig. 7: Model predicted endmember (blue) comparison to ground truth endmember averages (orange) and their variations (green) for each of the six classes. The endmembers (from top to bottom) represent: 1) Other data, 2) blueberries, 3) water and wet soil, 4) soil, 5) shadows, 6) grass.

References

Annam, S., Singla, A. (2021). Spectral unmixing of heavy metal content in agricultural soil using hyperspectral remote sensing data, *2021 Sixth International Conference on Image Information Processing (ICIIP)*, Vol. 6, pp. 433–438.

Aurelia Technologies Inc, . (2024). Aurelia x6 standard, `https://aurelia-aerospace.com/product/aurelia-x6-standard/`.

Bioucas-Dias, J. M., Figueiredo, M. A. T. (2010). Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing, *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pp. 1–4.

Borsoi, R. A., Imbiriba, T. (2020). Deep generative endmember modeling: An application to unsupervised spectral unmixing, *IEEE Transactions on Computational Imaging* **6**, 374–384.

Burger, J., Geladi, P. (2005). Hyperspectral nir image regression part i: Calibration and correction, *Journal of Chemometrics* **19**, 355 – 363.

Deng, L., Zhou, B. (2023). A noise estimation method for hyperspectral image based on stacked autoencoder, *IEEE Access* **11**, 89835–89843.

Dong, L., Yuan, Y. (2020). Spectral-spatial joint sparse nmf for hyperspectral unmixing, *IEEE Transactions on Geoscience and Remote Sensing* pp. 1–12.

Gabay, D., Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation, *Computers & Mathematics with Applications* **2**(1), 17–40.
`https://www.sciencedirect.com/science/article/pii/0898122176900031`

Ghosh, P., Roy, S. K. (2022). Hyperspectral unmixing using transformer network, *IEEE Transactions on Geoscience and Remote Sensing* **60**, 1–16.

Guerri, M. F., Distante, C. (2024). Deep learning techniques for hyperspectral image analysis in agriculture: A review, *ISPRS Open Journal of Photogrammetry and Remote Sensing* **12**, 100062.
`https://www.sciencedirect.com/science/article/pii/S266739322400005X`

Iordache, M., Bioucas-Dias, J. M. (2012). Total variation spatial regularization for sparse hyperspectral unmixing, *IEEE Transactions on Geoscience and Remote Sensing* **50**(11), 4484–4502.

Li, W., Liu, Q. (2022). Transfer learning with limited samples for the same source hyperspectral remote sensing images classification, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **XLIII-B3-2022**, 405–410.
`https://isprs-archives.copernicus.org/articles/XLIII-B3-2022/405/2022/`

Li, Z., Huang, H. (2022). Manifold-based multi-deep belief network for feature extraction of hyperspectral image, *Remote Sensing* **14**(6).
`https://www.mdpi.com/2072-4292/14/6/1484`

Mou, L., Ghamisi, P. (2018). Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing* **56**(1), 391–406.

MultiSpec, . (2020). Hyperspectral images, `https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html`.

Nanjing University, . (2024). Hyperspectral datasets, `http://lesun.weebly.com/hyperspectral-data-set.html`.

Nascimento, J., Dias, J. (2005). Vertex component analysis: a fast algorithm to unmix hyperspectral data, *IEEE Transactions on Geoscience and Remote Sensing* **43**(4), 898–910.

Paura, V., Marcinkevičius, V. (2024). Crop hyperspectral dataset unmixing using modified u-net model, *in* Lupeikienė, A., Ralyté, J., Dzemyda, G. (eds), *Digital Business and Intelligent Systems*, Springer Nature Switzerland, Cham, pp. 195–210.

Paura, V., Marcinkevičius, V. (2023). Benchmark for hyperspectral unmixing algorithm evaluation, *Informatica* **34**(2), 285–315.

Rasti, B., Zouaoui, A. (2023). Hysupp: An open-source hyperspectral unmixing python package, *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1134–1137.

Ronneberger, O., Fischer, P. (2015). U-net: Convolutional networks for biomedical image segmentation, *CoRR* **abs/1505.04597**.
`http://arxiv.org/abs/1505.04597`

Shi, Y., Han, L. (2022). A latent encoder coupled generative adversarial network (le-gan) for efficient hyperspectral image super-resolution, *IEEE Transactions on Geoscience and Remote Sensing* **60**, 1–19.

Specim, S. I. L. (2024). Specim afx10, `https://www.specim.com/products/specim-afx10/`.

Wang, X., Zhong, Y. (2017). Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing, *IEEE Transactions on Geoscience and Remote Sensing* **55**(11), 6287–6304.

Yokoya, N., Yairi, T. (2012). Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion, *IEEE Transactions on Geoscience and Remote Sensing* **50**(2), 528–537.

Zhang, J., Cai, Z. (2022). Hyperspectral image denoising via adversarial learning, *Remote Sensing* **14**(8).
`https://www.mdpi.com/2072-4292/14/8/1790`

Zhang, S., Li, J. (2018). Spectral–spatial weighted sparse regression for hyperspectral image unmixing, *IEEE Transactions on Geoscience and Remote Sensing* **56**(6), 3265–3276.

Zheng, X., Jia, J. (2022). Hyperspectral image classification with imbalanced data based on semi-supervised learning, *Applied Sciences* **12**(8).
`https://www.mdpi.com/2076-3417/12/8/3943`

Zhou, F., Hang, R. (2019). Hyperspectral image classification using spectral-spatial lstms, *Neurocomputing* **328**, 39–47. Chinese Conference on Computer Vision 2017.
`https://www.sciencedirect.com/science/article/pii/S0925231218309573`

Received March 11, 2025 , revised June 16, 2025, accepted June 18, 2025