https://doi.org/10.15388/vu.thesis.828
https://orcid.org/0000-0002-4315-729X

VILNIUS UNIVERSITY

Vytautas Paura

Hyperspectral Unmixing of Hyperspectral Data Gathered Using an UAV

DOCTORAL DISSERTATION

Natural Sciences,
Informatics (N 009)

VILNIUS 2025

The dissertation was prepared between 2020 and 2024 at Vilnius University.

Academic Supervisor – Prof. Dr. Virginijus Marcinkevičius (Vilnius University, Natural Sciences, Informatics – N 009).
Academic Consultant – Assoc. Prof. Dr. Valdas Rapševičius (Vilnius University, Natural Sciences, Informatics – N 009).

This Doctoral Dissertation will be Defended in a Public Meeting of the Dissertation Defence Panel:

Chairman – Prof. Habil. Dr. Gintautas Dzemyda (Vilnius University, Natural Sciences, Informatics – N 009).
Members:
Prof. Dr. Dalius Matuzevičius (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – T 007),
Prof. Dr. Audris Mockus (University of Tennessee, USA, Natural Sciences, Informatics – N 009),
Assoc. Prof. Dr. Stepas Toliautas (Vilnius University, Natural Sciences, Physics – N 002),
Prof. Dr. Povilas Treigys (Vilnius University, Natural Sciences, Informatics – N 009).

The dissertation shall be defended at a public meeting of the Dissertation Defence Panel at 12 a.m. on 29th September 2025 in room 203 of the Institute of Data Science and Digital Technologies of Vilnius University.
Address: Akademijos str. 4, LT-04812, Vilnius, Lithuania
Tel. +370 5 210 9300; e-mail: info@mii.vu.lt

The text of this dissertation can be accessed at the Library of Vilnius University, as well on the website of Vilnius University: *https://www.vu.lt/lt/naujienos/ivykiu-kalendorius*.

https://doi.org/10.15388/vu.thesis.828
https://orcid.org/0000-0002-4315-729X

VILNIUS UNIVERSITETAS

Vytautas Paura

Medžiagų ir jų kiekių nustatymas hiperspektriniuose duomenyse surinktuose naudojant bepiločius orlaivius

DAKTARO DISERTACIJA

Gamtos mokslai,
Informatika (N 009)

VILNIUS 2025

Disertacija rengta 2020–2024 metais Vilniaus universitete.

Mokslinis vadovas - prof. dr. Virginijus Marcinkevičius (Vilniaus universitetas, gamtos mokslai, informatika – N 009).
Mokslinis konsultantas - doc. dr. Valdas Rapševičius ((Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Gynimo taryba:
Pirmininkas – prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, gamtos mokslai, informatika – N 009).
Nariai:
prof. dr. Dalius Matuzevičius (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija - T 007),
prof. dr. Audris Mockus (Tenesio universitetas, JAV, gamtos mokslai, informatika – N 009),
doc. dr. Stepas Toliautas (Vilniaus universitetas, gamtos mokslai, fizika – N 002),
prof. dr. Povilas Treigys (Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Disertacija ginama viešame Gynimo tarybos posėdyje 2025 m. rugsėjo 29 d. 12 val. Vilniaus universiteto Duomenų mokslo ir skaitemeninių technologijo intituto 203 auditorijoje. Adresas: Akademijos g. 4, LT-04812, Vilnius, Lietuva, tel. +370 5 210 9300; el. paštas: info@mii.vu.lt.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje ir Vilniaus universiteto interneto svetainėje adresu: *https://www.vu.lt/lt/naujienos/ivykiu-kalendorius*.

ABSTRACT

Hyperspectral Imaging (HSI) is a remote sensing technique that is growing in popularity. Because hyperspectral cameras can collect large amounts of spectral data, HSI represents a fast, non-destructive method for the analysis of data on materials, food, agriculture, and other fields. Over the decades, the capabilities of HSI have expanded to include cameras mounted on conveyor lines, unmanned aerial vehicles, satellites, and other techniques. Due to the complexity of their sensors and the large amounts of data that hyperspectral cameras can collect, the spatial resolution of this data is still significantly lower than that of conventional cameras. This, in turn, leads to the problem of data gathered by hyperspectral cameras being formed of a mixture of light from multiple materials in each pixel of the image. This effect is even more exaggerated when analysing remote sensing data collected by satellites and UAVs flying at high altitudes. Hyperspectral Unmixing (HU) algorithms extract the data of individual materials, called endmembers, and the amounts thereof from hyperspectral images. HU methods are used to calculate the possible number of endmembers, their unique spectral signatures, and their abundances in each hyperspectral image pixel. This study aims to propose a new HU algorithm based on deep convolutional neural networks that is more efficient, accurate, and resilient to noise than existing algorithms when used for hyperspectral data using a UAV. A new algorithm is created based on the U-Net architecture, with better performance than the state-of-the-art transformer-based HU model. Reconstruction error is improved by around four times when tested using UAV-gathered hyperspectral data. To further evaluate the new HU algorithm, a HU benchmarking methodology is created.

ADMM	Alternating Direction Method of Multipliers.
BFHUD	Blueberry Field HU Dataset.
HSI	Hyperspectral Imaging.
HU	Hyperspectral Unmixing.
HUNET	Hyperspectral Unmixing U-NET.
LMM	Linear Mixture Models.
MSE	Mean Squared Error.
NMF	Non-Negative Matrix Factorization.
PLS	Partial Least Squares.
RE	Reconstruction Error.
RMSE	Root Mean Squared Error.
RSNR	Reconstruction Signal-To-Noise Ratio.
SAD	Spectral Angle Distance.
SAM	Spectral Angle Mapping.
SNR	Signal-To-Noise Ratio.
SRE	Signal Reconstruction Error.
UAV	Unmanned Aerial Vehicle.
USGS	United States Geological Survey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Hyperspectral Imaging (HSI) is a remote sensing technique that collects data across the entire electromagnetic spectrum to obtain spectral data for each pixel in a scene. This enables a non-destructive method for gathering data for material identification from spectral signatures, object detection, and other tasks related to material analysis.

Research Area

HSI is utilised as a tool for gathering detailed material data in various scientific fields, including geology, astronomy, agriculture, biology, and surveillance. With the growing popularity of HSI being used on UAVs, the need for improved hyperspectral analysis algorithms increases. In turn, part of this research focuses on agricultural hyperspectral data analysis gathered using UAVs, and finding solutions to existing problems such as:

- The lack of available open hyperspectral datasets gathered using UAVs.
- The absence of standardized testing for Hyperspectral Unmixing (HU) algorithms.
- The paucity of research on HU methods for hyperspectral data gathered by UAVs.

To address these issues, research was conducted on the methodologies for testing HU algorithms, existing HU algorithms, deep neural network models used for HU, and the creation methods for existing HU datasets. The majority of the work concentrates on hyperspectral data analysis and the study of the HU methods used to extract spectral and material information from HSI.

Research Problem

With the growing popularity and more widespread use of HSI [10], research in this area is also increasing. One of the research areas in the HSI field is HU [12]. HU combines one or multiple algorithms that extract material information from individual pixels in the hyperspectral

image. HU is required because spectral data in each pixel may be formed of a mixture of multiple materials. This problem arises due to the lower spatial resolution of data gathered by hyperspectral sensors and the light scattering of different materials.

The primary problem addressed in this dissertation is the development of the HU algorithm for use in hyperspectral field data collected using UAVs. The research was conducted to develop a more accurate HU algorithm using deep neural networks, as existing algorithms for agricultural field data analysis are lacking.

Actuality

With the growing popularity of HSI and UAVs in agriculture for crop monitoring and analysis, this research expands on existing hyperspectral analysis methods. It broadens the use of HU algorithms to UAV-gathered agricultural hyperspectral data. Additionally, a standardised HU algorithm testing methodology is developed that can be used to assess newly created HU algorithms more accurately.

Research Object

The research object of this dissertation is both classical and modern deep neural network-based HU algorithms, and their application to UAV-gathered hyperspectral data.

Research Aim and Objectives

The main aim of this dissertation is to propose a more accurate, efficient, and robust to noise HU algorithm based on deep convolutional neural networks for data gathered using UAVs.

To achieve this aim, the following objectives were formulated:

- To benchmark existing HU algorithms on various datasets to evaluate the dependency between the algorithms' accuracy and white noise, image scale, and the number of endmembers.
- To analyse hyperspectral band correlations and deep learning-based algorithms for synthetic hyperspectral data generation from multispectral images.

- To propose a modified U-Net model for HU and to compare it with transformer-based HU models on hyperspectral data gathered using UAVs.
- To optimise the proposed modified U-Net HU model for better reconstruction of the ground truth of the UAV-gathered hyperspectral image.

Research Methods

Multiple research methods were employed to achieve the set objectives. First, an analysis was conducted of related work in hyperspectral data analysis, specifically regarding HU algorithms. Second, both the available hyperspectral dataset and the type used in HU were analysed. Third came research on dataset creation methods, including methods of gathering hyperspectral data using UAVs, to produce a HU dataset. Fourth, an experiment was conducted on multiple algorithms, including the newly created model, and multiple different HU datasets. Lastly, the model architecture was modified based on the experimental results to enhance the accuracy of the HU model on hyperspectral data collected using UAVs.

Scientific Novelty

Research on HU algorithms with a focus on creating deep learning neural network models enables HU performance to be improved. Moreover, creating an unsupervised deep learning method enables the use of hyperspectral data for unmixing tasks. The published HU dataset, based on agricultural hyperspectral data, allows the testing of newly created HU algorithms on a variety of datasets. A new HU algorithm was also developed for the analysis of agricultural hyperspectral data gathered using UAVs, and a HU benchmark was published that aims to provide a standardised way of comparing HU algorithms.

Practical Significance

From the research and experiments conducted, the following main practical contributions can be identified:

1. Created a HU benchmarking methodology designed to be used in testing any newly created HU algorithm.

2. Created a new hyperspectral dataset from field data to be used for HU or classification algorithm testing.

3. Newly proposed an unsupervised HU algorithm based on the U-Net architecture with better performance than a transformer-based method for use in HU for agricultural hyperspectral data gathered using UAVs.

4. Proposed an unsupervised HU algorithm that provides easier hyperspectral analysis, because ground truths are not required for model training and usage.

Statements to be Defended

Research statements to be defended:

1. The performance of classical sparse regression Hyperspectral Unmixing (HU) algorithms does not depend on the number of endmembers and the density of information regulated by an image scale.

2. Classical sparse regression and HU algorithms are more resilient to Gaussian-type noise and less resilient to real hyperspectral camera noise.

3. Hyperspectral data bands close to each other have a high correlation between them, and this relationship can be exploited to generate better synthetic multispectral data.

4. The modified U-Net type model architecture proposed in this study requires fewer hyperparameters than the tested transformer HU models to achieve the most accurate unmixing results when used on UAV-gathered hyperspectral data.

5. The proposed modified U-Net HU model has the potential for lower reconstruction error than transformer-based unmixing algorithms when used on UAV-gathered hyperspectral data.

Approbation and Publications of the Research

Articles in CA WoS journals:

[A.1] Paura, Vytautas & Marcinkevičius, Virginijus. Hyperspectral Unmixing of Agricultural Images Taken From UAV Using Adapted U-Net Architecture. Paper accepted into *Baltic Journal of Modern Computing*.

[A.2] Paura, Vytautas & Marcinkevičius, Virginijus. Benchmark for Hyperspectral Unmixing Algorithm Evaluation. *Informatica* (Vilnius University Press, 2023). **34**, 285-315 (2024). https://doi.org/10.15388/23-INFOR522

Publications without citation index:

[B.1] Paura, Vytautas & Marcinkevičius, Virginijus. Crop Hyperspectral Dataset Unmixing Using Modified U-Net Model. *Digital Business and Intelligent Systems*. **2157**, 195-210 (2024). https://doi.org/10.1007/978-3-031-63543-4_13

[B.2] Paura, Vytautas & Marcinkevičius, Virginijus. Semi-supervised hyperspectral unmixing dataset creation methods for unmixing algorithm analysis. *Image and Signal Processing for Remote Sensing XXIX* (SPIE, 2023). **12733**, 127330Y (2023). https://doi.org/10.1117/12.2679826

[B.3] Paura, Vytautas & Marcinkevičius, Virginijus. The Analysis of Impact of Noise on Hyperspectral Unmixing Algorithms. *Data Analysis Methods for Software Systems*. Bernatavičienė, Jolita. (Vilnius University Press, 2021). **17**, 56 (2021). https://doi.org/10.15388/DAMSS.12.2021

Conferences and presentations:

[c.1] Paura, Vytautas & Marcinkevičius, Virginijus. Crop Hyperspectral Dataset Unmixing Using Modified U-Net Model. *Digital Business and Intelligent Systems*. Vilnius, Lithuania (2024).

[C.2] Paura, Vytautas & Marcinkevičius, Virginijus. Semi-Supervised Hyperspectral Unmixing Dataset Creation Methods For Unmixing Algorithm Analysis. *Image and Signal Processing for Remote Sensing XXIX*. Amsterdam, Netherlands (2023).

[c.3] Paura, Vytautas & Marcinkevičius, Virginijus. Hyperspectral data synthesis using deep neural networks. *Eastern European Machine Learning Summer School*. Vilniu, Lithuania (2022).

[C.4] Paura, Vytautas & Marcinkevičius, Virginijus. The Analysis of Impact of Noise on Hyperspectral Unmixing Algorithms. *Data Analysis Methods for Software Systems*. Druskininkai, Lithuania (2021).

Outline of the Thesis

This doctoral thesis comprises an introduction, three chapters, conclusions, and a summary, all written in the Lithuanian language. The introduction section provides an overview of the research and the dissertation as a whole. The first chapter is a literature review of hyperspectral data analysis. The second chapter discusses HU algorithms, experimentation methodology, and the proposed HU model. The final chapter focuses on experimentation and benchmarking. At the end of the thesis, 118 bibliographic references are included. The dissertation comprises 164 pages, 51 figures, and 11 tables.

This chapter describes the theory and the basis of HSI and its use cases in the remote sensing field, offers a brief introduction to the methods used to analyse hyperspectral data cubes and a description of HU, and outlines the available algorithms. The chapter is based on research into currently available HU algorithms, published research, state-of-the-art methods, and the theory behind hyperspectral data collection and HU [A.1] and [B.1].

1.1. Hyperspectral Imaging

HSI is a branch of remote sensing technology used to gather information about objects for research and analysis without any physical contact. HSI combines the imaging technologies used in regular cameras with spectroscopy methods to create sensors capable of collecting data from electromagnetic waves in multiple wavelengths along with spatial information.

Spectroscopy. HSI is most commonly used to collect spectrometric data from electromagnetic bands in the visible light, near-infrared, short-wave infrared, ultraviolet, and medium-wave infrared wavelengths. This is achieved with sensors that collect electromagnetic radiation reflected from the object into the sensor or via the scattering of electromagnetic radiation by these objects. A hyperspectral camera captures the light of a scene, separating it into its constituent wavelengths or spectral bands. It then provides a two-dimensional scene image while recording each pixel's spectral information in the image [91].

Remote sensing. Detecting electromagnetic radiation from the interaction between objects and camera sensors is considered a form of remote sensing technology because it operates at a distance, without physical interaction with any objects. Due to the nature of the spatial and spectral data collected with HSI sensors, it is most commonly collected from unmanned aircraft or satellites, and less commonly by placing cameras on factory conveyors. Most sensors mounted on aircraft or satellites present additional challenges compared to laboratory spectroscopy sensors. The authors of *Hyperspectral Imaging Remote Sensing* [62] distinguish four components of HSI:

- A source of electromagnetic radiation, which can be natural or man-made. With hyperspectral sensors mounted on aircraft or satellites, most radiation will be from the Sun or other blackbody radiation sources.
- With electromagnetic radiation travelling through the atmosphere, the radiation is affected by absorption or scattering and other interactions with air molecules. Figure 1.1 shows the differences between ideal blackbody radiation, solar radiation, and sunlight at sea level.
- The spectral radiation emitted or reflected by the materials on the surface.
- A sensor that converts electromagnetic energy into electrical signals and records the data. These sensors can record additional noise and distortion due to the intricate optics inside hyperspectral cameras.

1.1.1. Hyperspectral Data

Using new sensor technologies, optics, and spectroscopy, HSI cameras can collect large amounts of remote sensing data for future research. Remote sensing spectral sensors are divided into two main categories: hyperspectral and multispectral, with the main difference being the number of separate electromagnetic bands that the sensors collect. Typically, a spectral sensor is considered hyperspectral if it collects more than 100 separate bands. In turn, multispectral sensors have a higher spatial resolution due to their simpler design and reduced data transfer requirements [46].

Data sources. The most commonly used sensors in remote sensing are multispectral sensors. They feature specific bands, usually around 10 in total, to capture the reflectance of the visible RGB spectra, as well as some near-infrared or short-wave infrared bands for water and vegetation analysis. These sensors are common in satellites because of their balance between high usability, spatial resolution, and cost-effectiveness. The use of hyperspectral cameras in satellites is growing, but the costs

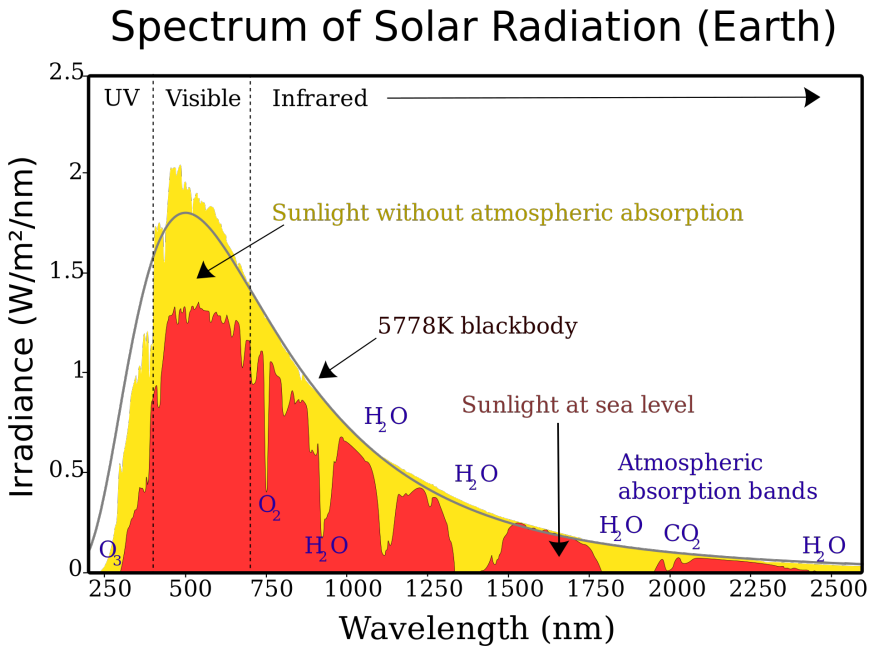


Figure 1.1 Comparison of the blackbody spectrum of the Sun and sunlight at sea level. Source [79].

of these sensors and their data are still too high to be widely adopted. Since most common use cases involve agriculture, vegetation, or landscape monitoring applications, hyperspectral sensors offer little benefit compared to multispectral instruments. In turn, hyperspectral sensors are used in smaller-scale experiments on local fields when mounted over conveyors, being either portable or mounted on a UAV.

Light reflectance. Apart from a few specific laboratory or conveyor uses, most hyperspectral sensors gather sunlight reflected from objects on the surface. The data collected relies on good atmospheric conditions for further research, especially when working with satellite data. As shown in Figure 1.1, the atmosphere absorbs some wavelengths of light more than others, with, for example, water absorbing all helpful information in some spectral ranges. Hyperspectral cameras attached to a UAV flown at low altitudes are more resilient to atmospheric effects because sunlight only passes through the atmosphere once, but the data is still subject to the influence of clouds and the sun's position in the

sky. Because of this, remote sensing data should be collected as close to midday as possible. For example, Sentinel-2 data is collected when the satellite is in orbit, when the Sun zenith angle is below a certain threshold (85 degrees over Europe) [69].

Signalling and noise. Hyperspectral data sensors are highly sensitive, and have the ability to gather data from hundreds of narrow spectral bands. This, in turn, may introduce a substantial amount of noise into their images. Camera manufacturers provide the characteristics of these sensors, and the spectral bands at the edges of sensor capabilities are the most prone to noise. It is sometimes advisable to discard the noisier spectral bands from gathered images. One of the techniques used to assess the noise of spectral bands is a calculation of the Pearson correlation coefficient for neighbouring band pairs [84]. The correlation coefficient for a camera's neighbouring bands used in the gathering of data for this research is provided in Figure 1.2. The formula used to calculate the Pearson correlation is given in equation 1.1

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{1.1}$$

, where n is the number of samples, x_i and y_i are samples, $\overline{x}$ and $\overline{y}$ are the averages of these samples, and r_{xy} is the resulting correlation. In this case, x represents all values from the hyperspectral data for the first selected band, and y represents the values for the second band from between which the correlation is calculated.

1.1.2. Hyperspectral Sensors

Sensor types. Most hyperspectral cameras fall into 4 main categories [33] [24]:

- **Whiskbroom cameras** gather all the bands from a single pixel at a time. These sensors are the slowest but may have higher spatial and spectral resolution than other sensor types.
- **Wavelength scanning cameras** gather all spatial data for a single spectral band at a time. These cameras may have a higher spectral resolution (the sensor can collect more separate spectral bands

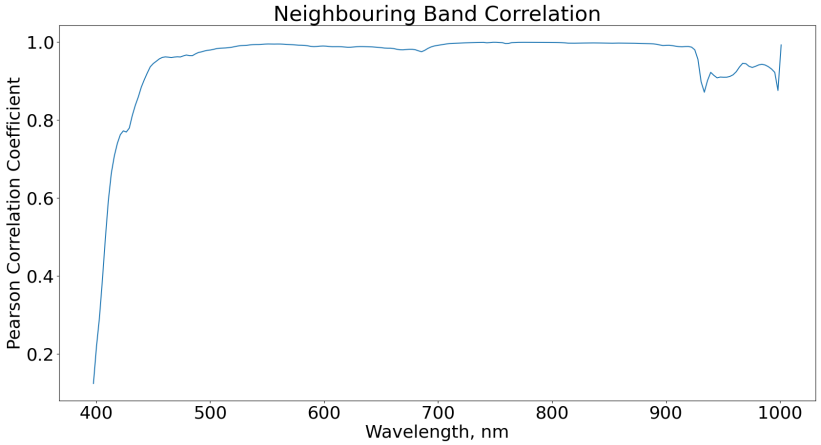


Figure 1.2 Pearson correlation coefficient for neighbouring bands.

for each pixel), but a lower spatial resolution (the total number of pixels in the image may be lower). These cameras are less suitable for capturing moving objects or scanning them. They are also poorly suited to mounting on drones because the cameras require some time to capture all spectral bands into a single hyperspectral data cube, creating the potential for spectral band layer shifting.

- **Pushbroom cameras** gather all spectral band data from a line of pixels simultaneously, and the data is gathered by moving the camera forward or the samples backwards to record different data for each line. This type of camera was used to collect data for this research due to its gathering of accurate spectral data for each pixel, compared to the possible spectral shifting of wavelength scanning cameras when used on a UAV. The gathered hyperspectral data cubes have a fixed width, but their length depends on the drone's flight distance.

- **Snapshot cameras** are a new type of sensor technology that can capture entire hyperspectral cubes at the same time. Due to limitations in data transfer and sensor complexity, these cameras have lower spatial and spectral resolutions. This is a new technology, and is rarely used as a result.

1.1.3. Hyperspectral Imaging Use Cases and Research In Lithuania

Research on remote sensing and HSI is being conducted across multiple universities in Lithuania, as documented in *Žemės stebėjimas iš kosmoso* (*Earth Observation From Space*) by Kilpys et al. [45], published by Vilnius University. This textbook teaches students about remote sensing technologies in university courses, and covers topics such as:

- An introduction to electromagnetic radiation.
- Types of sensors used in satellites for remote sensing and how they work, with the example of the hyperspectral sensors used by NASA satellites.
- Possible use cases for the Earth observation data: agriculture, forestry, extreme climate conditions, water, and icy area observations.

Research on HSI cameras mounted on ultralight aircraft has been conducted by Mozgeris et al. [67]. Hyperspectral images, combined with a colour-infrared (CIR) camera, were used to study the potential for recognising tree species in urban environments over Kaunas. The classification accuracy of the tree species results varied from poor to moderate, with an overall accuracy of approximately 63%.

1.1.4. Hyperspectral Data Calibration

Calibration Plates. Hyperspectral data calibration is usually performed using reflectance plates when using cameras mounted on a UAV and other similar devices. These plates are laboratory-calibrated to have a uniform light reflectance percentage across various electromagnetic spectra. A calibration plate with the required spectral range should be used for optimal efficiency, as more general plates covering broader spectra are often less accurate. These plates are created to have a nearly Lambertian reflectance, which means a uniform light reflectance and apparent brightness of the plate from any viewing angle [49]. An example reflectance plate with a calibrated value of 99% reflectance is shown in Figure 1.3. Example spectral radiance values of three different calibration plates with reflectances of 5%, 10%, and 40% collected by the hyperspectral camera used in this research are shown in Figure 1.4,

where the x axis represents the gathered wavelength index and the y axis the radiance value.

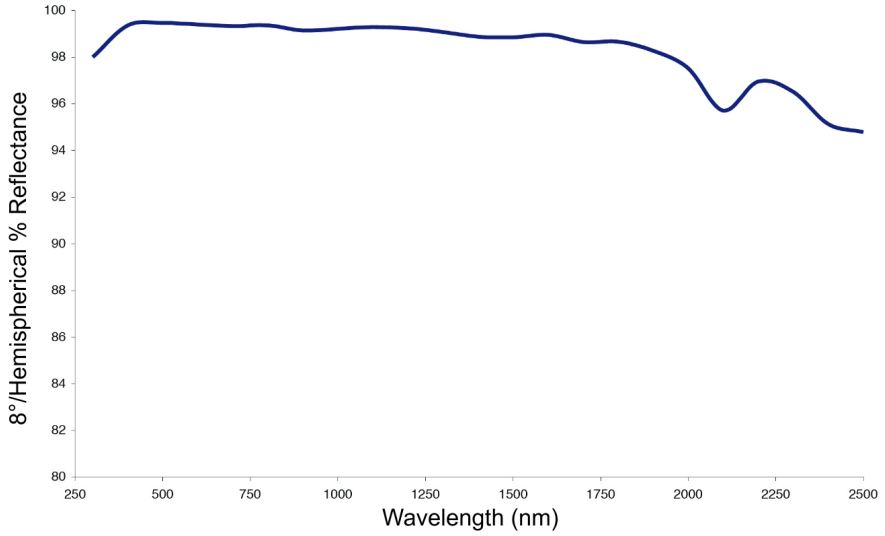


Figure 1.3 Reflectance profile of Spectralon 99% reflectance target plate. Source: [50].

Calibration Methods. One of the primary methods for calibrating hyperspectral data is the use of the aforementioned reflectance plates. Since hyperspectral sensors gather light data and record it as a radiance value, the data is not directly comparable between different data collection missions due to variations in solar irradiance at the time of data gathering. For example, the camera used in this research collects data that is radiometrically corrected to radiance measured in watts per square centimetre per steradian multiplied by a thousand $(mW/cm^2 * sr) * 1000.0000$. Using reflectance plates, we can recalculate these radiance values into reflectance percentages, which remain constant for the same material when scanned at different times. To calculate reflectance from radiance, Burger et al. [15] proposed a calibration model, as shown in Figure 1.5. This model utilises multiple calibration plates for improved reflectance accuracy; however, calculations can also be performed using only dark reflectance (0%) and white reflectance plates (99%). An actual 0% reflectance can be achieved by filming with a closed lens, which gathers

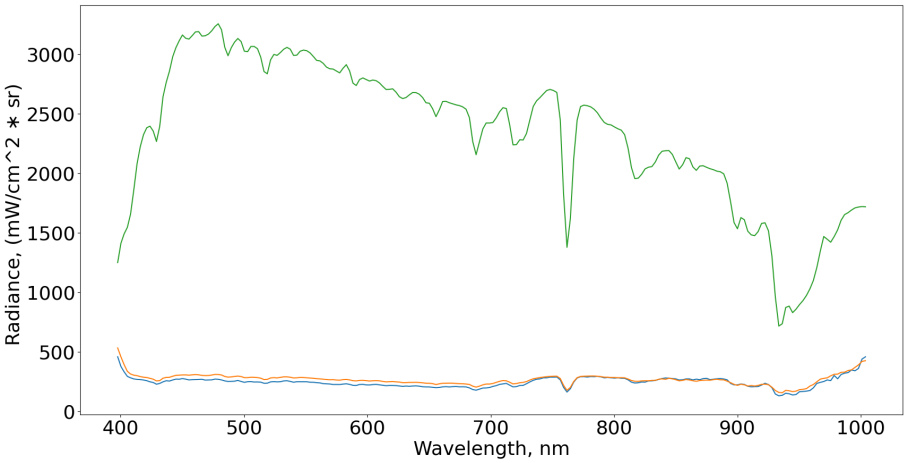


Figure 1.4 Example calibration spectral values gathered by the hyperspectral camera from three calibration plates with reflectances of 5%, 10%, and 40%.

only sensor noise data with no light. Hyperspectral data calibration code written in *Python* using the *NumPy* computation library is provided in Appendix D. The given code calculates calibration lines for each wavelength separately due to possible variations in reflectance percentages. Using the *Numpy polyfit* method calculates a line function that converts sensor radiance values into reflectance. The poly fit degree is currently set to 1, creating a linear conversion function with $reflectance = radiance * \alpha + \beta$, shown in line 30. *Numpy polyfit* fits a curve (a line if the degree is 1) over a collection of X and Y points, and then creates α and β parameters to then calculate Y given any X, which in this case is the raw hyperspectral radiance data.

1.1.5. Hyperspectral Imaging Analysis Methods

HSI is a remote sensing method that is increasingly being used in various fields for fast and accurate material analysis in a non-destructive manner. These cameras are used on conveyors, UAVs, and satellites for gathering remote sensing data. Due to the availability of data and the increasing popularity of remote sensing technologies in agriculture, this research focuses on gathering and analysing data collected in crop fields used for agricultural purposes.

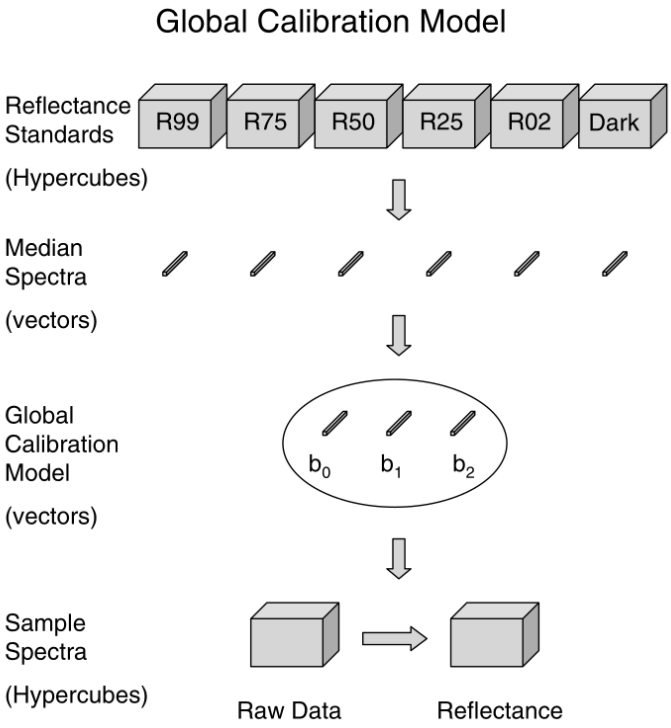


Figure 1.5 Hyperspectral image calibration model based on multiple reflectance plates. Source: [15].

Food analysis. One of the more widespread use cases of HSI is in the food industry for food analysis and quality control applications. Qin et al. [82] describe the usage of HSI as a quick and non-destructive method for evaluating food quality and parameters. A non-destructive method is necessary for food analysis in order to preserve food quality and other food safety factors. The authors focus on the capabilities of HSI to collect large amounts of spectral data from a wide area, compared to traditional spectrometry methods that were used in the past.

Nikzadfar et al. [73] expand on the use of HSI for food quality assessment and the various artificial intelligence methods used for hyperspectral data analysis. Their research focuses on food quality analysis using hyperspectral data for contaminant detection, spoilage prediction, and food safety applications. The methods described in the article are split into three categories, and their use in hyperspectral data analysis is explained in detail:

- Preprocessing:
  - Savitzky–Golay is a noise reduction and smoothing tool that uses a polynomial function with a moving window to smooth signal data. This method is commonly used in other signal-processing applications. Figure 1.6 shows an example of data filtered using this algorithm. This technique is used to smooth the spectral dimension of HSI and reduce noise, thereby compressing the data [8].
  - Multiplicative Scatter Correction uses regression computation to remove an estimate of the scattering component from the spectral data. Combined with other preprocessing methods, it can improve the accuracy and performance of prediction models [87].
  - Auto Scaling is a data normalisation method that standardises data for each spectral band separately, so the data has a mean of 0 and a standard deviation of 1. The algorithm is susceptible to outliers similar to the min-max scaler [23].
  - Log (1/R) uses logarithmic transformation to create linear data between wavelength and variables, and is used when data is not linear (see equation 1.2). According to the authors of the aforementioned article, this method has been widely used in food moisture monitoring applications, lipid prediction [18], and and the assessment of food powder mixtures [2].
- Hyperspectral Wavelength Selection:
  - Competitive Adaptive Reweighted Sampling is a variable selection method that uses Monte Carlo sampling, exponential decreasing function (EDF), and competitive variable selection based on adaptive reweighted sampling (ARS), with cross-validation using the RMSE metric [102]. The method aims to maintain robust data and minimise collinearity [87].
  - Principal Component Analysis (PCA) is a commonly used dimensionality reduction algorithm that transforms data into orthogonal dimensions (principal components), which retain most of the original information, based on their variance. It is

commonly used for data visualisation due to the algorithm's ability to transform data into two or three dimensions. Moreover, PCA improves the classification efficiency of HSI in agricultural applications [20].

  - The Successive Projections Algorithm is a dimensionality reduction method that projects variables to minimise variable collinearity in order to select the most informative wavelengths from a hyperspectral data cube. This method was successfully employed to choose the optimal wavelengths for seafood analysis tasks in predicting food freshness [109].

- Machine Learning:

  - Linear Discriminant Analysis is a supervised algorithm that is used to classify data or perform dimensionality reduction. The algorithm maximises class separability and can help prevent overfitting by reducing dimensions while retaining class-related features. It has been used to reduce dimensions in hyperspectral data by finding a linear combination of features that maximises the separation between different classes [25].
  - Decision Trees split data into subsets based on feature values, creating a tree-like structure where nodes represent decision rules and branches represent outcomes. Food quality assessment is the field in which decision trees can be applied in HSI to evaluate different food quality parameters with high accuracy. For example, analysing the quality of black tea [83].
  - Random Forest is an ensemble machine learning algorithm that trains many decision trees and combines the predictions of individual trees to improve the predictive performance of classification or regression. It is robust and capable of handling high-dimensional data, which gives it an advantage over the simpler decision tree algorithm for hyperspectral data analysis [5].
  - K-Nearest Neighbour is an effective classification and regression analysis algorithm. It simply classifies unknown food items by matching their spectral signatures with those of known ones. K-nearest neighbours assume that similar food

items are located around one another in a feature space. One of its use cases is to classify tea to detect mould in dried tea leaves [100].

- Cluster Analysis is an unsupervised statistical method of grouping objects into classes based on their characteristic features to maximise their similarity within one class and minimise similarity between classes. Significant techniques in cluster analysis can be divided into hierarchical and non-hierarchical methods. Hierarchical clustering structures data in a tree-like fashion, which can provide a way to visually show the relationships between data points at a higher level of similarity. Cluster analysis has been used to predict the amount of wholemeal flour in bread [64].
- Deep Learning is an aspect of machine learning which involves using multi-layer artificial neural networks to interpret large volumes of data for specific purposes, even complex ones. Deep learning addresses the training of neural networks to extract features from the raw data. The multiple layers interconnected in deep neural networks enable the representation of complex relationships between inputs and outputs. A review by Gul et al. [31] shows that an increasing number of research papers are being produced in the HSI field, and highlights the growing popularity of deep learning methods for hyperspectral data analysis.

The *Log (1/R)* transformation is used in spectroscopy to transform sample transmittance or reflectance to absorbance, and is calculated using the following equation:

$$A = log_{10}(\frac{1}{R}) \tag{1.2}$$

, where R is the sample reflectance for a specific wavelength and A is the calculated absorbance.

Agriculture and the Food Industry. Caporaso et al. [18] describe the usage of HSI to predict fat content in beans, with a final resulting coefficient of determination which shows the proportion of variance

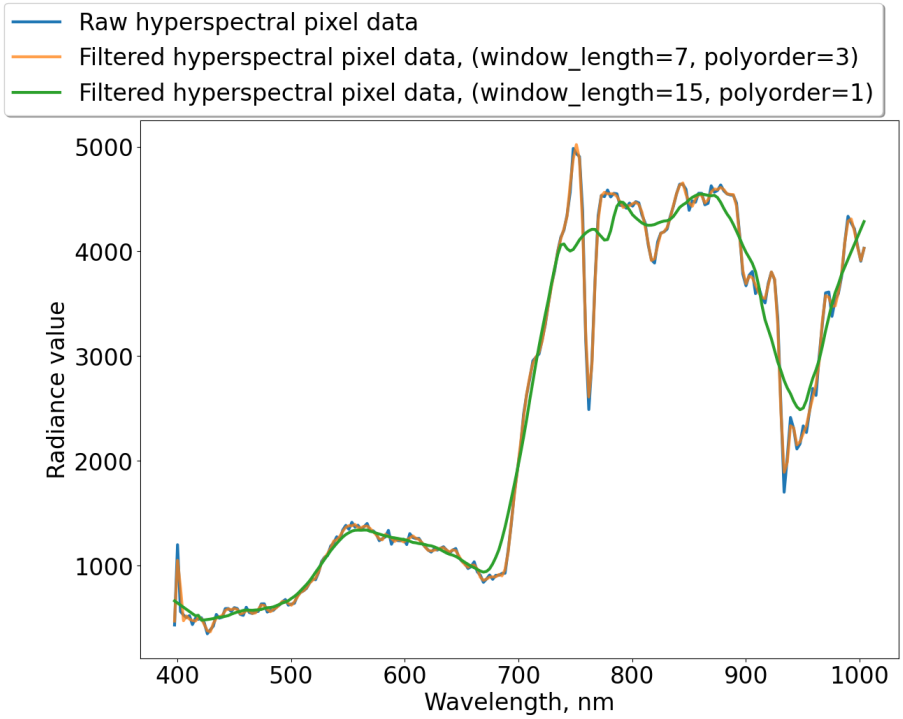


Figure 1.6 Hyperspectral data comparison between raw data and data denoised using a Savitzky–Golay filter.

that can be explained by the independent variable, being $R^2 = 0.85$, for single-shelled beans.

Similarly, Achata et al. [2] describe the usage of visible and near-infrared light HSI, as well as Raman spectroscopy, for the analysis and prediction of food powder mixtures. In their experiment, the authors used HSI with Partial Least Squares (PLS) regression algorithms to determine the mixed food powders of various concentrations.

Universal Hyperspectral Analysis Algorithms. Expanding on the methods reviewed by Nikzadfar et al. [73] for analysing hyperspectral data in food quality control applications, this section reviews hyperspectral data analysis algorithms that can be applied to any application [106].

- The Pixel Purity Index is an endmember extraction algorithm. This method was combined with Maximum Noise Fraction (MNF)

transformation, dimensionality reduction, and a denoising algorithm to iteratively project data to random unit vectors to determine pixel purity [96].

- The Automatic Target Generation Process (ATGP) is an algorithm used for unsupervised hyperspectral target detection. It extracts targets without class data using orthogonal subspace projections [103]. For hyperspectral data, the algorithm finds vectors at the ends of the constructed projections because these vectors have maximal orthogonal projection and are more likely to be endmembers.

- Minimum Volume Simplex Analysis is a fast HU algorithm that assumes a linear mixture model. The hyperspectral data cube is created from endmember spectral signatures weighted linearly by the abundance fractions. Jun Li and Jose M. Bioucas-Dias proposed an algorithm [53] that produced state-of-the-art performance in the HU of synthetic data cubes.

- Vertex Component Analysis (VCA) represents unsupervised endmember extraction from the hyperspectral data method [72]. The usage of the algorithm is explained in more detail in chapter 2.3.1. ATGP and VCA can be considered variants of each other, with the main difference being that a Gaussian random number generator is used to generate the initial set of endmembers.

Band Selection [6]. Based on whether prior knowledge is used, band selection methods can be classed as supervised or unsupervised [9]. Due to its more robust performance and higher application prospects, the unsupervised band selection method has garnered considerable attention over the past few decades. Based on a review paper by Sun and Du [95], hyperspectral band selection methods can be categorised into six groups:

- Ranking-based methods, which select spectral bands according to a predefined band-prioritisation criterion.

- Searching-based methods, which convert band selection into an optimisation problem for a given criterion function. The two

parts of these methods are the criterion function and the searching strategy.

- Clustering-based methods, which create clusters from the original bands and select the representative bands from these clusters.
- Sparsity-based methods, which use sparse representation or regression to find structures in HSI data.
- Embedding learning-based methods, which perform band selection as part of the optimisation step of the other models used, for example, during classification.
- Hybrid methods, which combine multiple previous methods for better band selection performance.

One of the searching-based methods used for band selection is described by Cai et al. [17]. The key to BS-Nets is to convert the band selection as a sparse band reconstruction task, i.e., to recover complete spectral information using a few informative bands. If the information is essential for a given spectral band, it will be necessary for spectral reconstruction. To this end, we design a deep neural network based on the attention mechanism. The architecture of the proposed framework consists of three components: the band attention module (BAM), band re-weighting (BRW), and the reconstruction network (RecNet).

An example of a hybrid ranking and clustering-based method is proposed by Phaneendra et al. [78]. The authors suggest the usage of correlation methods, which partition a HSI data cube based on the correlation coefficient, then generate a weight matrix to quantify how each pixel is approximated by neighboring pixels, then extract the intrinsic information from the selected bands, and finally train a three-dimensional convolutional neural network for classification and benchmarking the band selection results.

1.1.6. Hyperspectral Data Synthesis

This section describes research on hyperspectral data synthesis and generation algorithms, which could potentially increase the availability of hyperspectral data and further our understanding of how the data behaves under various conditions. HSI is becoming a popular remote

sensing method that is widely used in agriculture, the food industry, and mineralogy. However, due to the high costs of hardware, acquiring hyperspectral data is challenging and expensive. A possible solution to this problem is the use of hyperspectral data synthesis. Data synthesis models are needed to efficiently develop remote monitoring applications, including sensor development, testing, design, launch, and algorithm creation. Users and researchers could artificially create hyperspectral data from more widely available multispectral data or RGB images by creating an accurate synthesis model.

1.1.6.1. Data Synthesis Models

To create these synthesising models, two approaches are commonly used:

1. A simulation model that predicts data from given parameters.
2. A deep neural network model trained on hyperspectral data to extrapolate from a smaller number of spectral bands to hyperspectral data.

One example of a simulation model is the Digital Imaging and Remote Sensing Image Generation (DIRSIG) model [29]. It is used to create an artificial scene of data and simulate light propagation at different wavelengths in those scenes. Figure 1.7 shows an example of generated spectral data compared to actual data.

1.1.6.2. Data Synthesis Using Deep Learning

Wang et al. [99] proposed a network architecture and a series of algorithms as the basis for research and experimentation in creating a deep neural network for hyperspectral data synthesis. After multiple iterations of model architecture and hyperparameter tuning, a hybrid, fully connected, and convolutional network architecture was created. The model architecture is shown in Figure 17 in Appendix B. The developed model was tested on RGB data constructed from a hyperspectral image cube. In turn, an input size of 20 × 20 pixels was used. For lower-resolution input data (e.g., satellite imagery), the model should be retrained to process one pixel at a time.

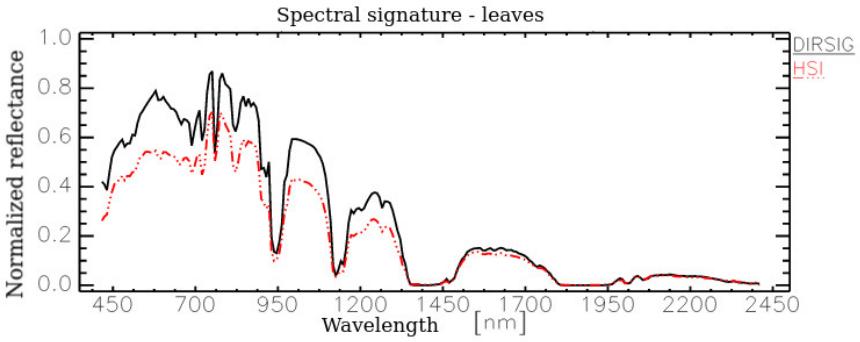


Figure 1.7 DIRSIG-generated hyperspectral data comparison, where the red line shows a generated hyperspectral signature and the black line shows a real spectral signature. Source: [29]

1.1.7. Hyperspectral Data Classification Algorithms

HSI classification is the predecessor problem to HU, and its resolution shares similar methods and ideas. In addressing these issues, Wenjing Lv and Xiaofei Wang [61] reviewed HSI classification methods and algorithms as follows:

- Support Vector Machine is a supervised classification method proposed by Boser et al. [14] that is used for classification and regression tasks. A non-linear classifier can be created using the kernel trick proposed by Boser et al. [14]. As a machine learning method, the support vector machine method plays a significant role in image and signal processing and recognition. It requires that the classification surface can not only separate the two types of sample points without error, but also maximise the classification gap between the two types.
- Minimum Distance Classifier is a supervised classification method that uses the distance of pixels in the feature space as a classification basis. In the feature space, feature points belonging to the same class are generally considered to be clustered in space. By using training samples, the algorithm forms regions of classes and uses a distance metric (for example, Euclidean) to measure the similarity between samples. An example that uses minimum-distance classifiers in the wavelet domain was proposed by Li et al. [55].

- Maximum Likelihood Classifier is a classification method based on the Bayesian criterion. The maximum likelihood of discrimination classification is a non-linear classification method. The algorithm calculates statistical features of training samples to create a classification discriminant function, which determines the probability of each sample belonging to each class [43].

- Artificial Neural Networks are the most popular artificial intelligence classification method. The back propagation neural network (feedforward network model) is currently the most widely used neural network model. It consists of an input layer, a hidden layer, and an output layer. However, artificial neural networks have weaknesses, such as the need for extensive training datasets and slower operation speed compared to classical algorithms. For example, a generative adversarial network combined with convolutional networks has been used for hyperspectral classification [118].

- Convolutional Neural Networks (CNN) are neural networks that learn features via kernel optimisation. This is the most popular network type for image processing, and has recently been replaced by transformer networks for some applications. One-, two-, or three-dimensional convolutional networks are viable for hyperspectral data. Using a CNN to classify the spectral features of hyperspectral images involves using a one-dimensional CNN to extract and classify spectral features for each pixel separately. Two-dimensional convolution can process groups of pixels, with the spectral data serving as the convolutional channel data. Three-dimensional convolution uses spatial and spectral information as inputs for small cubes of hyperspectral data. A U-Net network is a popular example of a convolutional neural network used for image classification [85].

An example of the use of deep convolutional networks for hyperspectral data classification is provided by Hyungtae and Heesung [51]. In their article, the authors propose a deeper convolutional network for improved classification performance compared to state-of-the-art convolutional models, emphasising finding the optimal width and depth combination for the CNN classification model.

A new hyperspectral analysis and unmixing model type is gaining popularity, utilising transformer layers that are now commonly employed in language processing tasks. As an example, Hong et al. [38] describe a transformer-based hyperspectral classification algorithm called SpectralFormer. Figure 16 in Appendix A provides the entire model architecture. The model consists of a group-wise spectral embedding layer based on the spectral patch size, which combines linear projections of neighbouring bands to learn the positional embeddings of spectral data. The embedded data is then encoded using the transformer-based encoding layer, and each spectral patch is encoded separately. In addition to the embeddings, a multi-layered perceptron performs the data classification. This consists of multiple parts in addition to the transformer network itself:

- Cross-layer Adaptive Fusion. The skip connection (SC) mechanism has been proven to be an effective strategy in deep networks, which can enhance information exchange between layers and reduce information loss in the network learning process. The information 'memory' ability of short SC remains limited, while long SC tends to yield insufficient fusion due to the presence of a large gap between high and low-level features. In turn, a new method called Cross-layer Adaptive Fusion was created.
- Group-wise Spectral Embedding (GSE). The spectral information in different positions reflects different absorption characteristics corresponding to other wavelengths. This, to a great extent, shows the physical properties of the current material. Capturing the locally detailed absorption (or changes) of such spectral signatures is crucial in accurately and finely classifying the materials in the hyperspectral scene. For this purpose, group-wise spectral embeddings are learned using the GSE algorithm, rather than band-wise input and representations.

1.2. Hyperspectral Unmixing

HU is the process that calculates the collection of spectral signatures, called endmembers, and their abundances for each pixel of the hyperspectral data cube. The HU problem can be separated into three main parts that HU algorithms try to solve separately or all at once:

- Finding the number of possible endmembers in each hyperspectral data cube.
- Extracting the spectral signatures of each of the endmembers.
- Calculating the abundance values of each endmember for all of the pixels in a hyperspectral data cube.

This section describes and reviews algorithms used for HU, and is split into four main parts that describe the different types of algorithms used and the datasets used to test these algorithms. These include supervised algorithms, semi-supervised algorithms, and unsupervised algorithms.

1.2.1. Supervised Algorithms

Supervised algorithms are most commonly used in machine learning to achieve optimal performance in various tasks, including object detection, classification, regression, and other areas. For HU, these algorithms are used less frequently because the supervised algorithms require labelled data for training, which may not be available if HU needs to be solved.

Koraila et al. [47], Bioucas-Dias et al. [11], and authors who have written on HU propose a supervised HU method that is based on the mapping of true hyperspectral image spectra and the corresponding linear spectra that are composed of the same endmember abundances. The suggested algorithm works via a series of steps:

- A real hyperspectral dataset is gathered.
- Ground truth abundance and endmembers are used to linearly mix spectra into an artificial hyperspectral image corresponding to the dataset.
- Both data sources are put into to a machine learning algorithm to learn the mappings between data.
- After training, the model is created and saved.
- The trained model is then tested with a portion of the real hyperspectral dataset.

- Linearly mixed spectra are generated due to the unmixing, resulting in the abundance map of the hyperspectral testing dataset.

Others used a neural network and two regression algorithms to learn the mapping between the generated linear and non-linear training spectra. The algorithms were tested using 10,000 mixed spectra with 50 dB Signal-To-Noise Ratio (SNR) Gaussian noise and spectral signatures from the United States Geological Survey (USGS) spectral library [48]. The spectral mixes were generated using the Hapke model [35].

1.2.2. Semi-Supervised Algorithms

Like supervised algorithms, semi-supervised algorithms require labelled data to be used properly. However, the key difference lies in the ability to learn the base structure from labelled data and the use of entirely unlabelled data to enhance training accuracy. Sparse regression is the most commonly used semi-supervised algorithm for HU.

In most common machine learning cases, the model is trained to learn the relationship between these variables, and the model created can in turn be used to predict the result based on input data. Sparse regression is a subcategory of regression machine learning algorithms. The number of possible independent variables, also referred to as predictors in HU, poses a challenge in collecting sufficient diverse data for machine learning algorithms to learn effectively. So, sparse regression models that depend on only a subset of possible predictors are used. The same regression algorithm can be used for sparse regression (linear, lasso, ridge, and others), but an additional step is often required to determine the subset of predictors. The problem of regression is achieving a function that allows one to estimate a certain quantity of the dependent variable from several observed variables, known as independent variables.

SUnSAL and C-SUnSAL. Sparse unmixing by variable splitting and both augmented Lagrangian (SUnSAL) and constrained SUnSAL (C-SUnSAL) algorithms [11] is based on the Alternating Direction Method of Multipliers (ADMM) algorithm [27]. The ADMM algorithm breaks down a complex problem into a series of more straightforward issues. The results provided by the aforementioned authors are in dB values

of the Reconstruction Signal-To-Noise Ratio (RSNR) metric, and both algorithms were tested using 50 dB of artificial noise. The SunSAL algorithm achieved RSNR values of 48 and 23 dB for the Gaussian and USGS [48] datasets, respectively, while C-SunSAL achieved 47 and 14.5 dB, respectively.

CLSUnSAL. Collaborative sparse unmixing by variable splitting and augmented Lagrangian (CLSUnSAL) [42] is an elaboration of a previous SUnSAL algorithm introduced in [11]. The difference between the SUnSAL and collaborative SUnSAL algorithms is that the non-constrained algorithm performs regression on each pixel independently, while the constrained algorithm calculates sparsity for all pixels. Algorithm performance results provided by the authors were in the SRE metric, with dB as the unit, and an artificial noise level of 40 dB SNR. The results were 21 dB for two endmembers, 14 dB for four endmembers, and 8.7 dB for six.

SUnSAL-TV. SUnSAL and total variation (SUnSAL-TV) [41] is a variation of the SUnSAL algorithm with an added total variation regularisation, which provides spatial information for better spectral unmixing results. The TV regulariser accounts for spatial homogeneity because it is very likely that neighbouring pixels will have a similar abundance fraction of the same endmembers. A similar algorithm, using total variation minimisation to unmix and increase the spectral resolution of hyperspectral images, was suggested in [32]. However, it employs the N-FINDR algorithm [110] to infer endmembers. The authors provided SRE results for the evaluation of algorithm performance with the following values: 12.67 dB for the USGS [48] dataset, and 14.64 dB for the ASTER dataset [70].

MCSU. Spectral-spatial-weighted multiview collaborative sparse unmixing (MCSU) [81] is a sparse regression HU algorithm that leverages spatial and spectral correlation. The main idea of the algorithm is to utilise the existing correlations between adjacent spectral bands and neighbouring pixels, which are assumed to contain highly correlated information. A hyperspectral camera may have captured the same mixture of materials in multiple groups of pixels or captured a transitioning

mix of materials that will have a robust correlation to its neighbouring pixels. The authors provide the SRE and RMSE metrics for evaluating the algorithm's performance. Three different datasets were used, with the following results: SRE = 33 for the simulated dataset; RMSE = 0.057 for the Cuprite dataset [71]; and SRE = 13.55 for the Jasper dataset [117].

SUSRLR-TV. Superpixel-based reweighted low-rank and total variation (SUSRLR-TV) [52] is a sparse unmixing algorithm that utilises the Simple Linear Iterative Clustering (SLIC) algorithm to segment hyperspectral images into homogeneous regions. It combines total variation and ADMM algorithms to calculate abundance maps. The algorithm was tested using three synthetic data cubes created with different abundances and endmembers gathered from the USGS spectral library and using the Cuprite dataset [71]. A set of different datasets was used to determine the algorithm's performance. The synthetic datasets were used to obtain accurate metrics, and the Cuprite dataset was utilised to inspect the algorithm's performance visually.

S^{2}WSU. Spectral–spatial weighted sparse unmixing [108] is a HU framework that seeks a sparse solution constrained by both spectral and spatial domains simultaneously. It implements ADMM for parameter and coefficient optimisation purposes. A dataset generated from the USGS spectral library [48] was used to determine the algorithm's performance and compare it to other popular solutions. Synthetic cubes were used to test the algorithm with SRE as the given metric, yielding results of 20.5 for the first cube and 19.6 for the second cube, with a given SNR of 30 dB.

1.2.3. Unsupervised Algorithms

Unlike other machine learning algorithms, unsupervised methods do not require trained, labelled data; the information is learned directly from the input data provided. Selected subcategories of unsupervised algorithms used for HU include Linear Mixture Models (LMM)s, Non-Negative Matrix Factorization (NMF), and Autoencoder networks.

1.2.3.1. Linear Mixture Models

Linear mixture models are regression models that simultaneously consider the variation of dependent and independent variables. The variations of both types of variables are often referred to as fixed and random effects, and because the model incorporates both of these effects, it is called a mixed model. Equation 1.3 represents the linear mixture model:

$$y = X\beta + Zu + \varepsilon \tag{1.3}$$

, where y is the outcome variable or mixture, X is the predictor multiplied by β regression coefficients, Z is the design matrix of random effects of mixed data groups, and ε is residual-like noise.

ALMM. An Augmented Linear Mixing Model (ALMM) [37] is a modified linear mixture model that utilises an endmember dictionary to determine the scaling factors and an additional dictionary to model the remaining spectral variabilities. The proposed algorithm also implements ADMM-based optimisation to solve multi-block optimisation problems [101]. In the experiment proposed by the authors, a combination of synthetic data generated from the USGS spectral library [48] and an AVIRIS-gathered hyperspectral image called Cuprite [71] was used. The result of reconstruction RMSE provided by the authors was 0.0003 for the Cuprite dataset.

1.2.3.2. Non-Negative Matrix Factorization

NMF is an algorithm group that, as the name suggests, factorises a matrix into two separate matrices with the additional assumption that all matrices have no negative elements. Because hyperspectral data cannot contain negative values, and consequently the endmember and abundance matrices are also non-negative, these algorithms are widely used in HU. The spectra at each hyperspectral image pixel are assumed to be a linear mixture of several members. Therefore, the image Z, which represents the whole hyperspectral image cube consisting of three dimensions, two spatial and one spectral, can be formulated as:

$$Z = WH + N \tag{1.4}$$

, where W is the spectral signature matrix of a size equal to the number of spectral bands (frequently annotated as λ) times the number of endmembers, H is the abundance matrix that is the size of the number of endmembers times the number of pixels, and N is the residual data size equal to the number of spectral bands times the number of pixels. HU is then performed by reversing the formula and determining the W and H matrices from the original hyperspectral image Z.

CNMF. Coupled non-negative matrix factorisation (CNMF) [104] is an algorithm used to unmix high spatial resolution multispectral data and high spectral resolution hyperspectral data simultaneously, achieving hyperspectral and multispectral data fusion. The algorithm utilisew a vertex component analysis (VCA) algorithm to calculate the initial endmember matrix from the spectral data, with a user-defined number of endmembers to identify. The authors used peak signal-to-noise ratio (PSNR) and spectral angle error (SAE) metrics to determine the performance of the unmixing algorithm. Spectral angle error is used to determine the accuracy of reconstructed spectra by calculating the angle between the estimated spectra in λ-dimensional space and the actual spectra. A smaller angle indicates a more accurate spectral reconstruction. A value of 40 dB for the PSNR metric is given for algorithm performance.

Manifold Regularised Sparse NMF. Graph-regularised $L_{1/2}$-NMF (GLNMF) [59] is a HU algorithm that takes into consideration the local geometrical structures of hyperspectral image data. Regularisation and sparsity constraints are used to detect the geometrical structure of the graph. Synthetic datasets were created using the endmembers from the USGS Spectral library [48], with the number of endmembers varying from 5 to 10. The AVIRIS Cuprite dataset [71] was used to test the accuracy of abundance estimation for different minerals. In total, the authors conducted six experiments to test various performance metrics. SAD metric results are provided by the authors: 0.019 for the synthetic dataset and 0.155 for the Jasper dataset with noisy bands, or 0.135 without noisy bands.

Subspace Clustering NMF. Subspace Clustering Constrained Sparse NMF (SC-NMF) [60] is a spectral unmixing framework that uses subspace clustering with NMF to improve the precision of the unmixing. Using a coefficient matrix derived from the aforementioned subspace clustering algorithm instead of a simple Euclidean distance creates a similarity graph. A synthetic hyperspectral image was made from the USGS Spectral library [48] to test the algorithm. Values of SAD for algorithm performance are as follows: 0.09 for the Indiana dataset and 0.089 for the Cuprite dataset [71].

CSsRS. The Correntropy-Based Spatial-Spectral Robust Unmixing Model (CSsRS) [57] is an unmixing model that utilises correntropy-based non-negative matrix factorisation, a loss function, and a sparsity penalty. The algorithm was tested using a synthetic dataset created from the USGS spectral library [48] and real datasets: Jasper [117] and Urban [116]. The authors provide values of SAD to evaluate the performance of the algorithm: 0.05 for a synthetic dataset with three endmembers, 1.4 for a synthetic dataset with eight endmembers, 0.084 for the Jasper dataset, and 0.17 for the Urban dataset.

GLNMF. General loss-based NMF (GLNMF) [77] is a HU algorithm that uses a general robust loss function instead of a least-squares loss function. The algorithm was tested using a synthetic dataset created from the USGS spectral library [48] and the Jasper dataset [117]. The values of RMSE were: 0.06 for five endmembers and 0.06 for ten endmembers.

CANMF-TV. Correntropy-based autoencoder-like non-negative matrix factorisation with total variation (CANMF-TV) [26] is an unmixing algorithm that adds a correntropy-induced metric to create the unmixing model, and a total variation regulariser is added to preserve spatial information. To test the algorithm, a synthetic dataset was created from the USGS spectral library [48] and the Cuprite [71] datasets. Values of SAD are provided by the authors: 0.13 for a synthetic dataset with a 10 dB SNR, 0.05 for a synthetic dataset with a 40 dB SNR, and 0.095 for the Cuprite dataset.

R-CoNMF. Robust collaborative non-negative matrix factorisation [54] is an unmixing algorithm that performs three steps of the HU chain. The three steps denoted by the authors are as follows:

- Estimation of the number of endmembers in the dataset being analysed.
- Identification of the endmember signatures.
- Estimation of the abundances in each pixel.

SAD values of 3.68 and 0.66 for a synthetic dataset with 20 and 80 dB SNR, respectively, are provided.

TV-RSNMF. Total variation regularised reweighted sparse NMF [36] is a blind HU algorithm based on non-negative matrix factorisation and is implemented using a reweighted sparse regulariser to promote abundance sparsity and a TV regulariser to enhance the spatial information because nearby pixels are likely to be highly correlated due to their similar chemical composition. RMSE values of 0.049 and 0.051 are provided for 10 dB SNR and 40 dB SNR with the synthetic dataset.

SGSNMF. Spatial group sparsity regularised NMF [98] is a blind unmixing method that incorporates a spatial group sparsity regulariser constraint, which takes into account pixel location (spatial data) and the fact that abundance matrices are sparse. A simulated dataset created from the USGS library [48] and a real dataset were used to test the algorithm. RMSE values of 0.02 for three endmembers and 0.06 for 15 endmembers are provided for the synthetic dataset.

EC-NTF-TV. Endmember constraint non-negative tensor factorisation via total variation for HU [97] is an algorithm that uses a proposed endmember constraint to mitigate the high correlation between spectral signatures for estimating endmembers and a total variation regularisation for exploiting the spatial correlation in calculating the abundance maps. The authors also use the augmented multiplicative algorithm to solve their abundance map objective function. The SAD and RMSE metrics were used in conjunction with synthetically generated data and the real Jasper Ridge dataset [117] to test the algorithm's performance.

For the Jasper Ridge dataset, the mean SAD score for each data class was calculated from the corresponding SAD values. Mean SAD for Jasper Ridge was 0.1248.

LIDAR-NTF. LIDAR-aided total variation regularised non-negative tensor factorisation for HU [44] proposes using a Digital Surface Model (DSM) created from LIDAR data to provide accurate elevation information about the observed scene. The provided DSM data is used in total variation regularisation as a spatial constraint, increasing the tensor decomposition accuracy, especially in areas of the hyperspectral image with a significant height difference between neighbouring pixels. Five randomly selected materials from the USGS library [48] were chosen to create a synthetic dataset. An additional Gaussian noise was added to corrupt the data. An RMSE value was calculated for synthetic images, and the results were 0.1197 with 20 dB noise and 0.1185 with 50 dB noise.

1.2.4. Neural Network Algorithms

This section describes the various HU algorithms based on neural networks, specifically deep neural networks such as autoencoder and generative models.

1.2.4.1. Autoencoder Networks

Autoencoders are a type of neural network architecture that utilises unsupervised learning. An artificial neuron bottleneck is created to form an autoencoder network that compacts the input data into a small number of features, extracting additional non-linear information from it. The model consists of the encoder part of the network, which compresses the data, and the decoder, which reconstructs the original data using the compressed features as input. This allows the networks to be trained by minimising the data reconstruction error, which measures the difference between input and reconstructed data. After the model has been trained, the compressed data can be extracted and used as an input for other algorithms. This method extracts hidden or latent features in the training data. Figure 1.8 presents a high-level diagram of an autoencoder neural network architecture, where the first part of the network serves as the encoder and the second part is the decoder, with a bottleneck layer in

between. The diagram also shows that the middle layers are smaller than the input and output layers, and the lines between nodes depict the neuron connections and weights.

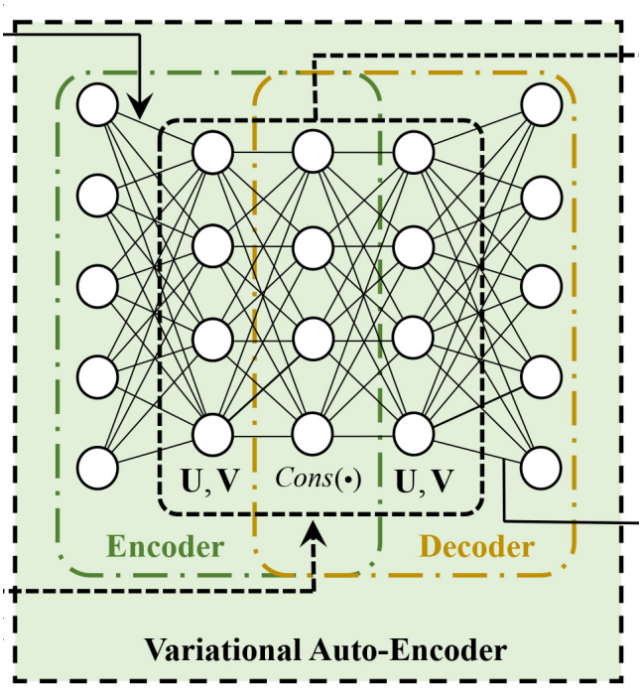


Figure 1.8 An example of a variational autoencoder network. Source: [93]

Deep Generative Unmixing. The Deep Generative Unmixing algorithm (DeepGUn) [13] 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 a more accurate abundance estimation at a minimal computational cost. 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 experiment was conducted using four synthetically created data cubes from the USGS Spectral Library [48] and hyperspectral images from the Houston [74], Samson [115], and Jasper Ridge [117] datasets. RMSE

values for these datasets are provided by the authors: 0.045 for the synthetic dataset, 0.236 for the Houston dataset [74], 0.086 for the Samson dataset [115], and 0.11 for the Jasper dataset [117].

CNNAEU. Convolutional neural network autoencoder unmixing (CNNAEU) [76] is a HU model based on autoencoder neural network architecture and the usage of convolutional layers. It is based on exploiting the spatial structures of hyperspectral images and their spectral information. It is achieved by using a convolutional neural network (CNN) to extract spatial features from the structure of hyperspectral images. The authors provide values of the mean MSE for different datasets used: 0.078 for the Samson dataset [115], 0.056 for the Urban dataset [116], 0.13 for the Houston dataset [74], and 0.10 for the Apex dataset [86].

AAS. Autoencoder network with adaptive abundance smoothing (AAS) [39] is a HU algorithm based on an autoencoder network with an adaptive spatial smoothing algorithm to improve unmixing performance. A synthetic dataset created from the USGS spectral library [48] and the Samson [115] and Jasper Ridge [117] datasets was used to carry out algorithm benchmark experiments. Endmember SAD values given for Samson and Jasper datasets are 0.11 and 0.16, respectively.

Deep HSNet. The Deep Half-Siamese network (Deep HSNet) [22] is a HU algorithm that consists of two distinct networks: an endmember-guided network and a reconstruction network. The first network maps extracted endmembers to abundances, while the reconstruction network is an autoencoder architecture that recreates hyperspectral pixels. Two different parameter networks were used in the experimentation, with a synthetic dataset created using the USGS spectral library [48] and the Urban dataset [116]. The following values of reconstruction RMSE are given: 0.12 for a synthetic dataset with 40 dB SNR and 0.087 for the Urban dataset.

DAEN. A deep autoencoder network [93] is a type of deep neural network that consists of two main parts and is used for spectral unmixing. The first part of the architecture consists of stacked autoencoders, while the second part is a variational autoencoder network.

DMBU. Deep autoencoders with multitask learning for bilinear HU [94] form an unmixing algorithm created using deep autoencoder networks and a multitask learning framework. In the proposed method, the authors train two instances of autoencoder networks together by minimising the errors between the encoder-reconstructed data and the original hyperspectral images. Using multitask learning frameworks, the authors create two models: one to obtain endmembers and abundances, and a second to estimate the bilinear components of hyperspectral data. As a result, a bilinear mixture model is created that can more accurately predict the non-linear interaction of light scattering. The authors used a variation of synthetic and real datasets to test the algorithm's accuracy and computation time. For the Jasper Ridge dataset [117], an RMSE of 0.247 was achieved, and the mean SAD over all of the different classes was 0.150.

LSTM-DNN. The LSTM-DNN-based autoencoder network for nonlinear hyperspectral image unmixing [111] is a proposed HU algorithm that utilises a long short-term memory (LSTM)-based deep learning network. The authors propose a recurrent neural network (RNN) architecture, specifically LSTM layers, and an autoencoder structure to calculate hyperspectral endmembers and abundances using the encoder, and reconstruct the hyperspectral data cube using the decoder network. The authors created synthetic datasets using USGS spectral library data and a laboratory-created mixture of data, urban datasets, and other scenes to test the algorithm's performance. Multiple metrics were calculated: average Spectral Angle Distance (SAD) (aSAD), average spectral information divergence (aSID), RMSE, and others that were not used in the experiment conducted for the Urban dataset [116]. The results for Urban dataset were as follows: aSAD – 9.2 $\pm$ 2.9, aSID ($*10^{-3}$) – 115.7 $\pm$ 84.7, RMSE ($*10^{-3}$) – 13.4 $\pm$ 3.4.

SC-CAE. A sparsity-constrained convolutional autoencoder network for hyperspectral image unmixing [112] is a convolution-based autoencoder network algorithm for HU with constrained sparsity. The authors propose an algorithm that utilises PCA on hyperspectral data, which is then fed into a convolutional autoencoder deep learning network. This network can identify abundant maps and spectral endmembers,

and reconstruct the original hyperspectral data, provided that sufficient training data and time are available. A combination of synthetic data generated using spectral information from the USGS library [48] and the Jasper Ridge dataset [117] was used to test the performance of the proposed algorithm. Mean SAD (mSAD) and mean abundance angle distance metrics (mAAD) were used to compare algorithms. For the synthetic dataset, mSAD values were 0.0135 with an SNR of 20 dB and 0.0051 with an SNR of 50 dB, and mAAD values were 0.0671 with an SNR of 20 dB and 0.0306 with an SNR of 50 dB.

1.2.5. Agricultural Hyperspectral Data Unmixing

A comprehensive and extensive review paper on the usage and unmixing of hyperspectral data in agriculture was written by Geurri et al. [30]. The authors explore the use cases of hyperspectral data in agriculture and the available algorithms used to unmix this hyperspectral data, examining the following types of algorithms: Autoencoder denoising [21], Convolutional Neural Networks [66], Recurrent Neural Networks for classification [114], Deep Belief Networks [58], Generative Adversarial Networks for denoising [107] and super-resolution [88], Transfer Learning for classification tasks [56], Semi-Supervised Learning classification [113], and Unsupervised Learning classification [66]. From their paper, it can be concluded that the most popular recent unmixing methods all fall within the domain of deep learning algorithms.

Annam and Singla [3] used supervised and unsupervised machine learning methods to detect heavy metals (arsenic (As), cadmium (Cd), and lead (Pb)) in soil from hyperspectral data, with an unsupervised k-means algorithm achieving the best resulting accuracy of around 98%.

1.2.6. U-Net Image Classification Model

The original U-Net model created by Ronneberger et al. [85] was used for biomedical image segmentation. Our previous work demonstrates that autoencoder networks are a widely used deep learning method in HU. 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. Figure 1.9 shows the original U-Net model

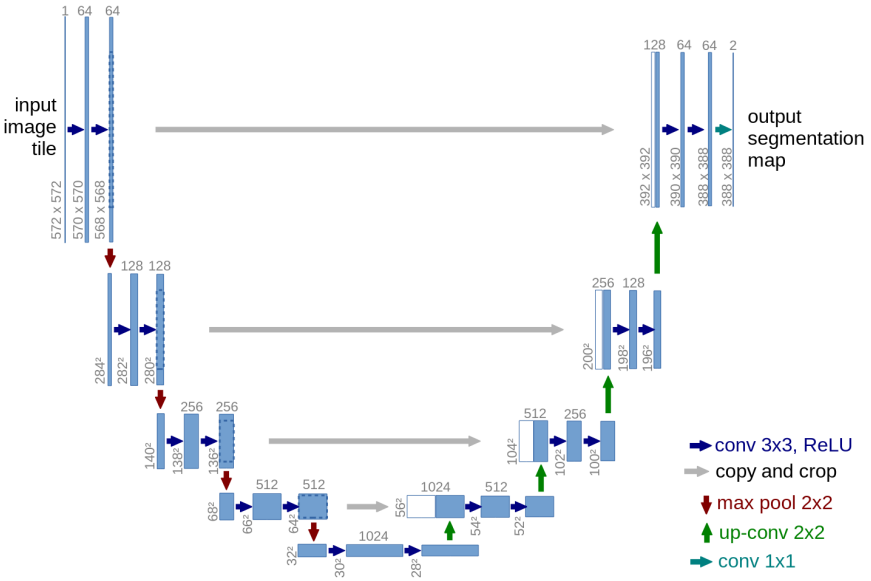


Figure 1.9 U-Net model architecture and layers. Source: [85].

architecture diagram.

1.2.7. Transformer-Based Hyperspectral Unmixing Model

This is the primary model used in creating the proposed model as the baseline was the unmixing neural network, which utilises a visual transformer for endmember extraction, developed by Ghosh et al. [28]. Their model code is implemented in Python using the PyTorch neural network creation library and is freely available at `https://github.com/preetam22n/DeepTrans-HSU`. The implemented code consists of:

- neural network implementation using PyTorch;
- various hyperspectral dataset reading and collection parameters for each of the datasets;
- resulting data visualisation wrappers;
- the implementation of a few loss functions used in model training.

1.2.8. Hyperspectral Datasets

This section describes hyperspectral datasets used in experiments reviewed in other publications, or those that are openly available for use. These datasets were used in experimentation and as a comparison to the new agricultural dataset that was created:

- A *synthetic data* cube created by artificially mixing different amounts of pure spectra taken from the USGS spectral library. To make the synthetic datasets, version 7 of the USGS spectral library (splib07a) [48] was utilised, which comprises over 2000 distinct spectral endmembers spanning a wavelength range of 0.2 to 200 micrometres. Several datasets were generated using the spectral data from this library to conduct the benchmark experiments.

- A hyperspectral dataset created by the authors of [105] in a laboratory setting that contains hyperspectral images and spectral ground truths. The dataset is split into three scenes, each containing spectra of pure-coloured materials mixed in various proportions to create mixed spectra. The difference between this dataset and the synthetic data created using the USGS library is that the mixtures are physically produced and represent a more realistic mixing model. In contrast, library members are usually mixed linearly. The first and third scenes utilise cyan, magenta, and yellow dyes and mixtures, while the second scene employs red, green, blue, and white dyes.

- *IEEE GRSS 2018 data fusion contest hyperspectral dataset* [80]. The hyperspectral dataset was gathered flying over the University of Houston and consists of an area measuring 1202 by 4172 pixels, each containing 48 spectral bands with wavelengths from 317 nm to 1047 nm. The RGB reconstruction of the data is shown in Figure 1.10.

- *Jasper Ridge* [75] – a 512 x 614-pixel-sized hyperspectral image with 224 spectral bands with four endmember classes: '#1 Road', '#2 Soil', '#3 Water' and '#4 Tree'. Other publications that use this dataset in experimentation include: [92], [97], and [13].

Figure 1.10 2018 IEEE GRSS data fusion hyperspectral data, RGB reconstruction. Source: [80]

- *Cuprite* [75]. A hyperspectral image 307 x 307 pixels with 210 spectral bands, with a ground truth consisting of 4, 5, or 6 classes – [98], [26].
- *Urban* [75]. A region of 250 x 190 pixels with 224 spectral bands and a collection of 14 different minerals (endmembers). Endmember spectra are depicted in Figure 1.11. Papers that use this dataset in experimentation include: [34], [111].
- *DC Mall* [75] [68]. A scanned area of part of Washington DC with a size of 1208 x 307 pixels and 191 spectral bands. Ground truth was created for classification with the following classes: Roofs, Streets, Paths, Grass, Trees, Water, and Shadows. This dataset was used in the experimentation presented in this paper and the Ghosh et al. [28] paper.

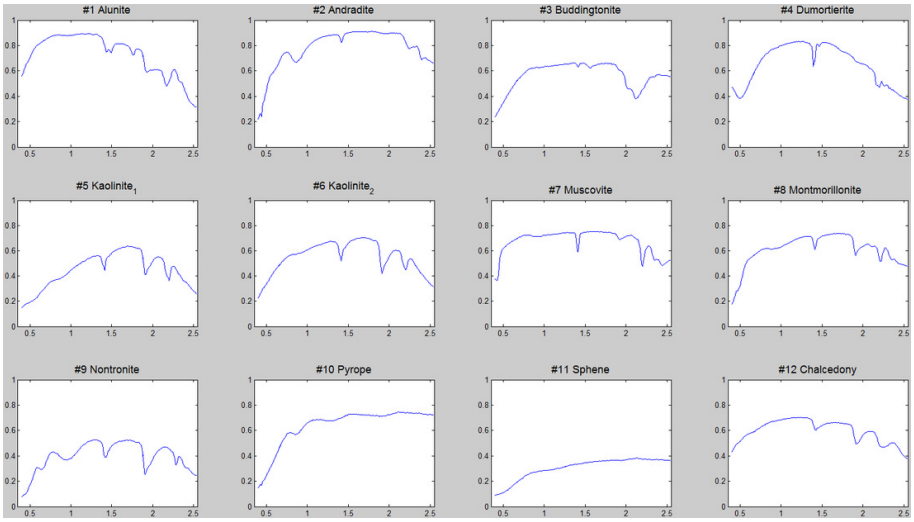


Figure 1.11 Cuprite dataset endmember spectra. The graphs show the reflectance values (y-axis) for different materials in specific wavelengths (x-axis) in micrometres. Source: [75]

1.3. Conclusions of the Chapter

- HSI is an increasingly popular remote sensing field that is used in different applications as a non-destructive data collection method.
- HSI is a vast field of science, and numerous algorithms are consequently employed for the analysis, preprocessing, classification, and clustering of hyperspectral data.
- With a large variety of algorithms available, different hyperspectral applications require different methods to be used. The newer, deep learning-based LSTM-DNN algorithm achieved a lower RMSE value on the Urban dataset compared to other methods tested using this dataset.
- The review of existing HU shows the absence of a standardised testing methodology for these HU algorithms, with many different datasets and metrics used across all of the algorithms. This in turn makes the comparison of these methods inaccurate.

2. HYPERSPECTRAL UNMIXING METHODS AND RESEARCH METHODOLOGY

This chapter describes the hyperspectral datasets that are commonly used in related works for HU, the hyperspectral data collection process, the HU dataset that was created using this process, the various metrics used to evaluate the performance of HU algorithms, and the proposed U-Net-based HU neural network model. The chapter is based on two published papers [A.1] [B.1] and one that has been submitted for publication [A.2].

2.1. Metrics Used

This section describes the various metrics used in evaluating the performance of the HU algorithm and its training accuracies.

2.1.1. General Metrics

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

- RMSE (Equation 2.1) shows the difference between the predicted spectra and the ground truth. Different authors have employed several variations of RMSE to assess different aspects of the created algorithms; these include 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 those that are predicted. This metric was selected due to its widespread use in papers on HU algorithms and its versatility in various error computations.

- SRE (Equation 2.2) is used to determine the quality of the spectral mixture reconstruction generated by the algorithms. A higher SRE value means better reconstruction quality. This metric was selected as a modification of the RE metric specifically used for signal data.

Metrics are calculated using the following formulas:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{2.1}$$

, 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.

$$SRE = 10log_{10}\left(\frac{E[\|x\|_2^2]}{E[\|x - \hat{x}\|_2^2]}\right), \tag{2.2}$$

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

2.1.2. Neural Network Algorithm Training And Evaluation Metrics

The metrics and losses in this section were chosen to train and evaluate the proposed neural network model based on the U-Net architecture.

Reconstruction Error (RE) (Equation 2.3) measures the average difference between model-generated data and actual ground truth data:

$$L_{RE}(I, \widehat{I}) = \frac{1}{W \cdot H}\sum_{i=1}^{H}\sum_{j=1}^{W}(\widehat{I}_{ij} - I_{ij})^2 \tag{2.3}$$

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

Spectral Angle Distance (SAD) (Equation 2.4) measures the angles between two vectors in a multidimensional space:

$$L_{SAD}(I, \widehat{I}) = \frac{1}{R}\sum_{i=1}^{R}\arccos\left(\frac{\left\langle I_i, \widehat{I}_i\right\rangle}{\|I_i\|_2\|\widehat{I}_i\|_2}\right) \tag{2.4}$$

, where R is the number of pixels, I_i is actual data, and $\widehat{I}_i$ is predicted data. When used on hyperspectral data, the hyperspectral data cube's

two spatial dimensions are combined into a single dimension of size R, and the angle is calculated for each pixel separately.

Cosine similarity (Equation 2.5) calculates the dot product of the vectors divided by the product of their lengths:

$$\cos(\theta) = \frac{I \cdot \widehat{I}}{\|I\| \left\|\widehat{I}\right\|} \tag{2.5}$$

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

2.2. Hyperspectral Data Collection Methodology

This section describes the methodology used to collect agricultural hyperspectral data. The hyperspectral data gathered was used to create a hyperspectral dataset and to test HU algorithms.

2.2.1. Hyperspectral Data Gathering Methodology Using UAVs

Raw hyperspectral data was gathered using an *Aurelia* X6 drone [1] with a *SPECIM* hyperspectral pushbroom camera [90] flying over a blueberry field. The raw hyperspectral data is gathered by the sensor and written into a proprietary format file created by the camera manufacturer. The raw hyperspectral data must be processed by the software provided by the camera manufacturer, producing a hyperspectral data cube in an ENVI raster format [89] that is readable by other software. The data from these specific blueberry fields was selected because they represent the highest quality data available from all the gathered hyperspectral data, and also because they have the lowest number of shadows in the hyperspectral data cubes.

The pushbroom hyperspectral camera records data from all 224 spectral bands, each 1024 pixels wide. In turn, the final hyperspectral cube size 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 to maintain a square pixel resolution was calculated. The final pixel size of UAV hyperspectral cubes was 5 x 5 centimetres. To maintain

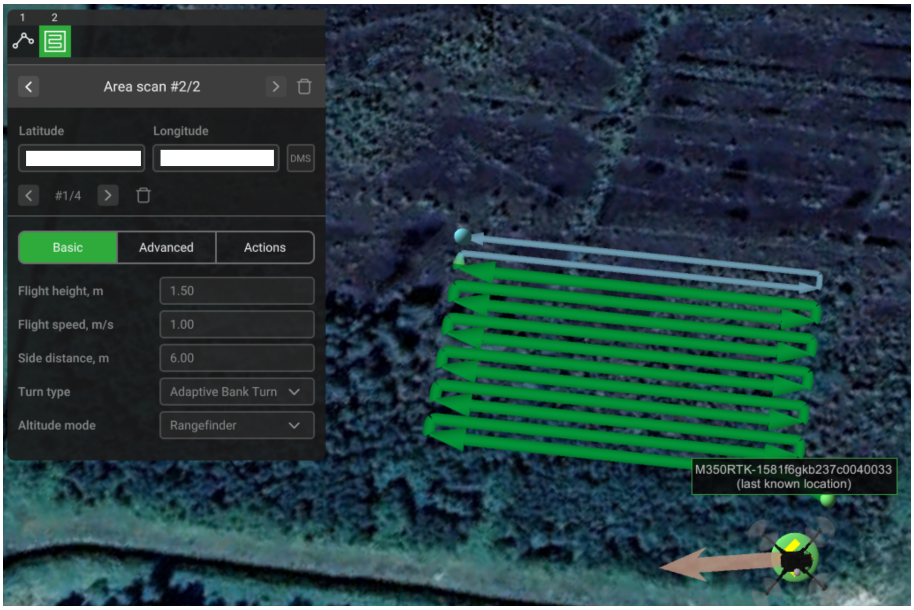


Figure 2.1 Example flight lines generated by the UAV-compatible software in a selected area using the given flight parameters.

the accuracy of the data, recording must be performed along straight flight paths, known as flight lines (example flight lines are shown in Figure 2.1). Each flight line generates a separate hyperspectral data cube of size $1024 \times x \times 224$, where x depends on the line length and camera recording speed. An exposure time of 6 ms was selected and the camera frames per second (fps, or lines per second in the case of a pushbroom camera) was set to 100.

From the multiple hyperspectral data cubes gathered during the UAV missions, three were selected as the base of the unmixing dataset. These data cubes form the basis of the newly created dataset for HU, which was created as part of the research on HU algorithms. The blueberry field HU dataset collected using a UAV was called Blueberry Field HU Dataset (BFHUD). All three BFHUD data cubes share the same set of endmembers. Still, the data was collected in the same blueberry field by gathering data over multiple days over different rows of blueberry bushes, with no overlap between the data. Three cubes were used to increase data variety and, in turn, check algorithm robustness to changes in field data. The three data cubes have the following parameters:

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

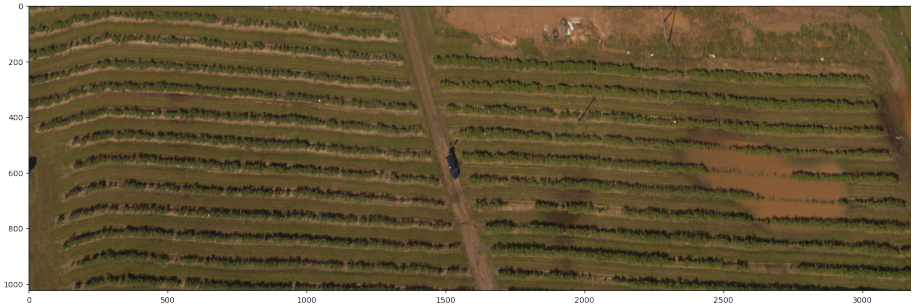
BFHUD cube RGB representations created by data integration over CIE 1931 XYZ colour matching functions and conversion from XYZ to RGB are given below in Figure 2.2. Code listing of the function used to convert hyperspectral data to XYZ colour space is provided in appendix C, and the formula to convert from XYZ to RGB values is shown in equation 2.6 [65]:

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 3.2404542 & -1.5371385 & -0.4985314 \\ -0.9692660 & 1.8760108 & 0.0415560 \\ 0.0556434 & -0.2040259 & 1.0572252 \end{vmatrix} \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \tag{2.6}$$

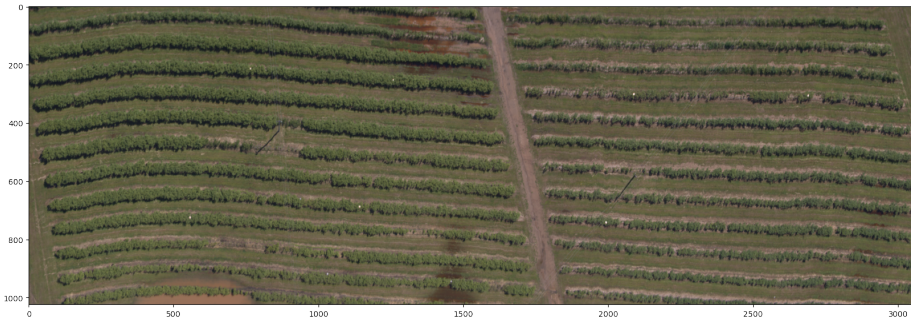
, where R, G, and B are the colour channel vectors that are used by most visible light cameras, and X, Y, and Z are the colour channel vectors calculated via integration from hyperspectral data.

2.2.2. Calibration

To maintain the consistency and comparability of the data between flights, a set of calibration carpets was deployed in the field, with laboratory-calibrated reflectance values of 5%, 10%, and 40%. A data cube was also created 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 is described in an article by Burger and Geladi [16], as outlined in Section 1.1.4.



(a)



(b)

(c)

Figure 2.2 RGB representation of the BFHUD cubes. BFHUD Cube 1 RGB representation is shown in (a), BFHUD Cube 2 RGB representation is shown in (b), and BFHUD Cube 3 RGB representation is shown in (c).

2.3. New Unmixing Dataset

This section describes the methodology used to create the HU dataset from the raw hyperspectral data gathered. The decision to develop BFHUD was made due to the lack of freely available high-quality HU datasets gathered by a UAV. Data collected in this way shows a significant increase in ground sampling distance compared to the open satellite-gathered datasets used, such as Cuprite [75], DC Mall [68], or Urban [75]. Ground sampling distance is defined as the distance between the pixel centres of an image captured from the air. For example, the Urban dataset has a ground sampling distance of 2 meters, while the UAV-gathered data has a ground sampling distance of 3 centimetres.

2.3.1. Data Classification

The most significant problem in creating the dataset was the lack of accurate ground truth values, as it was unknown which pixel belonged to which class. Manually labelling more than three million pixels individually is not feasible. To overcome this problem, approximations of endmembers and classes were calculated. A special unsupervised method called the Vertex Component Analysis (VCA) [72] algorithm was used to extract the endmembers. This algorithm attempts to extract possible endmembers from a hyperspectral image and the specified number of endmembers to be extracted. This method was chosen due to the fact that it offers best performance (a combination of computation speed and extraction accuracy) in extracting endmembers compared to more commonly used methods such as PCA [63], Independent Component Analysis (ICA) [40], and the Automatic Target Generation Process (ATGP) [19]. An endmember extraction experiment was conducted using the steps outlined in the diagram in Figure 2.3 to determine the most accurate endmember representation for this dataset. The endmember extraction experiment was performed in the following stages:

- A range of numbers from 3 to 50 was selected;
- For each number, the VCA algorithm generated a set of endmembers from the hyperspectral dataset. The output of the VCA algorithm is a collection of spectral endmembers that are then used to

represent the possible ground truth classes of the hyperspectral image.

- To select the best class for each of the pixels in the hyperspectral image, the differences between each endmember generated by the VCA algorithms and the hyperspectral image pixels were computed using RMSE. This example creates a matrix of values of an area 3177 x 1024 pixels x the number of endmembers when used on BFHUD Cube 1.

- For each pixel, the endmember with the lowest RMSE value was set as the new ground truth class. For example, if endmember 1 represents the 'soil' class, each pixel where RMSE between the pixel and the endmember 1 was the lowest will be classified as a pixel with the class 'soil'.

- To select the optimal number of endmembers to generate, which is the lowest number possible that sufficiently explains the data variation of the hyperspectral image, an average of all the smallest RMSE values for all of the pixels in the hyperspectral image was calculated.

- For each number of endmembers from 3 to 50, the average lowest RMSE value was calculated and plotted. The generated graph shows the plotted values.

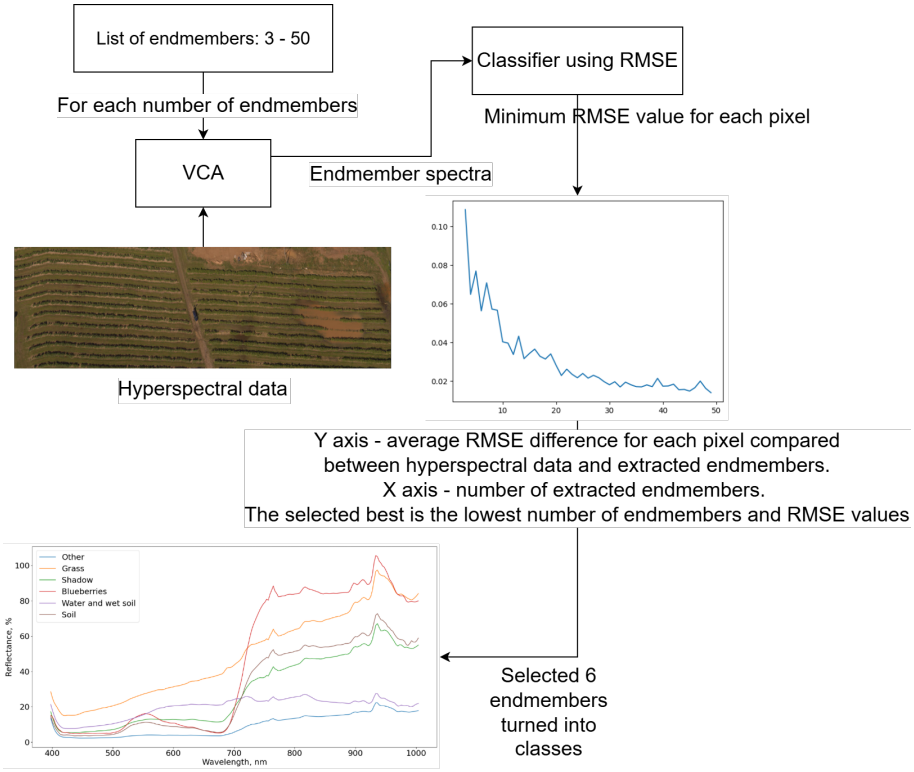


Figure 2.3 Diagram of the extraction of the endmembers from the hyperspectral image using the VCA algorithm for BFHUD cube classification.

The result from the endmember extraction experiment is the number of endmembers and how accurately they can describe the original hyperspectral image. A lower total RMSE indicates a better representation of the original data. At the same time, a higher number of endmembers increases the complexity of the dataset, and may result in classes that represent a minimal amount of actual data. The most optimal combination is one that combines the lowest total RMSE with the smallest number of endmembers. It was concluded that the best number of endmembers was 6 or 12. Both configurations were initially used. Still, after analysing the number of pixels in each endmember, it was determined that when using 12 endmembers, some classes have a tiny number of samples (<0.1% of pixels), and this would significantly increase unmixing difficulty. Therefore, six extracted endmembers were used for further research.

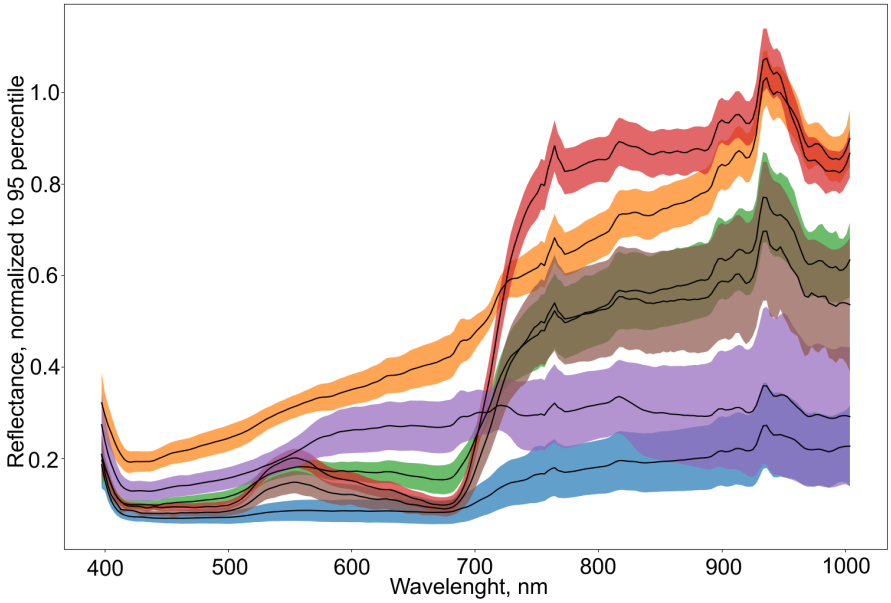


Figure 2.4 Six extracted endmembers represented by different colours and the standard deviations in each spectral band in the BFHUD.

The six extracted endmembers were selected as the ground truth classes for the BFHUD. The class was assigned to each pixel using the lowest RMSE value between the pixel spectrum and each endmember. With the image classified, all pixel data was gathered for each class, producing a collection of similar pixels. To remove outliers from the dataset, each pixel with a spectral variation in any bands more significant than 2 σ was replaced by a random neighbouring pixel that satisfied this spectral variation rule. A new filtered hyperspectral image was generated with the most significant outliers removed. A diagram showing the endmembers and their standard deviations in each spectral band is presented in Figure 2.4.

Raw class data is represented in Figure 2.5. Classes represent blueberry crops, bare soil, grass, data in shadow, water, and any other data combined into these classes.

2.3.2. Dataset Publishing

Complete BFHUD and classified data are published as open access data for use in other experiments and HU and classification tasks.

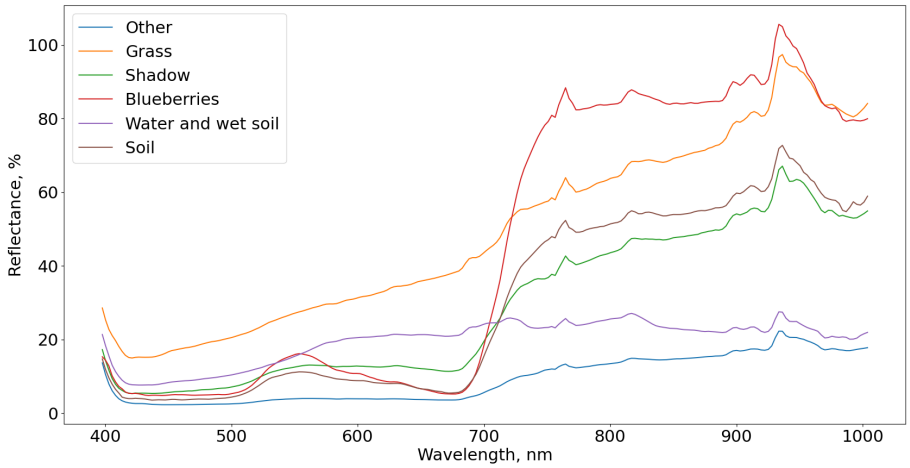


Figure 2.5 Averages of extracted endmembers used as the ground truth for the BFHUD for each of the six classes.

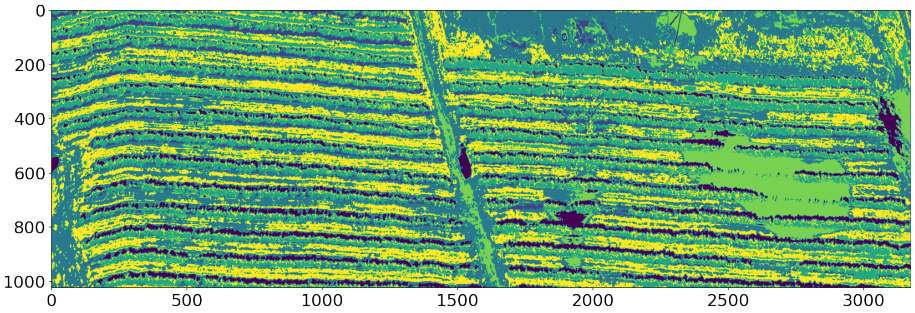


Figure 2.6 Class distribution in BFHUD Cube 1. Color class representation: Yellow – bare soil; Green – blueberries; Blue – grass; Dark blue – shadowed data; Light green – water and wet soil; Black – other data.

Data is published on the Zenodo platform with open access: `https://doi.org/10.5281/zenodo.13856357`

2.4. Benchmarking Methodology

This section describes the HU benchmarking methodology created to standardise HU algorithm testing, as there was no single way of testing these algorithms identified during the literature review. To test the different aspects of HU algorithms, the benchmark is divided into four main sections:

1. Endmember robustness. This tests the algorithm's ability to be generalised and its overall performance when the number of endmember inputs is changed. This type of test enables the algorithm to assess its ability to identify endmembers and reconstruct hyperspectral images, depending on the complexity of the scene. Due to the changing number of endmembers, a new synthetic dataset was created using a combination of IEEE GRSS [80] data as a basis, and the USGS spectral library [48] was used.

2. Robustness to noise. This experiment assesses the algorithm's ability to accurately unmix hyperspectral image spectra when varying levels of artificial noise are added to the image. This experiment tests algorithms with varying amounts of white noise and a noise profile created from a real-world scenario. The dataset created to test endmember robustness was used as a base hyperspectral dataset, and an artificial noise layer was added to it.

3. Impact of differences in input image sizes. By varying the dimensions of hyperspectral images, the amount of spatial and spectral information changes, which in turn affects the overall performance of the algorithms. This also enables the determination of the optimal image size for achieving the most accurate unmixing result and optimal performance combination. It also displays the data necessary for algorithms to achieve their optimal accuracy. The same endmember robustness dataset was used and then downscaled using the methodology described below to create the different spatial size hyperspectral images.

2.4.1. Endmember Robustness Experiment Schema

Endmember robustness testing is performed by creating a group of artificially generated datasets according to a set of rules:

- Datasets D_x (where x is the set number of endmembers) are created by selecting the endmembers from USGS spectral library data.

- The number of endmembers x selected is from 3 to 10 endmembers with a step of 1, from 10 to 30 endmembers with a step of 5, and from 30 to 100 endmembers with a step of 10.

- For each D_x, one abundance matrix A_{equal} is created by using an equal abundance of each endmember x.
- For each D_x, ten abundance matrices A_y are created by randomly generating endmember abundances y using a uniform distribution. y is normalized to conform to sum-to-one constraint (equation (2.7)).
- An artificial hyperspectral image I_i (an example RGB representation of such an image is shown in 2.7) with an area of 150 by 100 pixels is generated using the abundance matrix A_y and endmembers x. The image size was selected to represent a realistic hyperspectral image while keeping it low so as to reduce computation resource usage.

The sum-to-one constraint formula is given below:

$$\hat{y_i} = \frac{y_i}{\sum_{i=1}^{x} y_i} \tag{2.7}$$

, where $\hat{y_i}$ is the normalized abundance values, y_i are the model generated abundance values for each endmember, and x is the number of endmembers.

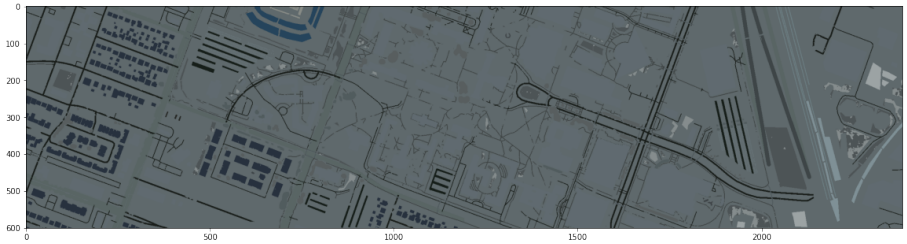


Figure 2.7 An RGB representation of an artificial hyperspectral image.

2.4.2. Robustness to Noise Experiment Schema

A collection of artificially generated hyperspectral images was created to test the algorithm's robustness to noise. Then, a different amount of noise was added to the images according to the following set rules:

- A collection of 4 different datasets was created with different endmembers using the same methodology as in the endmember robustness experiment in subsection 2.4.1.

- For each of the four datasets, a different amount of artificial noise was added.
- The created noise was measured in SNR dB, in which a lower number means a higher amount of white noise.
- A random noise with a mean value of 0 was generated with the desired SNR dB values of 20, 25, 30, 35, 40, 45, and 50.
- This noise was then applied to each of the four datasets to create noisy images.

In addition to the random white noise generated, a set of noise parameters was extracted from a HSI camera used for research in an uncontrolled field environment. The camera was a BaySpec OCI-F Hyperspectral Imager operating in the visible and near-infrared electromagnetic range [7]. A Pearson correlation coefficient was calculated to measure the noise generated by the camera at each wavelength. Each neighbouring wavelength was extracted from a hyperspectral image, and the correlation between the values of these wavelengths across the entire image was calculated. Figure 2.8 (orange line) shows the correlation coefficient at a wavelength index of x and $x - 1$. The exact Pearson correlation was calculated for one of the synthetically generated hyperspectral images used in this experiment, and the results are shown in Figure 2.8 (blue line). The Pearson correlation between neighbouring bands in the same image was calculated using formula (1.1) where r is the correlation value, x is the first set of values (in this case, values of specific wavelength), and y is the second set of values (values of neighbouring wavelength). The calculation was performed by taking a pair of values from the same pixel i, calculating the difference from the average value of each set ($\overline{x}$ and $\overline{y}$), multiplying them, taking the sum, and dividing by the root of their squared product sum.

A set of artificial noise parameters was found using a gradient descent minimisation algorithm. The band-to-band Pearson correlation coefficients closely resembled a real-world camera noise specification when applied to the synthetically generated hyperspectral image. In this instance, a multivariate optimisation algorithm was used to calculate the number of wavelengths, given a certain number of different variables. Specifically, an evolutionary algorithm from the *scipy* Python

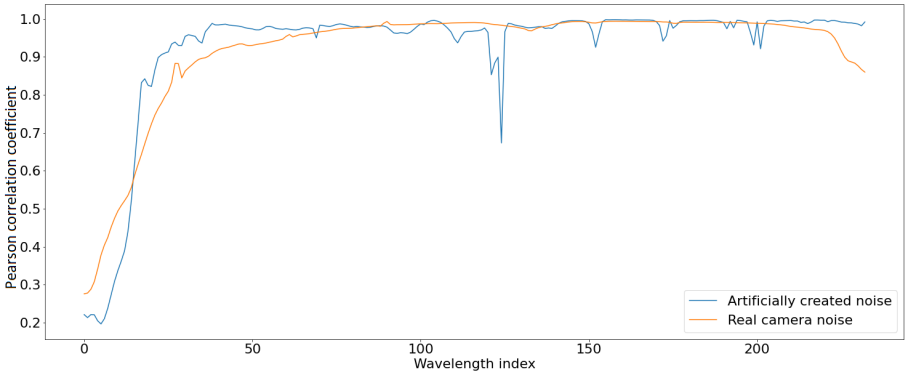


Figure 2.8 A comparison between the Pearson correlations of real and artificially created noise profiles.

library called differential evolution was used to minimise the difference between actual correlation and artificial correlation. Figure 2.8 shows the noise specification results. This noise profile was then used to create a more true-to-real-world version of camera noise to test how algorithms perform in this scenario.

2.4.3. Image Size Difference Experiment Methodology

Algorithm performance testing according to different image sizes was conducted via the following steps:

- A synthetically generated hyperspectral image dataset with different numbers of endmembers was created using the exact methodology of the endmember robustness experiment described in subsection 2.4.1.
- These datasets were then downscaled using mean values in an area of 2 x 2 pixels to a single value and 3 x 3 pixels to a single value, with the step size equal to the filter area. In other words, the sliding window was used without any overlapping. In turn, this created images that were 4 and 9 times smaller.
- RMSE and SRE metrics were then calculated on these three collections of datasets to compare the results.

2.5. Proposed U-Net Based Neural Network Model

With research conducted on existing HU algorithms and using the model mentioned in Chapter 1.2.7 as the baseline for experimentation, a new model was proposed for HU, with its architecture based on the U-Net deep convolutional autoencoder [85]. The newly created HU model was called Hyperspectral Unmixing U-NET (HUNET).

The transformer model created by Ghosh et al. [28] had some downsides that were discovered during testing:

- The original model was used only on square images.
- The entire image was used as an input to the model, which determined the size of the model's layers. A large input dataset must be compressed, or the latent space of the transformer must be reduced due to reaching the limits of available graphics card memory.
- A set of manually tuned hyperparameters was provided for each dataset tested in the paper. A new set of parameters must be generated to use the latest datasets.

The first problem was solved directly in the original code by allowing it to use rectangular images, not just squares. This was done only on the data reading part of the code, without any changes to the model itself. Optimisations and model improvements may be possible when using rectangular data as an input to the transformer patching algorithm used in the model. Running this model with BFHUD, a few iterations of training were conducted to obtain a set of parameters that yielded valid results without exceeding graphics card memory limits or encountering other errors (e.g., the input data shape must be divisible by the patch size provided). A manual solution to the second and third problems was found. The solution to the second problem of excessive graphics card memory usage was to reduce the latent space parameter by half compared to the value used by the authors for the DC Mall dataset. To solve the third problem, the manual tuning of hyperparameters was performed until the algorithm successfully extracted the endmembers from the BFHUD.

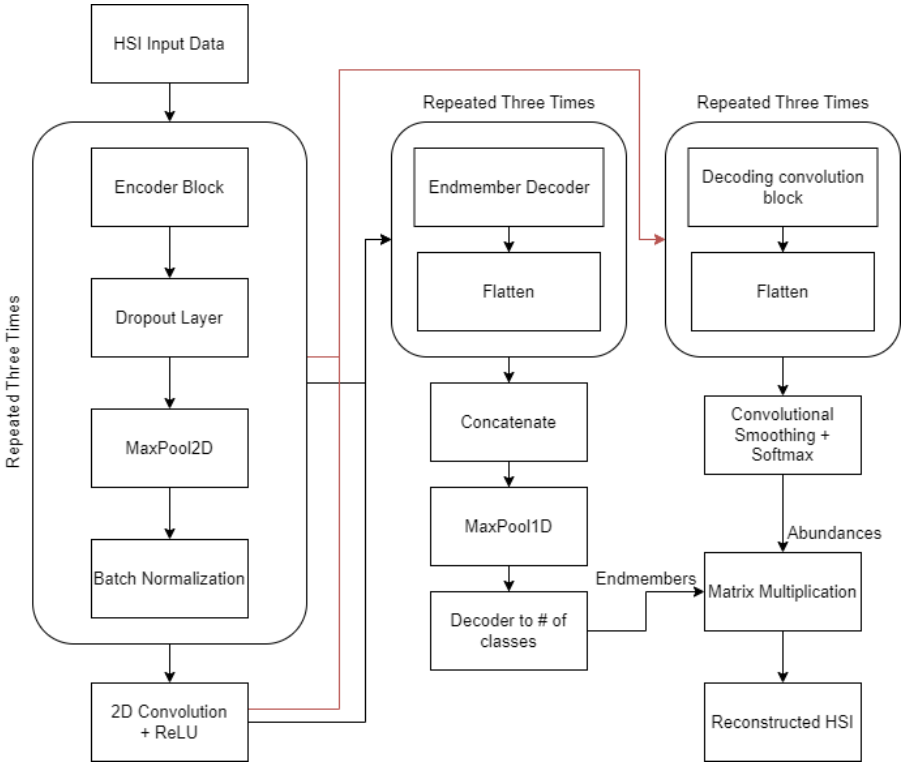


Figure 2.9 HUNET model architecture.

To solve all of these problems, a new HU model is proposed that combines ideas from the U-Net model architecture and the transformer-based unmixing model created by Ghosh et al. [28]. The original U-Net was used for segmentation, and modifications to the architecture were needed for endmember and abundance matrix extraction and image reconstruction. Figure 2.9 shows the new HUNET model architecture.

The main changes from the U-Net and transformer models were:

- Splitting the hyperspectral image into smaller, same-size photos to reduce the overall size of the model, enabling the use of augmentations (e.g., mirroring and rotations) on the input data and training the model by selecting these image patches in random order.
- Splitting the compressed data into endmember and abundance extraction sub-networks.
- The model architecture is dependent on fewer hyperparameters. The new model does not utilise the patch size and latent space

hyperparameters, as these parameters are already incorporated into the transformer layers.

- The addition of cosine similarity loss was used to encourage the model to extract endmembers that differ from each other. The cosine similarity loss computation function is provided in Appendix E. This function splits the predicted endmember vectors into two matrices so that the *PyTorch cosine_similarity* function can compute the similarity between all endmember pairs. The calculated result is then positive with the absolute function, and the average value is calculated. The resulting value is scaled across a range from 0 to 1 in order to keep all losses within a similar range and reduce the model's sensitivity to this loss during training.

The final HUNET model architecture is constructed from the following modules, which are depicted in the model architecture graph in Figure 2.9:

- the encoding convolution block, which is a collection of the Group normalisation, 2D convolution, and ReLU layers that is the base of the encoder network;
- the encoding block, which combines two encoding convolution blocks and adds a dropout layer after;
- the decoding convolution block, which combines two encoding convolution blocks with a linear interpolation upscaling algorithm to upscale the image two times;
- an endmember decoder, which is a 2D convolution layer combined with the LeakyReLU activation function that upscales the spectral dimension;
- 2D MaxPool layers, which are used to downscale the data 2 times in two dimensions and are used together with the decoder layers;
- a linear decoder with Sigmoid, which is used to reconstruct the convolution channels to spectral channels for the extraction of endmembers;
- a 2D convolution smoothing layer with Softmax, which is used to extract the abundance matrix from the decoded data.

2.6. Conclusions of the Chapter

- Hyperspectral data collection is a challenging task that necessitates data processing and calibration computations to render the data usable in research.
- A new HU dataset called BFHUD was created using the VCA algorithm to compute the ground truth endmember values and classify hyperspectral images.
- A new HU algorithm called HUNET based on a U-Net neural network architecture was created.
- The newly proposed HUNET algorithm implemented improvements on the transformer-based neural network HU model for better performance in HU.

3. EXPERIMENTS AND BENCHMARKING

This chapter describes the HU experiments conducted, the results of hyperspectral data synthesis, the proposed benchmark for testing HU algorithm performance, the algorithms tested using the created benchmark, and the performance of the HUNET model compared to the transformer-based neural network model. The chapter is based on results published in two other papers [A.1] [B.1] and one paper pending publication [A.2].

3.1. Experimentation Plan

Some observations and hypotheses can be formed from the research conducted on HU algorithms presented in Chapter 1.2:

- Hyperspectral data bands close to each other have a high correlation with each other and can be approximately synthetically generated from a multispectral dataset.
- HU algorithm performance depends on the number of endmembers, because a larger number of endmembers indicates a smaller amount of spectral data for each endmember.
- The amount of data in hyperspectral datasets for each endmember impacts the accuracy of the HU algorithm. Factors of four and nine were used to scale the hyperspectral dataset inputs for testing.
- Various HU algorithms perform worse when the hyperspectral data used is noisy.
- A specialised HU algorithm for agricultural hyperspectral data would achieve a higher level of accuracy than a generalised, state-of-the-art, transformer-based HU method.

The experimentation plan was created based on the hypotheses above to test aspects of various HU algorithms. The HU benchmark (explained in Chapter 2.4) was created to test the endmember, the amount of data for each endmember, and the robustness of each algorithm to noise.

3.2. Hyperspectral Data Synthesis

Expanding on the second type of data synthesis approach described in Chapter 1.1.6, a GAN network was proposed by Audbert et al. [4]. Their proposed model architecture and insights were used to create a generative model to synthesise hyperspectral data. The model generated a hyperspectral image of 224 spectral bands in the visible and near-infrared ranges from an RGB or multispectral dataset. The GAN model proposed by Audbert et al. [4] was simplified and adapted, with the following changes:

- The usage of a trainable discriminator was removed to simplify the model, and this was changed into the reconstruction loss. This was done because the model does not need to generate hyperspectral data from scratch, it only requires training on existing RGB or multispectral information.
- Class information was removed from the generative model; the newly created model is fully unsupervised. The developed model's primary use case would be to augment the existing hyperspectral data of a given area with newly generated hyperspectral data from other sources, such as satellites. The model created by Audbert et al. [4] is trained to create specific-class hyperspectral data given random noise and a class label.

For algorithm creation and testing, three hyperspectral images were selected that were gathered using a UAV over an area with diverse features. Using CIE 1931 colour space integration curves, RGB images were generated from the hyperspectral data. RGB images were generated to simulate data gathered by RGB cameras more accurately, rather than using single narrow bands for each colour. An example of an RGB image is shown in Figure 3.1: of one of the test data cubes used in creating the generative model. The function for RGB image creation from hyperspectral data is provided in Appendix C.

3.2.1. Generative Synthesis Model

A new generative model architecture was developed and refined through multiple iterations to achieve improved results. The experiment was conducted in the following steps:

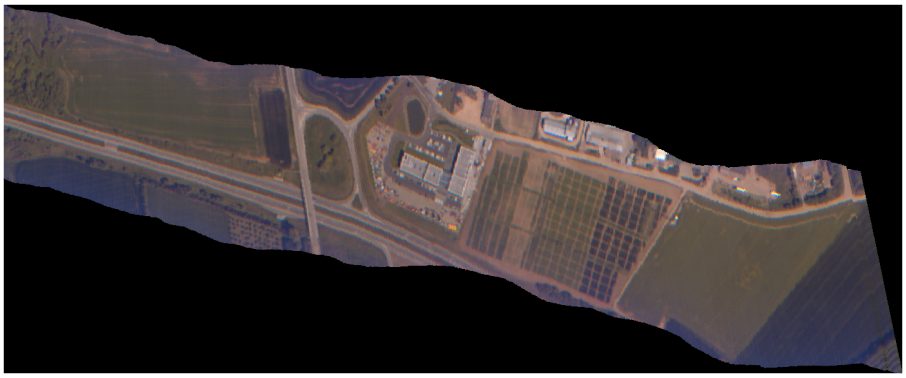


Figure 3.1 Example of an RGB image generated from one of the hyperspectral data cubes used in the creation of the generative model.

1. Three different hyperspectral data cubes were used as training and testing datasets to create the new hyperspectral data generative model. The generated dataset measured 1024 by 9230 pixels, and each pixel contained 224 spectral bands.

2. RGB images were generated from these hyperspectral data cubes using the Python function provided in Appendix C, and an additional spectral band was included based on the Sentinel-2 near-infrared band wavelength (842 nm).

3. Hyperspectral and RGB data were used to train the generator model using small batches and 10 epochs of training with the RMSprop optimiser, as this was used by Audbert et al. [4], and RMSE was employed as the loss function for checking the accuracy of the spectral data generated.

4. Following the example of the model created by Audbert et al. [4], the input data for the new generative model was formed of individual pixels, and the output was a generated hyperspectral pixel.

Figure 3.2 shows the model creation diagram.

Figure 3.3 presents the newly created generative model architecture for this experiment.

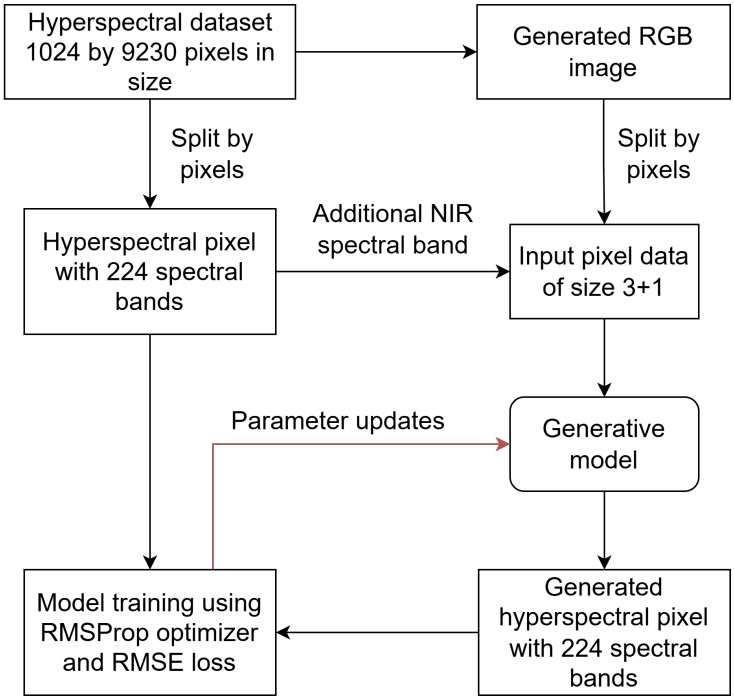


Figure 3.2 Generative model training and testing using hyperspectral datasets and multispectral images generated from them.

3.2.2. Generative Model Results

The initial purpose of creating a model capable of generating hyperspectral data was to determine whether it is possible to accurately recreate hyperspectral data from simple RGB images. The developed model was split into two variants, one using NIR data and the other using only RGB data. To compare both models and to better understand the differences in using only RGB data and adding near-infrared spectral data, the RMSE metric was calculated between the data generated by the models and the original hyperspectral dataset. By calculating the RMSE metric for each of the spectral bands, it is possible to see which parts of the hyperspectral data can be reconstructed more accurately. A diagram showing the RMSE for each band, comparing the performance of both models, is shown in Figure 3.4. From the RMSE values, it is evident that to accurately predict near-infrared data, at least a single band of this data is crucial.

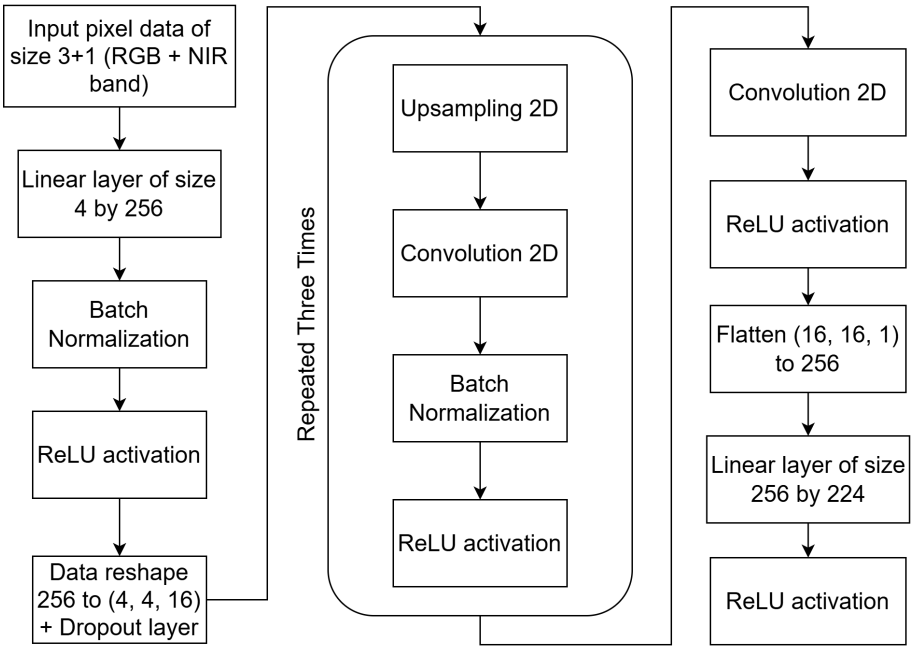


Figure 3.3 The newly created hyperspectral data generation model architecture.

3.3. Benchmarking Results

This subsection describes the results obtained by running the experiments created on the available algorithms. The code used to build and run these benchmarks can be accessed at `https://github.com/VytautasPau/HUBenchmark.git`.

3.3.1. Algorithms Used In Benchmarking

This section describes all of the algorithms used in the experiments and provides the final benchmark results. These algorithms were selected based on a few main factors: the algorithm code was made public by the authors and opened to use, and the algorithm solved at least one of the HU tasks.

The algorithm code was gathered from the author's GitHub or personal pages. All of the code used in the experiments, along with links to the author's pages, is provided in the GitHub repository (`https://github.com/VytautasPau/HUBenchmark.git`). Algorithms were

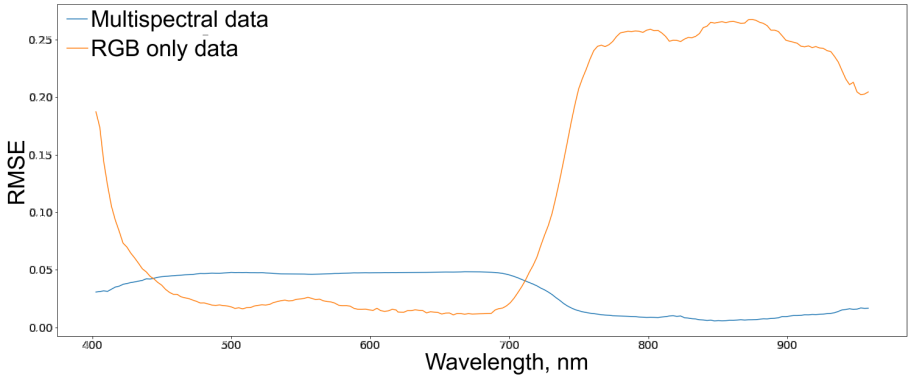


Figure 3.4 A comparison of hyperspectral data generation accuracy measured in RMSE between models.

implemented using the MATLAB software and the Python programming language, and ran using the MATLAB Python engine, which adds additional overheads to the calculations. For these reasons, direct performance comparison against pure MATLAB implementation is not recommended. This subsection describes the algorithms that were tested:

1. SUnSAL [11] solves an L1-L2 norm optimisation problem with several constraints: positivity, which checks if all resulting values are greater than or equal to 0; and Add-to-one, which calculates if the sum of the results (abundances) is equal to 1. The algorithm tries to minimise the L1 and L2 regularisation norms. In other words, L1 and L2 norm optimisation is simultaneously a sparse regression calculation on both linear and squared values.

2. SUnSAL-TV [41] is an extension of the SUnSAL algorithm that adds an isotropic or non-isotropic total variation spatial regularisation.

3. S^2WSU [108] is an algorithm that uses spectral and spatial data at the same time to calculate a sparse unmixing matrix.

4. CNMF [104] is an algorithm that fuses high-spatial-resolution multispectral data and high-spectral-resolution hyperspectral data to calculate image endmembers and unmix these spectra.

5. R-CoNMF [54] is an algorithm that performs three essential steps to find the endmembers, gather their signatures, and calculate the unmixing matrix.

6. SGSNMF [98] considers the spatial data and pixel locations, and runs assuming that unmixing matrices are sparse.

7. RSNMF [36] is a total variation regularised blind unmixing algorithm that considers pixel location and their correlation to nearby pixels.

8. ALMM [37] is a linear model that uses an endmember dictionary to help calculate spectral variability.

3.3.2. Results

Endmember robustness. This experiment was conducted using an artificial dataset generated using the pattern depicted in Figure 2.7. The pattern has a ground truth counterpart: a crop version of the same image but with 20 different classes (collections of more than 2 endmembers assigned to each pixel), which is labelled in the picture. Twenty-one randomly selected endmembers were mixed with different abundances using this classification pattern. The abundances selected followed a few steps that are also shown in Figure 3.5:

- 10 different datasets were created using the same 21 endmembers to add statistical differences to calculations.

- Endmembers were randomly placed into groups to create different numbers of endmembers, from 2 to 21, for each of the classes in the pattern.

- For each group of endmembers, uniformly distributed abundances were created.

- The other nine variations of abundances were randomly selected and mixed into 10 different hyperspectral images.

Figure 3.6 and Table 3.1 present the results obtained from the algorithms, which calculate the RMSE metric between the predicted values and the ground truth abundances generated. In Figure 3.6, almost all

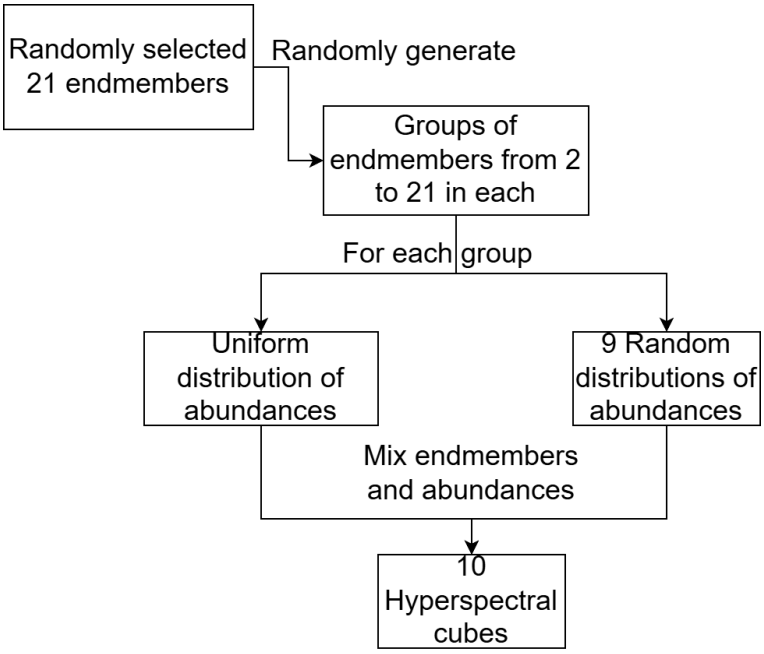


Figure 3.5 Endmember robustness experiment diagram.

algorithms, excluding RSNMF, show a consistent RMSE with different numbers of endmembers in the image, with SGSNMF having the largest errors and, in turn, the worst performance. In contrast, SUnSAL has the lowest error of all algorithms on average. The SGSNMF and RSNMF algorithms have the most significant value distributions among these algorithms. The smaller distributions yield more consistent results for these algorithms, whereas RSNMF is inconsistent with low numbers of endmembers. Table 3.1 displays the same information as Figure 3.6, but in a numerical form with the values averaged instead of their distributions. Bolded values show the best results for each algorithm relative to the number of endmembers.

Table 3.1: Endmember robustness experiment results with average RMSE values for each endmember group and algorithm. (Columns list algorithms tested, and rows are several endmembers.)

No. of endmembers	**SUnSAL**	**SUnSAL-TV**	**SGSNMF**	**RSNMF**
2	0.00124	0.0386	266.85	0.1
3	0.00193	0.038	274.59	0.096
4	0.00127	0.0494	203.004	0.091
5	0.00196	0.0424	171.066	0.089
6	0.00117	0.0445	**124.579**	0.083
7	0.00189	0.0381	349.082	0.081
8	0.0021	0.044	219.953	0.076
9	0.00138	0.0411	220.569	0.074
10	0.00124	0.0368	299.613	0.069
11	0.00169	0.0446	342.694	0.067
12	**0.00111**	0.0377	251.486	0.066
13	0.00152	0.0366	246.71	0.062
14	0.00174	**0.0346**	202.204	0.061
15	0.00133	0.0390	404.76	0.0588
16	0.00139	0.0369	177.42	0.054
17	0.00171	0.0383	215.487	0.054
18	0.00145	0.0360	219.009	0.054
19	0.00164	0.0391	335.102	0.051
20	0.00183	0.0358	173.696	0.049
21	0.00163	0.0354	218.699	**0.047**
Mean	0.001561	0.039345	245.82865	0.06914
Stdev	0.0002924	0.0038326	70.73378	0.016384

Robustness to noise. As with the previous experiment, hyperspectral images were generated using the classified surface image of Figure 2.7 to better represent the distribution of endmembers in the hyperspectral image. As described in the methodology in section 3.3.1, an image with five endmembers was generated, and artificial noise was added. Figure 3.7 shows the average results of 10 runs each of RMSE for each tested algorithm with different amounts of artificial noise added,

Table 3.2: Endmember robustness experiment results with average RMSE values for each endmember group and algorithm. (Columns list algorithms tested, and rows are several endmembers.) Cont.

No. of endmembers	**S2WSU**	**R-CoNMF**	**CNMF**	**ALMM**
2	**0.0001**	0.088	0.093	0.162
3	0.0064	0.086	0.091	0.157
4	0.0054	0.091	0.083	0.155
5	0.0038	**0.082**	0.081	0.151
6	0.0066	0.086	0.075	0.151
7	0.0076	0.091	0.076	0.143
8	0.010	0.090	0.072	0.150
9	0.013	0.090	0.073	0.148
10	0.011	0.090	0.065	0.146
11	0.013	0.0907	0.060	0.146
12	0.013	0.0904	0.055	0.141
13	0.0169	0.0906	0.057	0.146
14	0.015	0.0899	0.050	0.143
15	0.016	0.0874	0.049	0.145
16	0.019	0.0894	0.041	0.139
17	0.0209	0.090	0.042	0.142
18	0.0207	0.0909	0.042	0.140
19	0.0219	0.0909	0.039	0.140
20	0.0232	0.0916	**0.032**	0.138
21	0.0204	0.0895	0.035	**0.135**
Mean	0.013195	0.089265	0.06055	0.1459
Stdev	0.0067225	0.0023369	0.019121	0.006858

including the noise generated from real camera noise characteristics. Table 3.3 shows the mean and standard deviation of RMSE values for each of the algorithms over all of the noise levels. This Figure shows that the SUnSAL algorithm has a very linear correlation between the RMSE and the amount of noise in the image. RMSE is the worst of all algorithms using real-life noise characteristics. Other algorithms yield almost consistent results across all noise levels. The RSNMF algorithm

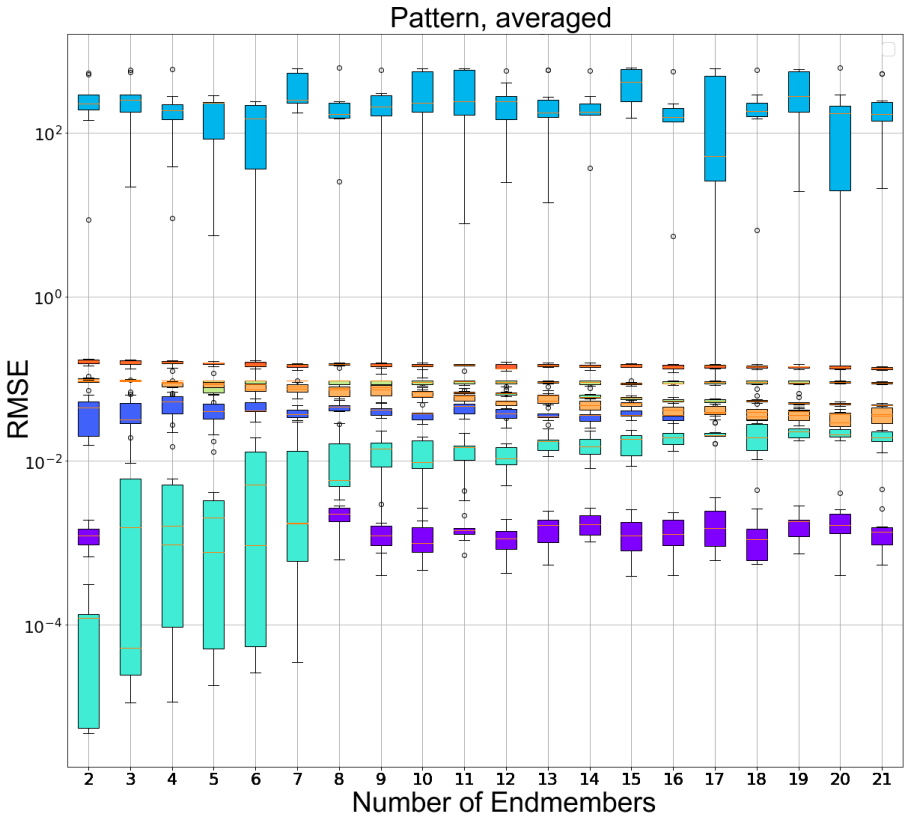


Figure 3.6 Endmember robustness experiment result with box plots for each endmember group and algorithm. (Colours: purple, SUnSAL; dark blue, SUnSAL-TV; blue, SGSNMF; light blue, S2WSU; cyan, RSNMF; yellow, R-CoNMF; orange, CNMF; red, ALMM.) A combined synthetic IEEE GRSS and USGS spectral library dataset was used as test data.

produces the most accurate overall RMSE, which increases at higher noise levels, and achieves the most accurate unmixing results in real noise experiments.

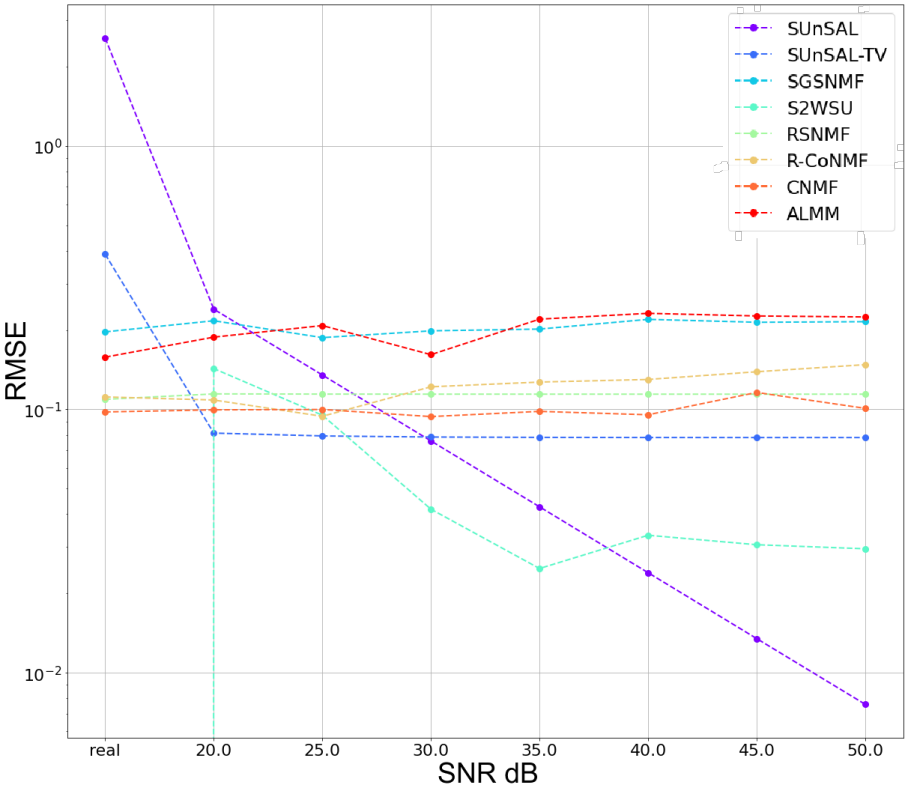


Figure 3.7 Algorithm robustness to noise experiment results. A combined synthetic dataset of the IEEE GRSS and USGS spectral libraries, with added noise, was used as test data. The values shown are the averages of RMSE all pixels and endmembers.

Table 3.3: Algorithm robustness to noise experiment results. Mean and standard deviation values of each algorithm over all of the tested noise levels.

Algorithm	RMSE Mean	RMSE Stdev
Sunsal	1.024	2.553
Sunsal-tv	0.151	0.211
SGSNMF	0.515	0.056
S2WSU	0.131	0.141
RSNMF	0.135	**0.006**
R-CoNMF	0.230	0.109
CNMF	**0.100**	0.014
ALMM	0.496	0.100

Image size difference. The RMSE results of 9-times down-scaled images are shown in Figure 3.8. Compared with the images scaled down 4 times, Figure 3.9 shows the algorithm performance results. Both figures show similar results, correlating with endmember robustness experiment RMSE values where images are not scaled down.

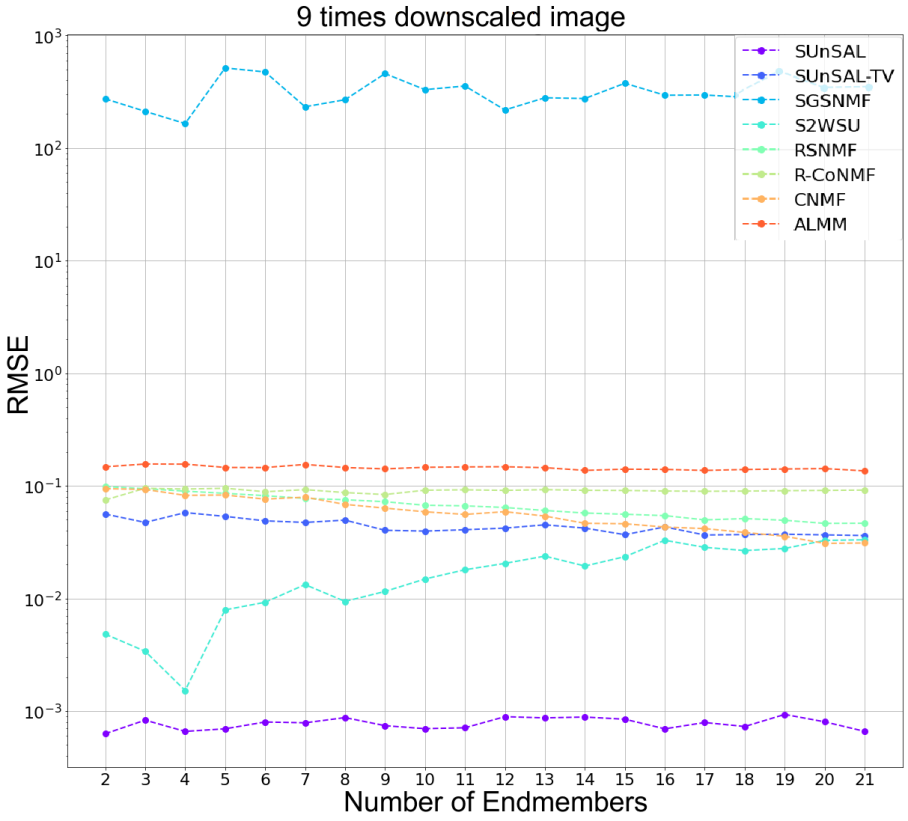


Figure 3.8 Algorithm performance with 9-times down-scaled hyperspectral images. A combined synthetic dataset of IEEE GRSS and USGS spectral library was scaled down 9 times. The values shown are the averages of RMSE, including all pixels and endmembers.

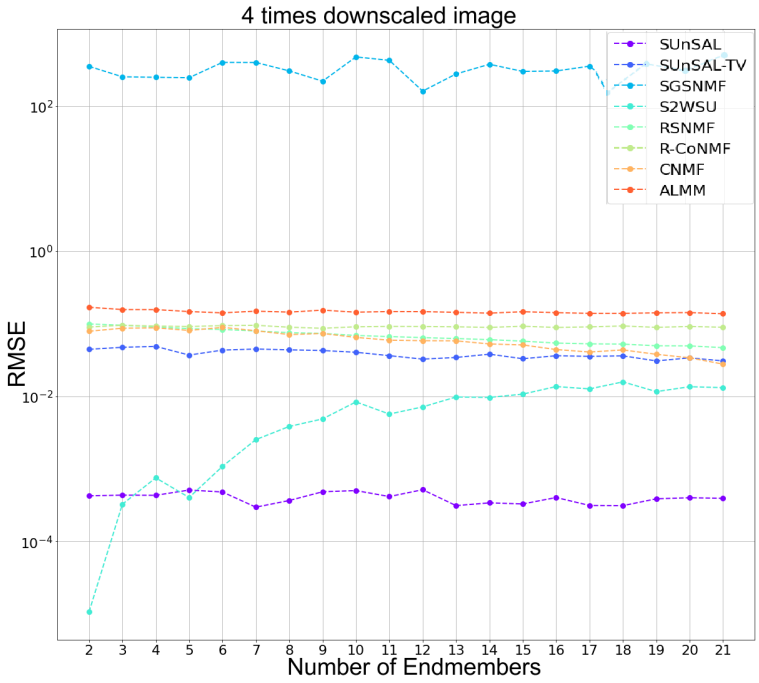


Figure 3.9 Algorithm performance with 4-times down-scaled hyperspectral images. A combined synthetic dataset of the IEEE GRSS and USGS spectral library was scaled down 4 times. The values shown are the averages of RMSE all pixels and endmembers.

To facilitate the more accurate comparison of these results, Table 3.4 was created, displaying the averages of the RMSE and SAD metrics for each algorithm and each set of scaled images. Negative SRE values represent worse reconstruction than positive values because the higher the SRE, the better the signal reconstruction. This table determines the effects of image scaling on the results of the tested algorithms. All tested algorithms yielded consistent results across different image scales. The SGSNMF algorithm achieved the worst results, while SUnSAL achieved the lowest RMSE results. The SRE values were consistent across the image scales, with S2WSU showing a significant difference in the metric values. Overall, the RSNMF and SUnSAL algorithm achieved the lowest RMSE results of all the values.

Table 3.4: Image size difference algorithm comparison results.

		Metrics	
Algorithm	**Downscale**	**RMSE**	**SRE**
SUnSAL	1	0.003	19.800
	4	0.001	**20.095**
	9	**0.001**	16.549
	Mean	0.0016	18.81
	Stdev	0.00115	1.967
SUnSAL-TV	1	0.046	2.184
	4	0.040	1.643
	9	0.047	1.114
	Mean	0.0443	1.647
	Stdev	0.0037	0.535
SGSNMF	1	257.327	-20.889
	4	305.677	-28.925
	9	197.180	-32.716
	Mean	253.3946	-27.51
	Stdev	54.3552	6.03913
S2WSU	1	0.176	2.125
	4	0.020	4.603
	9	0.042	1.558
	Mean	0.0793	2.762
	Stdev	0.084435	1.6193

Table 3.5: Image size difference algorithm comparison results, cont.

		Metrics	
Algorithm	**Downscale**	**RMSE**	**SRE**
RSNMF	1	0.053	0.569
	4	0.051	0.311
	9	0.050	0.452
	Mean	0.0513	0.444
	Stdev	0.00152	0.1291
R-CoNMF	1	0.217	-4.189
	4	0.218	-5.496
	9	0.216	-5.330
	Mean	0.217	-5.00
	Stdev	0.001	0.7166
CNMF	1	0.045	-0.025
	4	0.041	-0.153
	9	0.042	0.009
	Mean	0.0426	-0.0563
	Stdev	0.00208	0.0854
ALMM	1	0.195	-4.982
	4	0.204	-5.230
	9	0.200	-5.039
	Mean	0.1996	-5.0836
	Stdev	0.0045	0.1298

During the benchmark experiment calculations, a log of the time spent on calculations for each algorithm was recorded to compare the time differences between them. This is not a standardised test, so the time comparison is only relative and will depend on the hardware. To compare the running times of the different algorithms with each dataset, all experiments were performed on a desktop computer equipped with a 12-core, 24-thread AMD CPU and 64 GB of RAM, as well as an Nvidia GTX 1080 Ti with 11 GB of VRAM. The average recorded times are shown in Table 3.6.

Table 3.6: Algorithm average calculation times in seconds.

Algorithm	**Time (s)**
SUnSAL	228
SUnSAL-TV	2671
SGSNMF	636
S2WSU	7451
RSNMF	2106
R-CoNMF	257
CNMF	3855
ALMM	851

3.4. HUNET Model Experiments

This section describes the experiments conducted to evaluate the performance of the newly created HUNET model in comparison to the state-of-the-art transformer-based Hyperspectral Unmixing model, as well as the results achieved when using the proposed HU benchmarking methodology.

3.4.1. Datasets Used

This section describes the datasets used in the existing and HUNET models, as well as their evaluation and comparison. Freely available hyperspectral datasets that were selected to be used in algorithm performance experimentation are analysed, and the newly created HU dataset described in section 2.2.1 is included. The following datasets were used:

- *DC Mall* [75] [68]. This is an area in Washington, DC, with a size of 1208 by 307 pixels and 191 spectral bands. Ground truth was created for classification with the following classes: Roofs, Streets, Paths, Grass, Trees, Water, and Shadows.
- *Samson* [75]. This is a hyperspectral data cube cut to a size of 95 by 95 pixels with 156 spectral bands and three different classes.
- *Apex* [75]. This is a hyperspectral dataset measuring 110 by 110 pixels, with 285 spectral bands and four different classes.
- BFHUD Cube 1: 1024 pixels wide, 3177 pixels long, with 224 spectral depth bands.
- BFHUD Cube 2: 1024 pixels wide, 3047 pixels long, with 224 spectral depth bands.
- BFHUD Cube 3: 1024 pixels wide, 2815 pixels long, with 224 spectral depth bands.

3.4.2. HUNET Benchmark Results

This section describes the results of running the newly created U-Net-based neural network model through the proposed HU benchmark described in section 2.4.

The newly created HUNET algorithm was tested using the three benchmarking experiments. To compare the HUNET with other algorithms, the RMSE and Reconstruction Error metrics were calculated. Each experiment was performed 10 times, and the average values were recorded for both metrics. The best results are marked in bold.

The results of the robustness to noise benchmarks are provided in Table 3.7.

SNR dB	Real	20	25	30	35	40	45	50
RE	0.1123	0.1327	0.0930	0.0916	**0.0720**	0.2581	0.0928	0.0989
RMSE	0.1521	0.1367	**0.1321**	0.1498	0.1749	0.1627	0.1736	0.1823

Table 3.7: HUNET robustness to noise experiment results.

The results of the endmember robustness benchmarks are provided in Table 3.8.

Endmembers	**2**	**3**	**4**	**5**	**6**	**7**	**8**
RE	0.0587	**0.0527**	0.0968	0.0901	0.0827	0.0792	0.0716
RMSE	0.0111	0.0506	0.0138	0.0599	0.0850	0.0852	0.0586
Endmembers	**9**	**10**	**11**	**12**	**13**	**14**	**15**
RE	0.0601	0.0733	0.0789	0.0690	0.0832	0.0796	0.0922
RMSE	**0.0101**	0.0123	0.0143	0.0122	0.0113	0.0142	0.0152
Endmembers	**16**	**17**	**18**	**19**	**20**	**21**	
RE	0.0785	0.0833	0.0893	0.0921	0.0967	0.1043	
RMSE	0.0172	0.0233	0.0283	0.0382	0.0365	0.0312	

Table 3.8: HUNET model endmember robustness experiment results.

The results of the image size benchmarks are provided in Table 3.9.

		Metrics	
Algorithm	**Downscale**	**RMSE**	**RE**
HUNET	1	**0.0322**	0.0987
HUNET	4	0.0421	0.0873
HUNET	9	0.0657	**0.0774**

Table 3.9: HUNET model image size experiment results.

From the benchmarking results, the following conclusions and observations can be drawn:

- During the robustness to noise experiment, HUNET achieved similar RMSE and RE results for all artificial noise levels. Compared to the other algorithms, this result was close to the best, only trailing behind the SUnSAL and S2WSU algorithms.
- The enmember robustness experiment showed similar results, as HUNET achieved RMSE and RE values increasing with the number of endmembers. Compared to other algorithms, the HUNET was less accurate than the SUnSAL, SUnSAL-TV, and S2WSU algorithms for all endmember pairs.
- Image size benchmark results show a decreasing model RMSE value for each of the endmembers, while the RE value decreased due to the reduction in the number of pixels.

3.4.3. Experimentation Results

This section describes the experiment used to test the HUNET model and the existing transformer model using BFHUD. The testing methodology of Ghosh et al. [28] was selected to create the experiment baseline. Full experimentation steps are listed below:

- The selection of a few existing datasets used in testing the transformer neural network created by Ghosh et al. [28]. This included the *Samson* and *Washington DC* datasets.
- The adaptation of BFHUD to the existing model, followed by running a few test passes to gather a set of hyperparameters.
- The adaptation of the HUNET model to the existing data pipeline developed by Ghosh et al. [28] to keep testing between datasets and models consistent.
- For both model training phases, reconstruction error (RE, see equation 2.3) and spectral angle distance (SAD, see equation 2.4) were used as losses. An additional cosine similarity loss (see equation 2.5) was added to the HUNET model.

- For testing the results, RMSE (see equation 2.1) and SAD metrics were used.

The RMSE, SAD, and RE results achieved by running the HUNET model and the transformer model developed by Ghosh et al. [28] with three selected datasets are provided in Table 3.10, where BFHUD is the newly created dataset. Extracted endmembers compared with ground truth endmembers are provided as separate graphs for each of the six combinations of the two models and three datasets tested. Figures 3.10, 3.11, 3.12, 3.13, 3.14, and 3.15 show the extracted and ground truth endmembers for each of the classes, models, and datasets. In these graphs, the x-axis represents the wavelength array index values, and the y-axis the normalised reflectance values. The endmember values, extracted using the HUNET model, are closer to the ground truth values than those extracted by the transformer-based model.

Dataset	**Model**	**RE**	**mean RMSE**	**mean SAD**
BFHUD	Transformer	0.3129	0.5054	**0.4135**
	HUNET	**0.0754**	**0.3625**	0.6632
DC Mall	Transformer	**0.0253**	0.3853	0.3002
	HUNET	0.0451	**0.3832**	**0.1928**
Samson	Transformer	0.1651	0.6027	**0.2411**
	HUNET	**0.0401**	**0.5215**	0.7880

Table 3.10: RMSE, SAD, and RE metric results of proposed and transformer models for different datasets, including BFHUD. Bold values show better results for each dataset and metric between the two models.

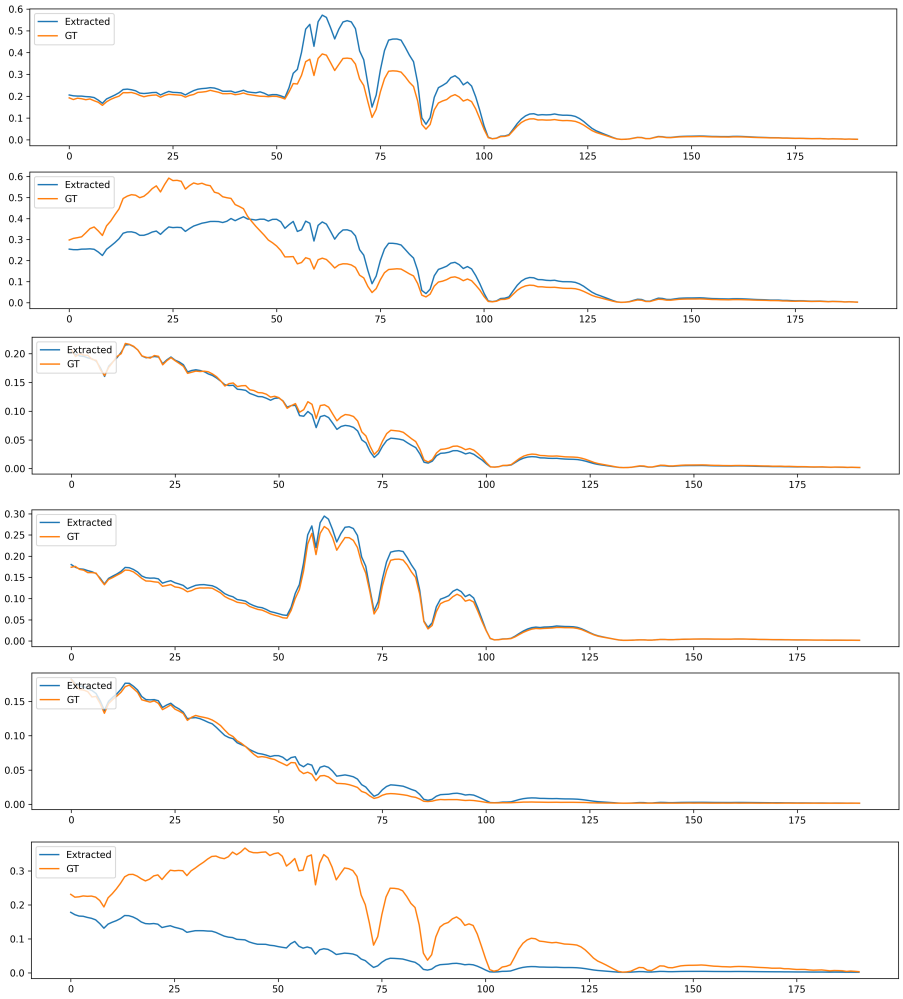


Figure 3.10 Extracted and ground truth endmembers for each class in the DC Mall dataset. The results were acquired by the HUNET model.

3.5. Conclusions of the Chapter

- The newly proposed HUNET model achieved a better reconstruction error rate than the transformer-based HU model in two out of the three hyperspectral datasets tested.
- Extracted endmembers comparing the HUNET model and the transformer-based HU model showed the more accurate extraction of endmembers from hyperspectral datasets when using the

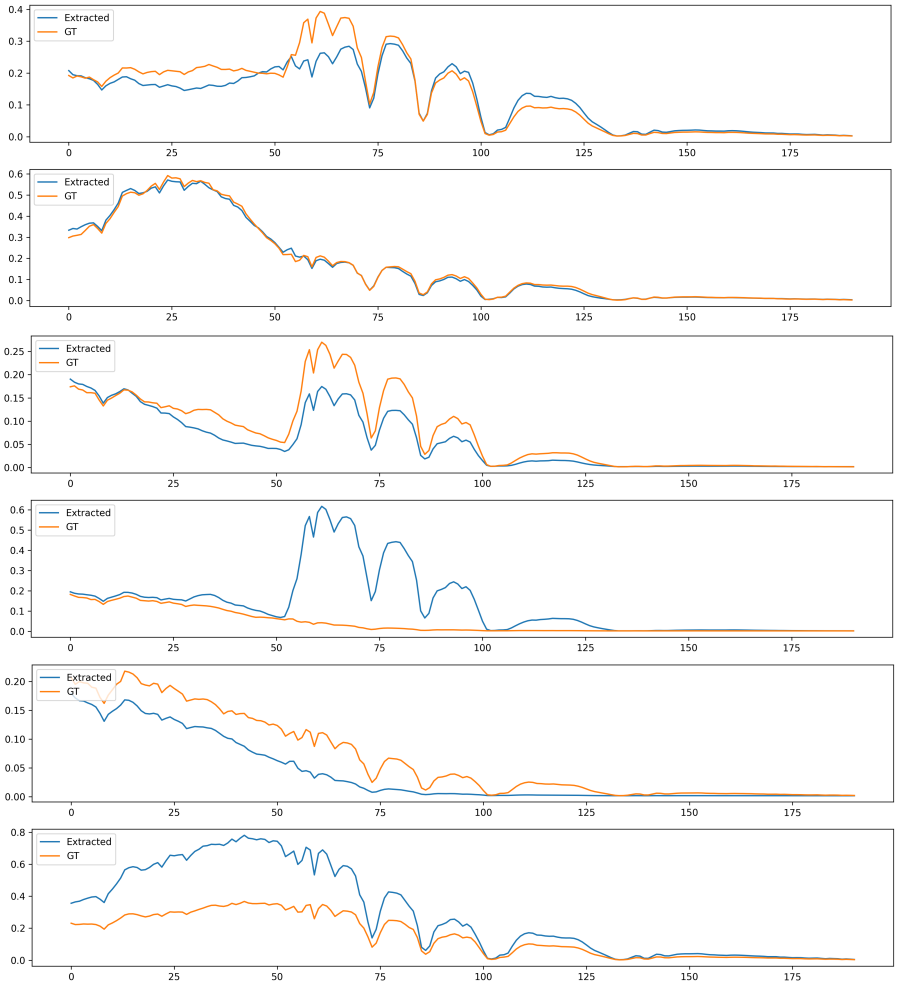


Figure 3.11 Extracted and ground truth endmembers for each class in the DC Mall dataset. The results were acquired by the transformer model developed by Ghosh et al. [28].

proposed U-Net-based model.

- The CNMF and RSNMF algorithms achieved up to three times lower RMSE values when the endmember counts were increased from 2 to 21.

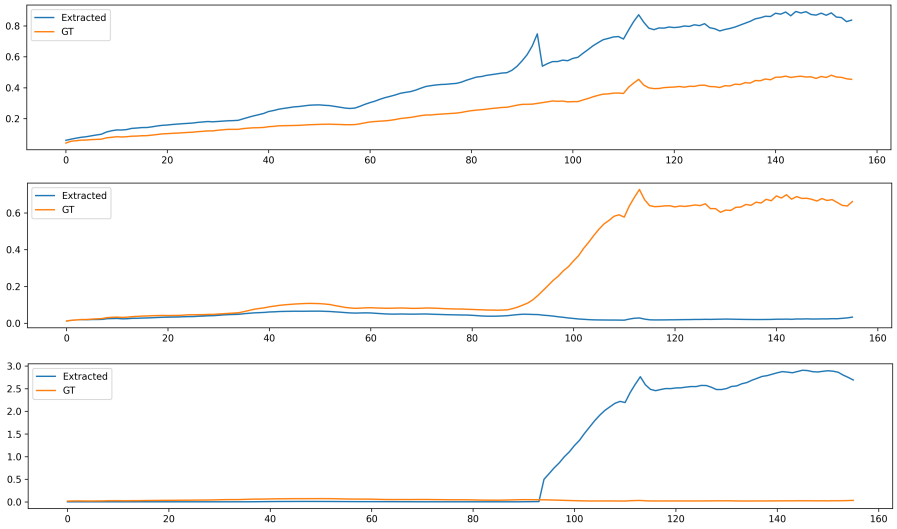


Figure 3.12 Extracted and ground truth endmembers for each class in the Samson dataset. The results were acquired by the HUNET model.

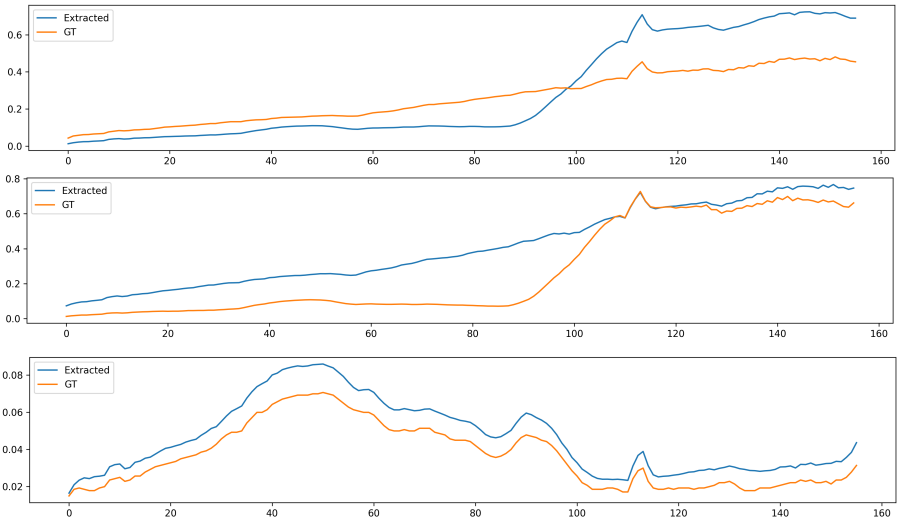


Figure 3.13 Extracted and ground truth endmembers for each class in the Samson dataset. The results were acquired by the transformer model developed by Ghosh et al. [28].

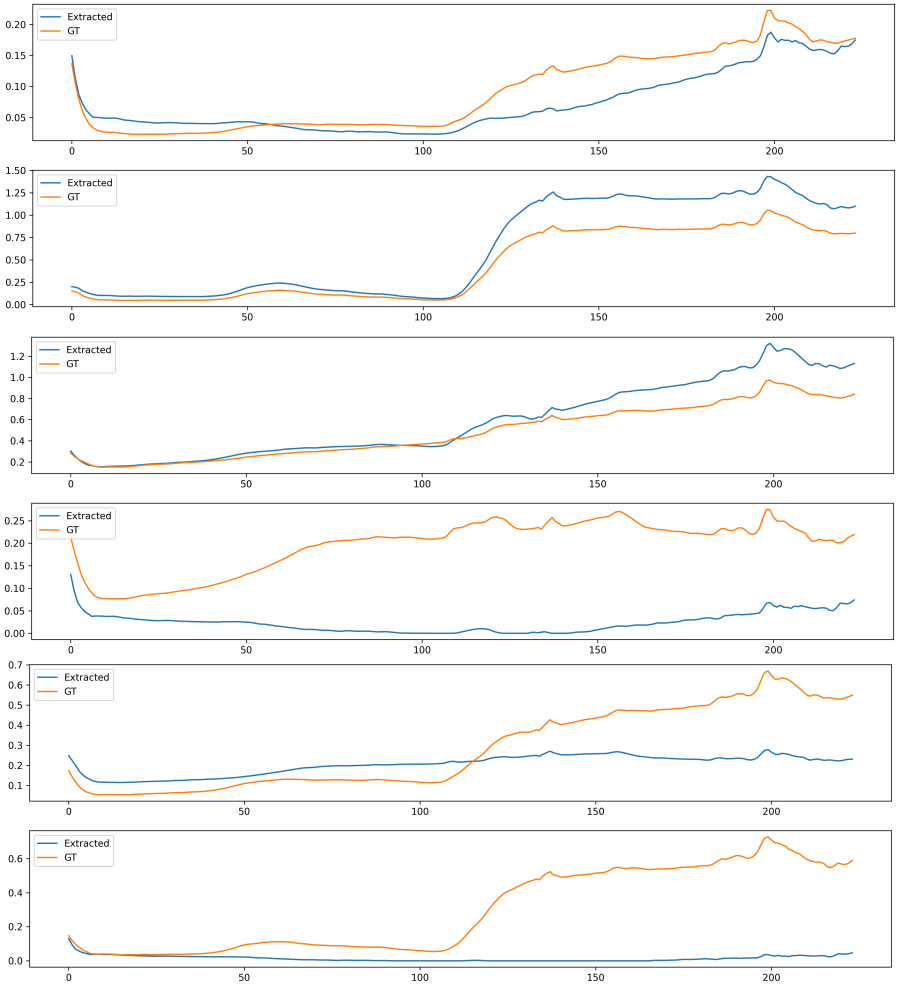


Figure 3.14 Extracted and ground truth endmembers for each class in BFHUD. The results were acquired by the HUNET model.

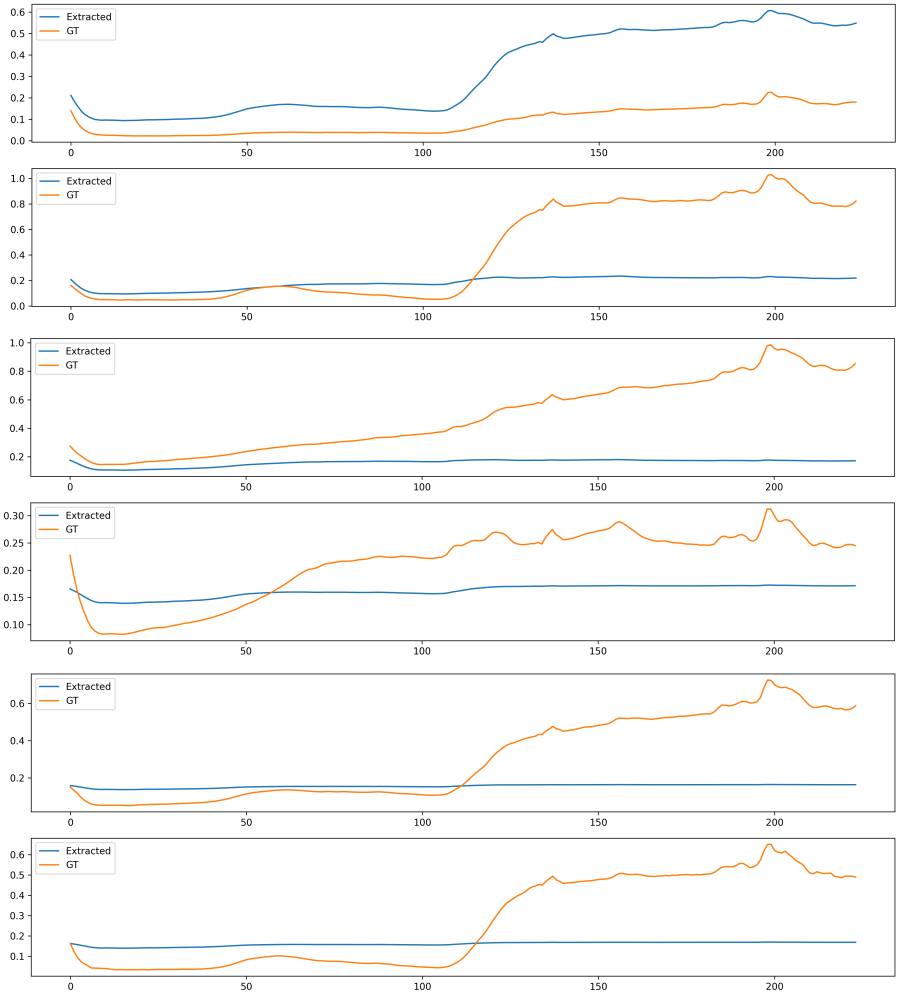


Figure 3.15 Extracted and ground truth endmembers for each class in BFHUD. The results were acquired by the transformer model developed by Ghosh et al. [28].

1.
   - HU algorithms like SUnSAL, SUnSAL-TV, R-CoNMF and ALMM showed almost no difference in RMSE values when the endmember counts increased from 2 to 21 and are therefore categorised as algorithms independent of the number of endmembers. The performance of other algorithms, including the newly proposed HUNET model, was dependent on the number of endmembers.
   - Image size difference experimentation showed no difference in performance for the RSNMF, R-CoNMF, CNMF, ALMM, SUnSAL, and SUnSAL-TV algorithms, which shows that the downscaled hyperspectral images contained enough information for algorithms to extract the endmember data. The resulting RMSE values changed less than 10% (or 0.002) between the experiments with different-sized hyperspectral images.
   - When running the entire benchmark, algorithm calculation times were lowest for the SUnSAL and R-CoNMF algorithms, with S2WSU being the fastest, slower than SUnSAL by about 32 times.

2.
   - When testing algorithm noise robustness, the HU algorithms that were created earlier, like SUnSAL and SUnSAL-TV, produced more accurate results and offered the lowest RMSE values (around 0.008 for SUnSAL and 0.08 for SUnSAL-TV) when the noise added to the hyperspectral images was Gaussian. However, the worst results were observed when the added noise was based on real hyperspectral camera noise. The newly created HUNET model achieved nearly identical RMSE values independently of the artificial noise.
   - In turn, HU algorithms like CNMF, SGSNMF, and RSNMF performed almost the same independently of the type and amount of noise added, with RMSE values equal to around 0.1 for CNMF and RSNMF and 0.2 for SGSNMF.

3. Generating hyperspectral data when using multispectral data resulted in much lower RMSE values (around 0.034 averaged over

all of the wavelengths) compared to hyperspectral data generated from RGB images (average RMSE values of 0.1023 over all of the wavelengths), due to the high correlation of near-infrared wavelengths.

4.
   - The newly proposed HUNET model architecture reduces the number of hyperparameters required while achieving lower HU RE and RMSE values compared to the transformer-based unmixing algorithm by 0.12 and 0.0732, respectively, on average across all datasets.

5.
   - Using the HUNET model architecture, better values of RE by 0.28 and RMSE by 0.14 were achieved compared to the transformer-based unmixing algorithm for the UAV-gathered hyperspectral dataset.
   - The HUNET model achieved a lower mean RMSE (on average 12% lower) in all three hyperspectral datasets tested (BFHUD, DC Mall and Samson) compared to the transformer-based unmixing algorithm. In contrast, the transformer-based model achieved lower mean SAD values (on average 45% lower) in two of the three datasets.

[1] A. T. I. 2024. Aurelia x6 standard. https://aurelia-aerospace.com/product/aurelia-x6-standard/, 2024. [Last read on: 2025-05-10].

[2] E. M. Achata, C. Esquerre, A. A. Gowen, and C. P. O'Donnell. Feasibility of near infrared and raman hyperspectral imaging combined with multivariate analysis to assess binary mixtures of food powders. *Powder Technology*, 336:555–566, 2018. ISSN 0032-5910. doi: https://doi.org/10.1016/j.powtec.2018.06.025. URL https://www.sciencedirect.com/science/article/pii/S0032591018304650.

[3] S. Annam and A. Singla. Spectral unmixing of heavy metal content in agricultural soil using hyperspectral remote sensing data. In *2021 Sixth International Conference on Image Information Processing (ICIIP)*, volume 6, pages 433–438, 2021. doi: 10.1109/ICIIP53038.2021.9702646.

[4] N. Audebert, B. L. Saux, and S. Lefèvre. Generative adversarial networks for realistic synthesis of hyperspectral samples. 2018. doi: 10.48550/ARXIV.1806.02583. URL https://arxiv.org/abs/1806.02583.

[5] R. Bai, J. Zhou, S. Wang, Y. Zhang, T. Nan, B. Yang, C. Zhang, and J. Yang. Identification and classification of coix seed storage years based on hyperspectral imaging technology combined with deep learning. *Foods*, 13(3), 2024. ISSN 2304-8158. doi: 10.3390/foods13030498. URL https://www.mdpi.com/2304-8158/13/3/498.

[6] P. Bajcsy and P. Groves. Methodology for hyperspectral band selection. *Photogrammetric Engineering and Remote Sensing journal*, 70:793–802, 08 2004. doi: 10.14358/PERS.70.7.793.

[7] BaySpec. OCI™-F Hyperspectral Imager (VIS-NIR, SWIR), 2021. URL https://www.bayspec.com/spectroscopy/oci-f-hyperspectral-imager/. [Last read on: 2025-05-10].

[8] M. Beitollahi and S. A. Hosseini. Using savitsky-golay smoothing filter in hyperspectral data compression by curve fitting. In *Electrical Engineering (ICEE), Iranian Conference on*, pages 452–457, 2018. doi: 10.1109/ICEE.2018.8472702.

[9] M. Bevilacqua and Y. Berthoumieu. Unsupervised hyperspectral

band selection via multi-feature information-maximization clustering. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 540–544, 2017. doi: 10.1109/ICIP.2017.8296339.

[10] A. Bhargava, A. Sachdeva, K. Sharma, M. H. Alsharif, P. Uthansakul, and M. Uthansakul. Hyperspectral imaging and its applications: A review. *Heliyon*, 10(12):e33208, 2024. ISSN 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2024.e33208. URL https://www.sciencedirect.com/science/article/pii/S2405844024092399.

[11] J. M. Bioucas-Dias and M. A. T. Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2010. doi: 10.1109/WHISPERS.2010.5594963.

[12] 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 Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, 2012. doi: 10.1109/JSTARS.2012.2194696.

[13] 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*, 6:374–384, 2020. doi: 10.1109/TCI.2019.2948726.

[14] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130401. URL https://doi.org/10.1145/130385.130401.

[15] J. Burger and P. Geladi. Hyperspectral nir image regression part i: Calibration and correction. *Journal of Chemometrics*, 19:355 – 363, 05 2005. doi: 10.1002/cem.938.

[16] J. Burger and P. Geladi. Hyperspectral nir image regression part i: Calibration and correction. *Journal of Chemometrics*, 19:355 – 363, 05 2005. doi: 10.1002/cem.938.

[17] Y. Cai, X. Liu, and Z. Cai. Bs-nets: An end-to-end framework

for band selection of hyperspectral image. *CoRR*, abs/1904.08269, 2019. URL http://arxiv.org/abs/1904.08269.

[18] N. Caporaso, M. B. Whitworth, and I. D. Fisk. Total lipid prediction in single intact cocoa beans by hyperspectral chemical imaging. *Food Chemistry*, 344:128663, 2021. ISSN 0308-8146. doi: https://doi.org/10.1016/j.foodchem.2020.128663. URL https://www.sciencedirect.com/science/article/pii/S0308814620325255.

[19] C.-I. Chang and Y. Li. Recursive band processing of automatic target generation process for finding unsupervised targets in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 54:1–14, 05 2016. doi: 10.1109/TGRS.2016.2553845.

[20] J. Cruz-Tirado, M. Oliveira, M. de Jesus Filho, H. T. Godoy, J. M. Amigo, and D. F. Barbin. Shelf life estimation and kinetic degradation modeling of chia seeds (salvia hispanica) using principal component analysis based on nir-hyperspectral imaging. *Food Control*, 123:107777, 2021. ISSN 0956-7135. doi: https://doi.org/10.1016/j.foodcont.2020.107777. URL https://www.sciencedirect.com/science/article/pii/S0956713520306939.

[21] L. Deng, B. Zhou, J. Ying, and R. Zhao. A noise estimation method for hyperspectral image based on stacked autoencoder. *IEEE Access*, 11:89835–89843, 2023. doi: 10.1109/ACCESS.2023.3307200.

[22] L. Dong, Y. Yuan, and X. Lu. Spectral-spatial joint sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–12, 2020. doi: 10.1109/TGRS.2020.3006109.

[23] A. I. Durojaiye, S. T. Olorunsogo, B. A. Adejumo, A. Babawuya, and I. I. Muhamad. Deep learning techniques for the exploration of hyperspectral imagery potentials in food and agricultural products. *Food and Humanity*, 3:100365, 2024. ISSN 2949-8244. doi: https://doi.org/10.1016/j.foohum.2024.100365. URL https://www.sciencedirect.com/science/article/pii/S294982442400140X.

[24] G. Elmasry, M. Kamruzzaman, D.-W. Sun, and P. Allen. Principles and applications of hyperspectral imaging in quality evaluation of agro-food products: A review. *Critical reviews in food science and nutrition*, 52:999–1023, 11 2012. doi: 10.1080/10408398.2010.543495.

[25] S. D. Fabiyi, P. Murray, J. Zabalza, and J. Ren. Folded lda: Extending the linear discriminant analysis algorithm for feature extraction and data reduction in hyperspectral remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:12312–12331, 2021. doi: 10.1109/JSTARS.2021.3129818.

[26] X. R. Feng, H. C. Li, S. Liu, and H. Zhang. Correntropy-based autoencoder-like nmf with total variation for hyperspectral unmixing. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020. doi: 10.1109/LGRS.2020.3020896.

[27] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. ISSN 0898-1221. doi: https://doi.org/10.1016/0898-1221(76)90003-1. URL https://www.sciencedirect.com/science/article/pii/0898122176900031.

[28] P. Ghosh, S. K. Roy, B. Koirala, B. Rasti, and P. Scheunders. Hyperspectral unmixing using transformer network. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022. doi: 10.1109/TGRS.2022.3196057.

[29] A. A. Goodenough and S. D. Brown. Dirsig5: Next-generation remote sensing data and image simulation framework. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(11):4818–4833, 2017. doi: 10.1109/JSTARS.2017.2758964.

[30] M. F. Guerri, C. Distante, P. Spagnolo, F. Bougourzi, and A. Taleb-Ahmed. Deep learning techniques for hyperspectral image analysis in agriculture: A review. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 12:100062, 2024. ISSN 2667-3932. doi: https://doi.org/10.1016/j.ophoto.2024.100062. URL https://www.sciencedirect.com/science/article/pii/S266739322400005X.

[31] N. Gul, K. Muzaffar, S. Z. Shah, A. Assad, H. Makroo, and B. Dar. Deep learning hyperspectral imaging: a rapid and reliable alternative to conventional techniques in the testing of food quality and safety. *Quality Assurance and Safety of Crops & Foods*, 16:78–97, 02 2024. doi: 10.15586/qas.v16i1.1392.

[32] Z. Guo, T. Wittman, and S. Osher. L1 unmixing and its application to hyperspectral image enhancement. *Proc SPIE Conference*

on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, 05 2009. doi: 10.1117/12.818245.

[33] M. Halicek, H. Fabelo, S. Ortega, G. Marrero Callico, and B. Fei. In-vivo and ex-vivo tissue analysis through hyperspectral imaging techniques: Revealing the invisible features of cancer. *Cancers*, 11: 756, 05 2019. doi: 10.3390/cancers11060756.

[34] Z. Han, D. Hong, L. Gao, B. Zhang, and J. Chanussot. Deep half-siamese networks for hyperspectral unmixing. *IEEE Geoscience and Remote Sensing Letters*, 18(11):1996–2000, 2021. doi: 10.1109/LGRS.2020.3011941.

[35] B. Hapke. Bidirectional reflectance spectroscopy: 1. theory. *Journal of Geophysical Research: Solid Earth*, 86(B4):3039–3054, 1981. doi: https://doi.org/10.1029/JB086iB04p03039. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JB086iB04p03039.

[36] W. He, H. Zhang, and L. Zhang. Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3909–3921, 2017. doi: 10.1109/TGRS.2017.2683719.

[37] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu. An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Transactions on Image Processing*, 28(4): 1923–1938, 2019. doi: 10.1109/TIP.2018.2878958.

[38] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot. Spectralformer: Rethinking hyperspectral image classification with transformers. *CoRR*, abs/2107.02988, 2021. URL https://arxiv.org/abs/2107.02988.

[39] Z. Hua, X. Li, Q. Qiu, and L. Zhao. Autoencoder network for hyperspectral unmixing with adaptive abundance smoothing. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020. doi: 10.1109/LGRS.2020.3005999.

[40] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(00)00026-5. URL https://www.sciencedirect.com/science/article/pii/S0893608000000265.

[41] M. Iordache, J. M. Bioucas-Dias, and A. Plaza. Total variation

spatial regularization for sparse hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4484–4502, 2012. doi: 10.1109/TGRS.2012.2191590.

[42] M. Iordache, J. M. Bioucas-Dias, and A. Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):341–354, 2014. doi: 10.1109/TGRS.2013.2240001.

[43] X. Jia. Simplified maximum likelihood classification for hyperspectral data in cluster space. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 5, pages 2578–2580 vol.5, 2002. doi: 10.1109/IGARSS.2002.1026706.

[44] A. Kaya, K. Ataş, and S. Kahraman. Lidar-aided total variation regularized nonnegative tensor factorization for hyperspectral unmixing. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 5063–5066, 2021. doi: 10.1109/IGARSS47720.2021.9553137.

[45] J. Kilpys, L. Jukna, E. Stonevicius, R. Šimanauskienė, and L. Bevainis. *Žemės stebėjimas iš kosmoso*. 01 2021. ISBN 9786090705742. doi: 10.15388/vup-book-0007.

[46] P. D. P. Kogut. Multispectral vs. hyperspectral: Choose the right tech. `https://eos.com/blog/multispectral-vs-hyperspectral-imaging/`, 2024. [Last read on: 2025-05-10].

[47] B. Koirala, M. Khodadadzadeh, C. Contreras, Z. Zahiri, R. Gloaguen, and P. Scheunders. A supervised method for nonlinear hyperspectral unmixing. *Remote Sensing*, 11(20):2458, Oct 2019. ISSN 2072-4292. doi: 10.3390/rs11202458. URL `http://dx.doi.org/10.3390/rs11202458`.

[48] R. F. Kokaly, R. N. Clark, G. A. Swayze, K. E. Livo, T. M. Hoefen, N. C. Pearson, R. A. Wise, W. M. Benzel, H. A. Lowers, R. L. Driscoll, and A. J. Klein. Usgs spectral library version 7. Technical report, Reston, VA, 2017. URL `https://doi.org/10.3133/ds1035`. Report.

[49] S. J. Koppal. *Lambertian Reflectance*, pages 441–443. Springer US, Boston, MA, 2014. ISBN 978-0-387-31439-6. doi: 10.1007/978-0-387-31439-6_534. URL `https://doi.org/10.1007/978-0-387-31439-6_534`.

[50] Labsphere. Spectralon® diffuse reflectance targets.

https://www.labsphere.com/product/spectralon-reflectance-targets/, 2024. [Last read on: 2025-05-10].

[51] H. Lee and H. Kwon. Contextual deep CNN based hyperspectral classification. *CoRR*, abs/1604.03519, 2016. URL http://arxiv.org/abs/1604.03519.

[52] H. Li, R. Feng, L. Wang, Y. Zhong, and L. Zhang. Superpixel-based reweighted low-rank and total variation sparse unmixing for hyperspectral remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–19, 2020. doi: 10.1109/TGRS.2020.2994260.

[53] 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*, volume 3, pages III – 250–III – 253, 2008. doi: 10.1109/IGARSS.2008.4779330.

[54] 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*, 54(10):6076–6090, 2016. doi: 10.1109/TGRS.2016.2580702.

[55] W. Li, S. Prasad, E. W. Tramel, J. E. Fowler, and Q. Du. Decision fusion for hyperspectral image classification based on minimum-distance classifiers in thewavelet domain. In *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, pages 162–165, 2014. doi: 10.1109/ChinaSIP.2014.6889223.

[56] W. Li, Q. Liu, Y. Wang, and H. Li. 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, 2022. doi: 10.5194/isprs-archives-XLIII-B3-2022-405-2022. URL https://isprs-archives.copernicus.org/articles/XLIII-B3-2022/405/2022/.

[57] X. Li, R. Huang, and L. Zhao. Correntropy-based spatial-spectral robust sparsity-regularized hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–19, 2020. doi: 10.1109/TGRS.2020.2999936.

[58] Z. Li, H. Huang, Z. Zhang, and G. Shi. Manifold-based multi-deep

belief network for feature extraction of hyperspectral image. *Remote Sensing*, 14(6), 2022. ISSN 2072-4292. doi: 10.3390/rs14061484. URL https://www.mdpi.com/2072-4292/14/6/1484.

[59] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li. Manifold regularized sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2815–2826, 2013. doi: 10.1109/TGRS.2012.2213825.

[60] X. Lu, L. Dong, and Y. Yuan. Subspace clustering constrained sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3007–3019, 2020. doi: 10.1109/TGRS.2019.2946751.

[61] W. Lv and X. Wang. Overview of hyperspectral image classification. *Journal of Sensors*, 2020(1):4817234, 2020. doi: https://doi.org/10.1155/2020/4817234. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/4817234.

[62] D. G. Manolakis, R. B. Lockwood, and T. W. Cooley. *Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms*. Cambridge University Press, 2016.

[63] A. Maćkiewicz and W. Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993. ISSN 0098-3004. doi: https://doi.org/10.1016/0098-3004(93)90090-R. URL https://www.sciencedirect.com/science/article/pii/009830049390090R.

[64] M. Medina-García, E. A. Roca-Nasser, M. A. Martínez-Domingo, E. M Valero, A. Arroyo-Cerezo, L. Cuadros-Rodríguez, and A. M. Jiménez-Carvelo. Towards the establishment of a green and sustainable analytical methodology for hyperspectral imaging-based authentication of wholemeal bread. *Food Control*, 166:110715, 2024. ISSN 0956-7135. doi: https://doi.org/10.1016/j.foodcont.2024.110715. URL https://www.sciencedirect.com/science/article/pii/S0956713524004328.

[65] C. Mobley. From xyz to rgb. https://www.oceanopticsbook.info/view/photometry-and-visibility/from-xyz-to-rgb, 2021. [Last read on: 2025-05-10].

[66] L. Mou, P. Ghamisi, and X. X. Zhu. 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, 2018. doi: 10.1109/TGRS.2017.2748160.

[67] G. Mozgeris, S. Gadal, D. Jonikavičius, L. Straigytė, W. Ouerghemmi, and V. Juodkienė. Hyperspectral and color-infrared imaging from ultralight aircraft: Potential to recognize tree species in urban environments. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5, 2016. doi: 10.1109/WHISPERS.2016.8071756.

[68] MultiSpec. Hyperspectral images. `https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html`, 2020. [Last read on: 2025-05-10].

[69] D. J. Musiał. Timeliness and frequency of sentinel products explained. `https://creodias.eu/cases/timeliness-and-frequency-of-sentinel-satellite-products-explained/`, 2024. [Last read on: 2025-05-10].

[70] NASA. The Advanced Spaceborne Thermal Emission and Reflection Radiometer , 2004. URL `https://asterweb.jpl.nasa.gov/`. [Last read on: 2025-05-10].

[71] NASA. AVIRIS Data - Ordering Free AVIRIS Standard Data Products, 2015. URL `https://aviris.jpl.nasa.gov/data/free_data.html`. [Last read on: 2025-05-10].

[72] J. Nascimento and J. Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005. doi: 10.1109/TGRS.2005.844293.

[73] M. Nikzadfar, M. Rashvand, H. Zhang, A. Shenfield, F. Genovese, G. Altieri, A. Matera, I. Tornese, S. Laveglia, G. Paterna, C. Lovallo, O. Mammadov, B. Aykanat, and G. C. Di Renzo. Hyperspectral imaging aiding artificial intelligence: A reliable approach for food qualification and safety. *Applied Sciences*, 14(21), 2024. ISSN 2076-3417. doi: 10.3390/app14219821. URL `https://www.mdpi.com/2076-3417/14/21/9821`.

[74] N. U. of Science and Technology. HYPERSPECTRAL DATA

SET. URL http://lesun.weebly.com/hyperspectral-data-set.html. [Last read on: 2025-05-10].

[75] N. U. of Science and Technology. Hyperspectral datasets. http://lesun.weebly.com/hyperspectral-data-set.html, 2024. [Last read on: 2025-05-10].

[76] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson. Convolutional autoencoder for spectral-spatial hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–15, 2020. doi: 10.1109/TGRS.2020.2992743.

[77] J. Peng, W. Sun, F. Jiang, H. Chen, Y. Zhou, and Q. Du. A general loss-based nonnegative matrix factorization for hyperspectral unmixing. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020. doi: 10.1109/LGRS.2020.3017233.

[78] B. L. N. Phaneendra Kumar, R. Vaddi, P. Manoharan, L. Agilandeeswari, and V. Sangeetha. A new band selection framework for hyperspectral remote sensing image classification. *Scientific Reports*, 14(1):31836, Dec 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-83118-8. URL https://doi.org/10.1038/s41598-024-83118-8.

[79] M. T. Physics. Blackbody radiation. http://www.alevelphysicsnotes.com/astrophysics/black_body_rad.php, 2024. [Last read on: 2025-05-10].

[80] S. Prasad, B. Le Saux, N. Yokoya, and R. Hansch. 2018 ieee grss data fusion challenge – fusion of multispectral lidar and hyperspectral data. 2020. doi: 10.21227/jnh9-nz89. URL https://dx.doi.org/10.21227/jnh9-nz89.

[81] L. Qi, J. Li, Y. Wang, Y. Huang, and X. Gao. Spectral–spatial-weighted multiview collaborative sparse unmixing for hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 58 (12):8766–8779, 2020. doi: 10.1109/TGRS.2020.2990476.

[82] J. Qin, K. Chao, M. S. Kim, R. Lu, and T. F. Burks. Hyperspectral and multispectral imaging for evaluating food safety and quality. *Journal of Food Engineering*, 118(2):157–171, 2013. ISSN 0260-8774. doi: https://doi.org/10.1016/j.jfoodeng.2013.04.001. URL https://www.sciencedirect.com/science/article/pii/S0260877413001659.

[83] G. Ren, Y. Wang, J. Ning, and Z. Zhang. Using near-

infrared hyperspectral imaging with multiple decision tree methods to delineate black tea quality. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 237:118407, 2020. ISSN 1386-1425. doi: https://doi.org/10.1016/j.saa.2020.118407. URL https://www.sciencedirect.com/science/article/pii/S1386142520303851.

[84] H. Roelofsen, P. Bodegom, L. Kooistra, and J.-P. Witte. Trait estimation in herbaceous plant assemblages from in situ canopy spectra. *Remote Sensing*, 5:6323–6345, 11 2013. doi: 10.3390/rs5126323.

[85] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

[86] 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*, 158:207–219, 2015. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2014.11.014. URL https://www.sciencedirect.com/science/article/pii/S0034425714004568.

[87] B. Shekarau Luka, B. Mohammed Yunusa, Q. Msurshima Vihikwagh, K. Fanan Kuhwa, T. Hannah Oluwasegun, R. Ogalagu, T. Kenneth Yuguda, and M. Adnouni. Hyperspectral imaging systems for rapid assessment of moisture and chromaticity of foods undergoing drying: Principles, applications, challenges, and future trends. *Computers and Electronics in Agriculture*, 224:109101, 2024. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2024.109101. URL https://www.sciencedirect.com/science/article/pii/S0168169924004927.

[88] Y. Shi, L. Han, L. Han, S. Chang, T. Hu, and D. Dancey. 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, 2022. doi: 10.1109/TGRS.2022.3193441.

[89] N. G. Solutions. Envi image files. https://www.nv5geospatialsoftware.com/docs/ENVIImageFiles.html, 2025. [Last read on: 2025-05-10].

[90] S. I. L. Specim. Specim afx10. https://www.specim.com/products/specim-afx10/, 2024. [Last read on: 2025-05-10].

[91] S. I. L. Specim. What is hyperspectral imaging: A comprehensive guide. https://www.specim.com/technology/what-is-hyperspectral-imaging/, 2024. [Last read on: 2025-05-10].

[92] H. Su, C. Jia, P. Zheng, and Q. Du. Superpixel-based weighted collaborative sparse regression and reweighted low-rank representation for hyperspectral image unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:393–408, 2022. doi: 10.1109/JSTARS.2021.3133428.

[93] 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*, 57(7):4309–4321, 2019. doi: 10.1109/TGRS.2018.2890633.

[94] Y. Su, X. Xu, J. Li, H. Qi, P. Gamba, and A. Plaza. Deep autoencoders with multitask learning for bilinear hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10): 8615–8629, 2021. doi: 10.1109/TGRS.2020.3041157.

[95] W. Sun and Q. Du. Hyperspectral band selection: A review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2):118–139, 2019. doi: 10.1109/MGRS.2019.2911100.

[96] M. A. Veganzones and M. Graña. Endmember extraction methods: A short review. In I. Lovrek, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 400–407, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-85567-5.

[97] J.-J. Wang, D.-C. Wang, T.-Z. Huang, and J. Huang. Endmember constraint non-negative tensor factorization via total variation for hyperspectral unmixing. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 3313–3316, 2021. doi: 10.1109/IGARSS47720.2021.9554468.

[98] X. Wang, Y. Zhong, L. Zhang, and Y. Xu. Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral

unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 55 (11):6287–6304, 2017. doi: 10.1109/TGRS.2017.2724944.

[99] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du. Cva2e: A conditional variational autoencoder with an adversarial training process for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(8):5676–5692, 2020. doi: 10.1109/TGRS.2020.2968304.

[100] Z. Xin, S. Jun, W. Xiaohong, L. Bing, Y. Ning, and D. Chunxia. Research on moldy tea feature classification based on wknn algorithm and nir hyperspectral imaging. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 206:378–383, 2019. ISSN 1386-1425. doi: https://doi.org/10.1016/j.saa.2018.07.049. URL https://www.sciencedirect.com/science/article/pii/S1386142518307030.

[101] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 7(2):365–384, Apr 2012. ISSN 1673-3576. doi: 10.1007/s11464-012-0194-5. URL http://dx.doi.org/10.1007/s11464-012-0194-5.

[102] W. Xuan and Y. Wang. Competitive adaptive reweighted sampling method for fault detection. *Journal of Physics: Conference Series*, 1820:012078, 03 2021. doi: 10.1088/1742-6596/1820/1/012078.

[103] B. Xue, S.-Y. Chen, C. Yu, Y. Wang, L. Wang, M. Song, S. Li, and C.-I. Chang. Kernel automatic target generation process. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 636–639, 2017. doi: 10.1109/IGARSS.2017.8127034.

[104] N. Yokoya, T. Yairi, and A. Iwasaki. Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2): 528–537, 2012. doi: 10.1109/TGRS.2011.2161320.

[105] M. Z6196219hao, J. Chen, and Z. He. A laboratory-created dataset with ground-truth for hyperspectral unmixing evaluation. *CoRR*, abs/1902.08347, 2019. URL http://arxiv.org/abs/1902.08347.

[106] Z. Zaman, S. B. Ahmed, and M. I. Malik. Analysis of hyperspectral data to develop an approach for document images. *Sensors*, 23 (15), 2023. ISSN 1424-8220. doi: 10.3390/s23156845. URL https:

//www.mdpi.com/1424-8220/23/15/6845.

[107] J. Zhang, Z. Cai, F. Chen, and D. Zeng. Hyperspectral image denoising via adversarial learning. *Remote Sensing*, 14(8), 2022. ISSN 2072-4292. doi: 10.3390/rs14081790. URL https://www.mdpi.com/2072-4292/14/8/1790.

[108] S. Zhang, J. Li, H. Li, C. Deng, and A. Plaza. Spectral–spatial weighted sparse regression for hyperspectral image unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3265–3276, 2018. doi: 10.1109/TGRS.2018.2797200.

[109] W. Zhang, A. Cao, P. Shi, and L. Cai. Rapid evaluation of freshness of largemouth bass under different thawing methods using hyperspectral imaging. *Food Control*, 125:108023, 2021. ISSN 0956-7135. doi: https://doi.org/10.1016/j.foodcont.2021.108023. URL https://www.sciencedirect.com/science/article/pii/S0956713521001614.

[110] X. Zhang, X.-h. Tong, and M.-l. Liu. An improved n-findr algorithm for endmember extraction in hyperspectral imagery. In *2009 Joint Urban Remote Sensing Event*, pages 1–5, 2009. doi: 10.1109/URS.2009.5137677.

[111] M. Zhao, L. Yan, and J. Chen. Lstm-dnn based autoencoder network for nonlinear hyperspectral image unmixing. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):295–309, 2021. doi: 10.1109/JSTSP.2021.3052361.

[112] Z. Zhao, H. Wang, Y. Liang, T. Huang, Y. Xiao, and X. Yu. Sparsity constrained convolutional autoencoder network for hyperspectral image unmixing. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 3317–3320, 2021. doi: 10.1109/IGARSS47720.2021.9553239.

[113] X. Zheng, J. Jia, J. Chen, S. Guo, L. Sun, C. Zhou, and Y. Wang. Hyperspectral image classification with imbalanced data based on semi-supervised learning. *Applied Sciences*, 12(8), 2022. ISSN 2076-3417. doi: 10.3390/app12083943. URL https://www.mdpi.com/2076-3417/12/8/3943.

[114] F. Zhou, R. Hang, Q. Liu, and X. Yuan. Hyperspectral image classification using spectral-spatial lstms. *Neurocomputing*, 328:39–47, 2019. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.02.105. URL

https://www.sciencedirect.com/science/article/pii/S0925231218309573. Chinese Conference on Computer Vision 2017.

[115] F. Zhu. Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey, 2017. [Last read on: 2025-05-10].

[116] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan. Spectral unmixing via data-guided sparsity. *IEEE Transactions on Image Processing*, 23(12):5412–5427, Dec 2014. ISSN 1941-0042. doi: 10.1109/tip.2014.2363423. URL http://dx.doi.org/10.1109/TIP.2014.2363423.

[117] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan. Structured sparse method for hyperspectral unmixing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:101–118, 2014. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2013.11.014. URL https://www.sciencedirect.com/science/article/pii/S0924271613002761.

[118] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson. Generative adversarial networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5046–5063, 2018. doi: 10.1109/TGRS.2018.2805286.

A. SpectralFormer Model Architecture

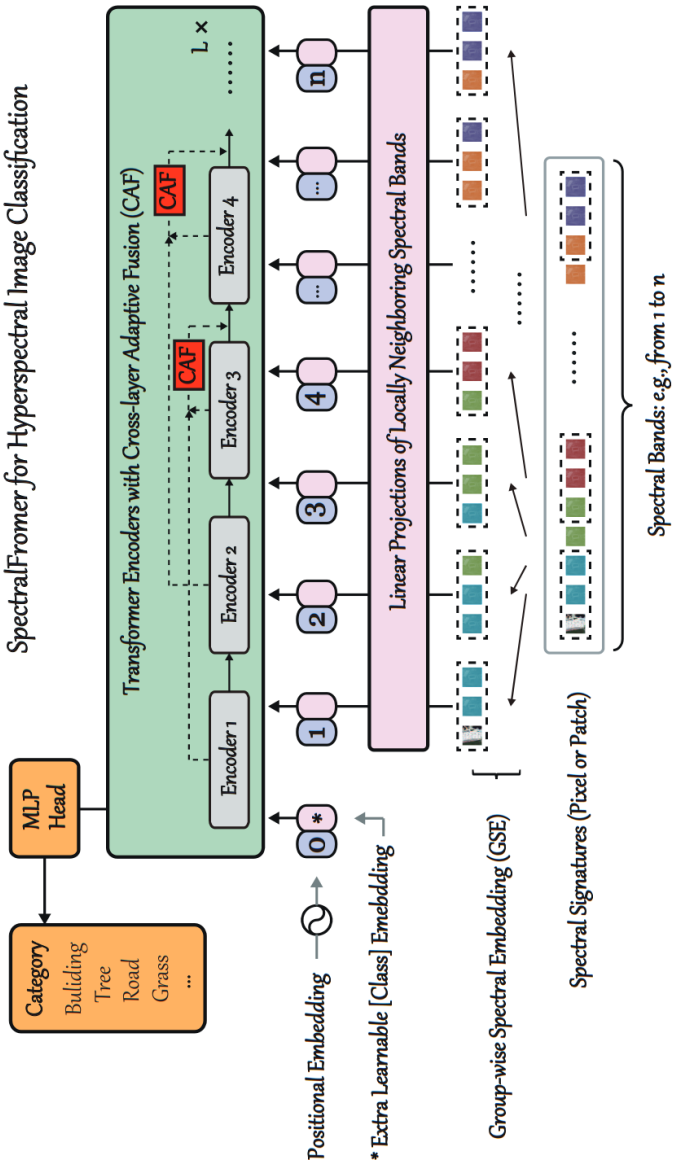


Figure 16 SpectralFormer transformer-based deep neural network model architecture. Source: [38]

B. Convolutional Neural Network (CNN) Model

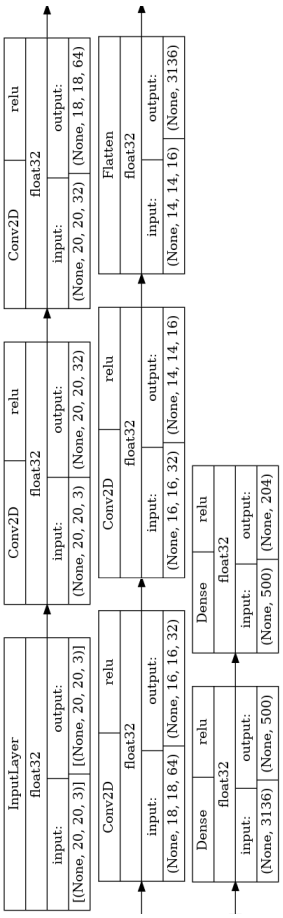


Figure 17 Convolutional Neural Network (CNN)-based deep neural network model architecture for hyperspectral data reconstruction from an RGB image. A newly proposed model specifically for hyperspectral data reconstruction based on the research of Wang et al. [99].

C. Hyperspectral Data to RGB Generation

Collecting CIE 1931 XYZ values and the required wavelength indices.

Link to the function on GitHub

https://github.com/VytautasPau/UAVHyperspectral-ext/blob/master/cie.py

A function to integrate the XYZ data over hyperspectral data to create the three XYZ bands from multiple hyperspectral bands. The

function includes optional *MinMax* normalisation, and the *numpy.trapz* function integrates hyperspectral data over CIE 1931 XYZ colour curves. After extracting XYZ data, it is converted to RGB. Link to the function on GitHub

https://github.com/VytautasPau/UAVHyperspectral-ext/blob/master/int.py

D. Hyperspectral Calibration

A *Python* function used to calculate reflectance values for a hyperspectral data cube using the calibration plate information gathered in the field during the flight. The function is based on the *Numpy* matrix computation library and a *polyfit* function that fits a line according to the given parameters – in this case, the raw hyperspectral values and reflectance values. Link to the function on GitHub

https://github.com/VytautasPau/UAVHyperspectral-ext/blob/master/cal.py

E. Cosine Similarity Loss

Link to the function on GitHub

https://github.com/VytautasPau/UAVHyperspectral-ext/blob/master/cos.py

SUMMARY IN LITHUANIAN

HSI yra nuotolinio stebėjimo technologija, kuri renka duomenis visame elektromagnetiniame spektre, kad gautų spektrinius duomenis kiekvienam scenos pikseliui. Tai leidžia nenaikinamuoju būdu rinkti duomenis medžiagų identifikavimui pagal spektrinius parašus, objektų aptikimui ir kitiems su medžiagų analize susijusiems uždaviniams spręsti.

Tyrimų sritis

HSI naudojama kaip priemonė detaliems medžiagų duomenims rinkti įvairiose mokslo srityse, įskaitant geologiją, astronomiją, žemės ūkį, biologiją ir stebėseną. Didėjant HSI populiarumui, kai ji naudojama su bepiločiais orlaiviais (BO), didėja poreikis tobulinti hiperspektrinės analizės algoritmus. Todėl dalis šio tyrimo yra skirta žemės ūkio hiperspektrinių duomenų analizei, surinktai naudojant BO, ir sprendimų paieškai esamoms problemoms, tokioms kaip:

- Nepakankamas viešai prieinamų hiperspektrinių duomenų rinkinių, surinktų naudojant BO, kiekis.
- Standartizuoto HU algoritmų testavimo nebuvimas.
- Ribotas tyrimų kiekis apie HU metodus, taikomus hiperspektriniams duomenims, surinktiems BO pagalba.

Siekiant spręsti šias problemas, buvo atliktas tyrimas apie HU algoritmų testavimo metodikas, esamus HU algoritmus, giliųjų neuroninių tinklų modelius, naudojamus HU, ir esamų HU duomenų rinkinių kūrimo metodus. Didžioji darbo dalis skirta hiperspektrinių duomenų analizei ir HU metodų, naudojamų spektrinei ir medžiagų informacijai iš HSI išgauti, tyrimui.

Tyrimo problema

Didėjant HSI populiarumui ir platesniam jos naudojimui [10], šios srities tyrimų apimtis taip pat auga. Viena iš tyrimų krypčių HSI srityje yra HU [12]. HU apjungia vieną arba kelis algoritmus, kurie išgauna medžiagų informaciją iš atskirų pikselių hiperspektriniame vaizde. HU yra būtina, nes kiekvieno pikselio spektriniai duomenys gali būti

sudaryti iš kelių medžiagų mišinio. Ši problema kyla dėl mažesnės hiperspektrinių jutiklių surinktų duomenų erdvinės raiškos ir skirtingų medžiagų šviesos sklaidos.

Pagrindinė problema, nagrinėjama šioje disertacijoje, yra HU algoritmo kūrimas, skirtas naudoti su hiperspektriniais duomenimis, surinktais naudojant BO. Tyrimas buvo atliktas siekiant sukurti tikslesnį HU algoritmą, naudojant giliuosius neuroninius tinklus, kadangi esami algoritmai, skirti žemės ūkio lauko duomenų analizei, yra nepakankami.

Darbo aktualumas

Didėjant HSI ir BO populiarumui žemės ūkyje pasėlių stebėsenai ir analizei, šis tyrimas plečia esamus hiperspektrinės analizės metodus. Jis išplečia HU algoritmų taikymą hiperspektriniams duomenims, surinktiems BO pagalba žemės ūkio srityje. Be to, sukuriama standartizuota HU algoritmų testavimo metodika, kuri leidžia tiksliau įvertinti naujai sukurtus HU algoritmus.

Tyrimo objektas

Šios disertacijos tyrimo objektas yra tiek klasikiniai, tiek modernūs giliaisiais neuroniniais tinklais paremti HU algoritmai ir jų taikymas BO surinktiems hiperspektriniams duomenims.

Tyrimo tikslas ir uždaviniai

Pagrindinis šios disertacijos tikslas – pasiūlyti tikslesnį, efektyvesnį ir triukšmui atsparesnį HU algoritmą, pagrįstą giliuoju konvoliuciniu neuroniniu tinklu, skirtą duomenims, surinktiems naudojant BO.

Siekiant šio tikslo, buvo suformuluoti šie uždaviniai:

- Įvertinti esamus HU algoritmus įvairiuose duomenų rinkiniuose, siekiant nustatyti algoritmų tikslumo priklausomybę nuo baltojo triukšmo, vaizdo mastelio ir komponentų (endmembers) skaičiaus.
- Analizuoti hiperspektrinių juostų koreliacijas ir giluminio mokymosi algoritmus, skirtus sintetinių hiperspektrinių duomenų generavimui iš multispektrinių vaizdų.

- Pasiūlyti modifikuotą U-Net modelį HU uždaviniui ir palyginti jį su transformatoriniais HU modeliais, naudojant BO surinktus hiperspektrinius duomenis.
- Optimizuoti siūlomą modifikuotą U-Net HU modelį, siekiant geresnio BO surinkto hiperspektrinio vaizdo tikrosios informacijos (ground truth) atstatymo.

Tyrimo metodai

Buvo taikyti keli tyrimo metodai siekiant įgyvendinti suformuluotus uždavinius. Pirmiausia buvo atlikta susijusių darbų analizė hiperspektrinių duomenų analizės srityje, ypač susijusi su HU algoritmais. Antra, buvo analizuojami tiek prieinami hiperspektriniai duomenų rinkiniai, tiek jų tipai, naudojami HU. Trečia, buvo atliktas tyrimas apie duomenų rinkinių kūrimo metodus, įskaitant hiperspektrinių duomenų rinkimą naudojant BO, siekiant sukurti HU duomenų rinkinį. Ketvirta, buvo atliktas eksperimentas su keliais algoritmais, įskaitant naujai sukurtą modelį, ir su keliais skirtingais HU duomenų rinkiniais. Galiausiai, remiantis eksperimentiniais rezultatais, buvo modifikuota modelio architektūra, siekiant pagerinti HU modelio tikslumą hiperspektriniuose duomenyse, surinktuose naudojant BO.

Mokslinis darbo naujumas

Tyrimai apie HU algoritmus, sutelkiant dėmesį į giliojo mokymosi neuroninių tinklų modelių kūrimą, leidžia pagerinti HU veikimą. Be to, nesupervizuoto giliojo mokymosi metodo sukūrimas leidžia naudoti hiperspektrinius duomenis mišinių išskaidymo (unmixing) uždaviniams spręsti. Paskelbtas HU duomenų rinkinys, pagrįstas žemės ūkio hiperspektriniais duomenimis, leidžia testuoti naujai sukurtus HU algoritmus įvairiuose duomenų rinkiniuose. Taip pat buvo sukurtas naujas HU algoritmas, skirtas žemės ūkio hiperspektrinių duomenų, surinktų naudojant BO, analizei, ir paskelbtas HU etalonas (benchmark), kurio tikslas – pateikti standartizuotą HU algoritmų palyginimo metodą.

Praktinė darbo vertė

Remiantis atliktais tyrimais ir eksperimentais, galima išskirti šiuos pagrindinius praktinius indėlius:

1. Sukurta HU testavimo metodika, skirta naudoti bet kurio naujai sukurto HU algoritmo testavimui.

2. Sukurtas naujas hiperspektrinis duomenų rinkinys iš lauko duomenų, skirtas HU arba klasifikavimo algoritmų testavimui.

3. Pasiūlytas naujas nesupervizuotas HU algoritmas, pagrįstas U-Net architektūra, kuris pasižymi geresniu veikimu nei transformatorinis metodas, taikant HU žemės ūkio hiperspektriniams duomenims, surinktiems naudojant BO.

4. Pasiūlytas nesupervizuotas HU algoritmas, kuris palengvina hiperspektrinę analizę, nes modeliui treniruoti ir naudoti nereikia žinomų atsakymų (ground truths).

Ginamieji teiginiai

Remiantis atliktu tyrimu, buvo suformuluoti šie pagrindiniai teiginiai, ginami disertacijoje:

1. Klasikinių retųjų regresijos pagrindu veikiančių Hyperspectral Unmixing (HU) algoritmų veikimas nepriklauso nuo komponentų (endmembers) skaičiaus ir informacijos tankio, kurį reguliuoja vaizdo mastelis.

2. Klasikiniai retųjų regresijos ir HU algoritmai yra atsparesni triukšmui, turinčiam Gauso pasiskirstymą, tačiau mažiau atsparūs realiam hiperspektrinės kameros triukšmui.

3. Artimos hiperspektrinės juostos pasižymi didele tarpusavio koreliacija, ir šis ryšys gali būti išnaudojamas geresnių sintetinių multispektrinių duomenų generavimui.

4. Šiame tyrime pasiūlyta modifikuota U-Net tipo modelio architektūra reikalauja mažiau hiperparametrų nei testuoti transformatoriniai HU modeliai, kad būtų pasiekti tiksliausi mišinių išskaidymo rezultatai naudojant BO surinktus hiperspektrinius duomenis.

5. Pasiūlytas modifikuotas U-Net HU modelis turi potencialą pasiekti mažesnę rekonstrukcijos paklaidą nei transformatoriniai mišinių išskaidymo algoritmai, kai jie taikomi BO surinktiems hiperspektriniams duomenims.

Tyrimo aprobavimas ir publikavimas

Publikacijos CA WoS žurnaluose:

[A.1] Paura, Vytautas & Marcinkevičius, Virginijus. Hyperspectral Unmixing of Agricultural Images Taken From UAV Using Adapted U-Net Architecture. Straipsnis priimtas į *Baltic Journal of Modern Computing*.

[A.2] Paura, Vytautas & Marcinkevičius, Virginijus. Benchmark for Hyperspectral Unmixing Algorithm Evaluation. *Informatica* (Vilniaus universiteto leidykla, 2023). **34**, 285-315 (2024). https://doi.org/10.15388/23-INFOR522

Publikacijos be citavimo indekso:

[B.1] Paura, Vytautas & Marcinkevičius, Virginijus. Crop Hyperspectral Dataset Unmixing Using Modified U-Net Model. *Digital Business and Intelligent Systems*. **2157**, 195-210 (2024). https://doi.org/10.1007/978-3-031-63543-4_13

[B.2] Paura, Vytautas & Marcinkevičius, Virginijus. Semi-supervised hyperspectral unmixing dataset creation methods for unmixing algorithm analysis. *Image and Signal Processing for Remote Sensing XXIX* (SPIE, 2023). **12733**, 127330Y (2023). https://doi.org/10.1117/12.2679826

[B.3] Paura, Vytautas & Marcinkevičius, Virginijus. The Analysis of Impact of Noise on Hyperspectral Unmixing Algorithms. *Data Analysis Methods for Software Systems*. Bernatavičienė, Jolita. (Vilniaus universiteto leidykla, 2021). **17**, 56 (2021). https://doi.org/10.15388/DAMSS.12.2021

Konferencijos ir pristatymai:

[C.1] Paura, Vytautas & Marcinkevičius, Virginijus. Crop Hyperspectral Dataset Unmixing Using Modified U-Net Model. *Digital Business and Intelligent Systems*. Vilnius, Lietuva (2024).

[C.2] Paura, Vytautas & Marcinkevičius, Virginijus. Semi-Supervised Hyperspectral Unmixing Dataset Creation Methods For Unmixing Algorithm Analysis. *Image and Signal Processing for Remote Sensing XXIX*. Amsterdamas, Olandija (2023).

[C.3] Paura, Vytautas & Marcinkevičius, Virginijus. Hyperspectral data synthesis using deep neural networks. *Eastern European Machine Learning Summer School*. Vilnius, Lietuva (2022).

[C.4] Paura, Vytautas & Marcinkevičius, Virginijus. The Analysis of Impact of Noise on Hyperspectral Unmixing Algorithms. *Data Analysis Methods for Software Systems*. Druskininkai, Lietuva (2021).

MEDŽIAGŲ IR JŲ KIEKIO NUSTATYMO HIPERSPEKTRINIUOSE DUOMENYSE ALGORITMAI

Šiame skyriuje aprašomi hiperspektriniai duomenų rinkiniai, dažnai naudojami susijusiuose HU algoritmų straipsniuose, taip pat hiperspektrinių duomenų surinkimo procesas. Naudojantis šiuo procesu sukurtas HU duomenų rinkinys. HU algoritmų vertinimui pasitelktos įvairios metrikos. Siūlomas U-Net modeliu paremtas HU dirbtinio neuroninio tinklo modelis. Skyrius pagrįstas publikuotais straipsniais [A.1] ir [B.1].

S.1. Naudojamos metrikos

Šioje dalyje aprašomos įvairios metrikos, naudojamos vertinti HU algoritmų veikimą ir mokymo tikslumą.

S.1.1. Bendros metrikos

HU uždaviniuose naudojama keletas skirtingų metrikų. Dažniausiai pasitelkiamos Root Mean Squared Error (RMSE), Signal Reconstruction Error (SRE), Spectral Angle Distance (SAD) ir Spectral Angle Mapping (SAM). RMSE ir Signal Reconstruction Error (SRE) metrikos. Jos buvo pasirinktos dėl jų populiarumo vertinant HU algoritmų našumą, taip pat dėl bendro paprastumo apibūdinant skirtumus tarp įvertintų ir realių spektrų:

- RMSE (lygtis S.1) parodo skirtumą tarp prognozuojamų spektrų ir tikrų duomenų. Skirtingi autoriai naudojo keletą RMSE variantų, kad išbandytų skirtingus sukurtų algoritmų aspektus. Tai apima RMSE vidurkį tarp visų galutinių narių, rekonstrukcijos RMSE ir gausos RMSE. Vidutinis RMSE apskaičiuojamas įvertinant RMSE kiekvienam galutiniam nariui atskirai ir apskaičiuojant vidutinę reikšmę. Rekonstrukcijos RMSE apskaičiuojamas tarp tikrojo signalo ir signalo, prognozuoto modeliu. Gausos RMSE skaičiuojamas tarp tikrųjų duomenų gausos ir prognozuotų. Ši metrika buvo pasirinkta dėl jos plataus naudojimo straipsniuose apie HU algoritmus bei dėl jos universalumo, pritaikomo įvairiems klaidų skaičiavimams.
- SRE (lygtis S.2) naudojama algoritmų generuojamo spektrinio mišinio rekonstrukcijos kokybei nustatyti. Didesnė SRE reikšmė

reiškia geresnę rekonstrukcijos kokybę. Ši metrika buvo pasirinkta kaip metrikos RE, specialiai naudojamos signalų duomenims, modifikacija.

Metrikos apskaičiuojamos naudojant šias formules:

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

, kur N – testuojamos vektoriaus reikšmių skaičius, x_i – i-oji tikroji reikšmė ir $\hat{x}_i$ – i-oji prognozuojama reikšmė.

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

, kur x – tikroji reikšmė, $\hat{x}$ – prognozuojama reikšmė, E – viduje esančių reikšmių vidurkis.

S.1.2. Neuroninio tinklo algoritmo mokymo ir vertinimo metrika

Šie rodikliai ir nuostoliai buvo pasirinkti mokant ir vertinant siūlomą dirbtinio neuroninio tinklo modelį, pagrįstą U-Net architektūra.

$$L_{RE}(I, \widehat{I}) = \frac{1}{W \cdot H}\sum_{i=1}^{H}\sum_{j=1}^{W}(\widehat{I}_{ij} - I_{ij})^2 \tag{S.3}$$

Reconstruction Error (RE) (Lygtis S.3) – matuoja vidutinį skirtumą tarp modelio sugeneruotų ir tikslių tikrų duomenų. Čia W – vaizdo plotis, H – vaizdo aukštis, $\widehat{I}_{ij}$ – prognozuojama spektrinė informacija pikselyje ij, I_{ij} – tikroji spektrinė informacija pikselyje ij. Skaičiavimas atliekamas kiekvienam pikseliui vaizde atskirai.

$$L_{SAD}(I, \widehat{I}) = \frac{1}{R}\sum_{i=1}^{R}\arccos\left(\frac{\left\langle I_i, \widehat{I}_i\right\rangle}{\|I_i\|_2\|\widehat{I}_i\|_2}\right) \tag{S.4}$$

Spectral Angle Distance (SAD) (Lygtis S.4) – matuoja kampus tarp dviejų vektorių daugiamatėje erdvėje. Čia R – pikselių skaičius, I_i – tikrieji duomenys, $\widehat{I}_i$ – prognozuojami duomenys. Naudojant šią formulę hiperspektriniams duomenims, dvi hiperspektrinio kubo erdvinės dimensijos sutraukiamos į vieną dydžio R dimensiją ir kampas skaičiuojamas kiekvienam pikseliui atskirai.

$$\cos(\theta) = \frac{I \cdot \widehat{I}}{\|I\| \left\|\widehat{I}\right\|} \tag{S.5}$$

Kosinuso panašumas (lygtis S.5) – apskaičiuoja vektorių taškinę sandaugą, padalytą iš jų ilgių sandaugos. Čia I ir $\widehat{I}$ – du matuojami įvesties vektoriai, hiperspektrinių duomenų atveju vektoriai yra kiekvieno pikselio spektrinės reikšmės.

S.2. Hiperspektrinių duomenų rinkimo metodika

Šiame skyriuje aprašoma duomenų rinkimo metodika, taikoma žemės ūkio hiperspektriniams duomenims rinkti. Surinkti hiperspektriniai duomenys panaudojami hiperspektrinių duomenų rinkiniui sukurti ir HU algoritmams testuoti.

S.2.1. Hiperspektrinių duomenų rinkimo metodika taikant BO

Neapdoroti hiperspektriniai duomenys buvo surinkti naudojant *Aurelia* X6 droną [1] su *SPECIM* hiperspektrine kamera [90], skraidant virš šilauogių lauko. Neapdorotus hiperspektrinius duomenis renka jutiklis ir įrašo į kameros gamintojo sukurto formato failą. Neapdorotus hiperspektrinius duomenis apdoroja fotoaparato gamintojo pateikta programinė įranga, sukuriamas hiperspektrinių duomenų kubas ENVI rastriniu [89] formatu, kurį galima nuskaityti kita programine įranga. Duomenys iš šių specifinių šilauogių laukų buvo pasirinkti todėl, kad jie yra aukščiausios kokybės iš visų surinktų hiperspektrinių duomenų, taip pat dėl minimalaus šešėlių kiekio hiperspektrinių duomenų kubuose.

Išilgai kelio skenuojanti (angl. *pushbroom*) hiperspektrinė kamera fiksuoja visų (šios duomenų rinkimo misijos) 224 spektrinių juostų duomenis 1024 pikselių pločio linijomis. Galutinis surinktas hiperspektrinio kubo dydis priklauso nuo skrydžio ilgio, bet visada bus panašaus dydžio – 1024 pikselių pločio ir surinktų 224 spektrinių juostų. Šis duomenų rinkimo skrydis buvo atliktas 70 metrų virš žemės, o reikiamas drono greitis buvo apskaičiuotas pagal aukštį, kad pikseliai išliktų kvadratiniai. Galutinis surinktų BO hiperspektrinių kubelių pikselių dydis yra 5 x 5 cm. Kad duomenys būtų kuo tikslesni, jie turi būti registruojami tiesiomis skrydžio trajektorijomis, vadinamomis skrydžio

linijomis. Kiekviena skrydžio linija sukuria atskirą 1024 * x * 224 dydžio hiperspektrinių duomenų kubą, kur *x* priklauso nuo linijos ilgio ir kameros įrašymo greičio. Nustatytas 6 ms sensoriaus išlaikymas, o kameros kadrų per sekundę (arba linijų per sekundę) skaičius nustatytas į 100.

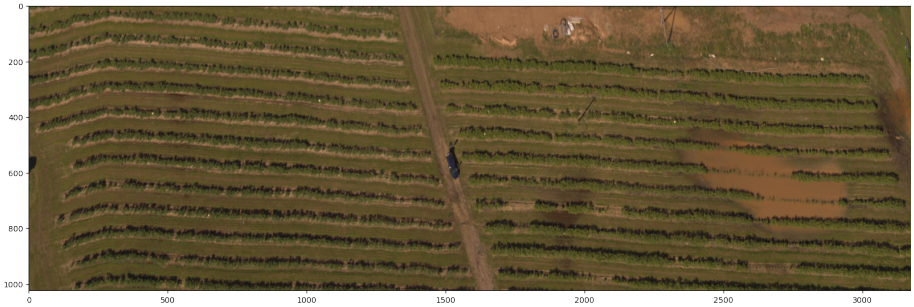
Iš kelių hiperspektrinių duomenų kubų, surinktų BO misijų metu, trys buvo atrinkti kaip HU algoritmams naudojamo duomenų rinkinio pagrindas. Šie duomenų kubai sudaro naujo HU duomenų rinkinio, sukurto atliekant HU algoritmų tyrimą, pagrindą. Sukurtas šilauogių lauko HU duomenų rinkinys, surinktas naudojant BO, buvo pavadintas Blueberry Field HU Dataset (BFHUD). Visi trys BFHUD duomenų kubai turi tą patį medžiagų (angl. *endmember*) rinkinį. Tačiau duomenys buvo renkami tame pačiame šilauogių lauke, skirtingomis dienomis fiksuojant duomenis apie atskiras šilauogių krūmų eilutes, užtikrinant, kad surinkti duomenys nesidubliuotų. Trys duomenų kubai buvo pasirinkti siekiant padidinti duomenų įvairovę ir patikrinti algoritmo atsparumą skirtingoms lauko sąlygoms. Trys duomenų kubai pasižymi šiais parametrais:

- BFHUD 1 kubo forma: 1024 pikselių pločio, 3177 pikselių ilgio su 224 spektrinėmis juostomis;
- BFHUD 2 kubo forma: 1024 pikselių pločio, 3047 pikselių ilgio su 224 spektrinėmis juostomis;
- BFHUD 3 kubo forma: 1024 pikselių pločio, 2815 pikselių ilgio su 224 spektrinėmis juostomis.
- Visų BFHUD kubų spektriniai duomenys tokie patys, jie surinkti nuo 400 iki 1000 nm, o vidutinis atstumas tarp juostų yra maždaug 2,5 nm.

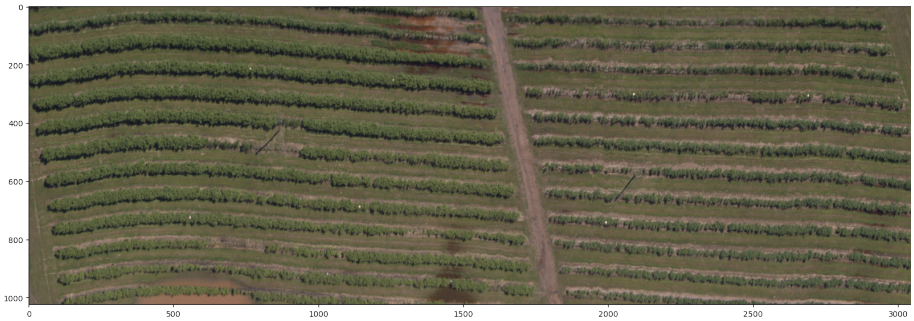
BFHUD kubo RGB atvaizdai, sukurti integruojant duomenis per CIE 1931 XYZ spalvų atitikimo funkcijas ir konvertuojant iš XYZ į RGB, pateikiami žemiau, 2.2 paveiksle. Funkcijos, naudojamos hiperspektriniams duomenims konvertuoti į XYZ spalvų erdvę, kodų sąrašas pateiktas priede C.

S.2.2. Kalibracija

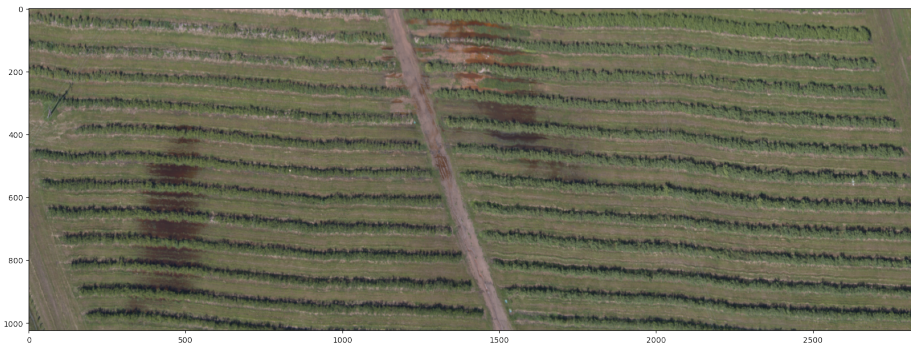
Siekiant duomenis suderinti ir palyginti tarp skrydžių, lauke buvo panaudotas kalibravimo kilimėlių rinkinys. Jo laboratoriškai kalibruotos



(a)



(b)



(c)

S.1 pav. BFHUD kubų RGB vaizdas. BFHUD 1 kubo RGB vaizdas parodytas (a), BFHUD 2 kubo RGB vaizdas parodytas (b), o BFHUD 3 kubo RGB vaizdas parodytas (c).

atspindžio vertės – 5 %, 10 % ir 40 %, duomenų kubas su uždarytu kameros objektyvu, kad būtų galima surinkti visiškai tamsius duomenis arba jutiklio triukšmą. Kalibravimas atliktas naudojant etaloninius / atspindinčius kilimus, padėtus lauke, kai viena iš drono skrydžio linijų kerta kilimus. Pirminė metodika pasitelkta iš J. Burger ir P. Geladi straipsnio [16], aprašyto skyriuje 1.1.4.

S.3. Naujas duomenų rinkinys, skirtas medžiagų nustatymo iš hiperspektrinių vaizdų algoritmams analizuoti

Šiame skyriuje aprašoma metodika, naudojama kuriant HU duomenų rinkinį iš neapdorotų surinktų hiperspektrinių duomenų. Sprendimas sukurti BFHUD buvo priimtas dėl to, kad trūko laisvai prieinamų aukštos kokybės HU duomenų rinkinių, surinktų su BO. Tokiu būdu surinkti duomenys pasižymi daug didesniu žemės skenavimo nuotoliu, palyginti su atvirais palydovo surinktais duomenų rinkiniais, kurie taip pat buvo naudojami, pvz.: Cuprite [75], DC Mall [68] arba Urban [75]. Žemės skenavimo nuotolis apibūdinamas kaip atstumas tarp vaizdo, užfiksuoto iš oro, pikselių centrų; pvz., Urban duomenų rinkinys yra 2 m, o BO surinktų duomenų atstumas – 3 cm.

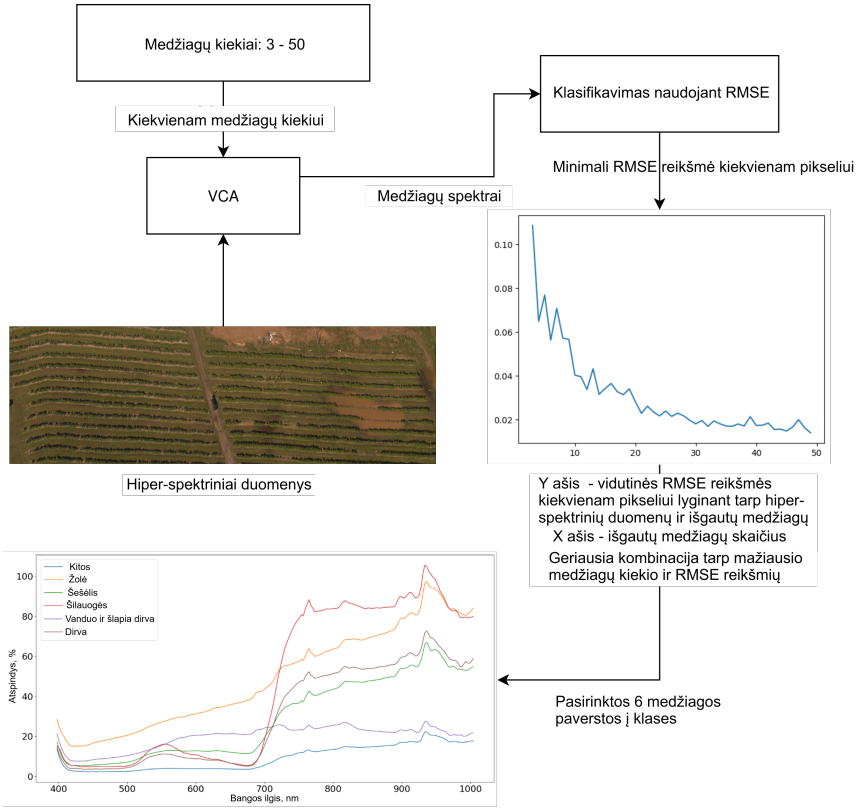
S.3.1. Duomenų klasifikavimas

Didžiausia problema kuriant duomenų rinkinį yra tikslių klasių verčių trūkumas, nes nežinoma, kuriai klasei priklauso kiekvienas pikselis, o rankiniu būdu pažymėti daugiau nei tris milijonus pikselių neįmanoma. Siekiant išspręsti šią problemą buvo apskaičiuotas apytikslis medžiagų ir klasių skaičius. Medžiagoms išgauti pasitelktas specialus metodas, nereikalaujantis klasių informacijos, vadinamas Vertex Component Analysis (VCA) [72] algoritmu. Šis algoritmas bando išgauti galimas medžiagas, atsižvelgiant į hiperspektrinį vaizdą ir medžiagų skaičių. Šis metodas pasirinktas dėl geriausio našumo – optimalaus skaičiavimo greičio ir išgavimo tikslumo derinio – palyginti su dažniau taikomais metodais, tokiais kaip Principal Component Analysis (PCA) [63], Independent Component Analysis (ICA) [40] ir Automatic Target Generation Process (ATGP). Medžiagų išgavimo eksperimentas buvo atliktas naudojant S.2 paveikslėlyje pateiktus veiksmus, siekiant rasti tiksliausią šio duomenų rinkinio medžiagų atvaizdavimą. Medžiagų išgavimo

eksperimentas buvo atliktas šiais veiksmais:

- pasirinktas skaičių diapazonas nuo 3 iki 50;
- kiekvienam skaičiui VCA algoritmas sugeneravo medžiagų rinkinį iš hiperspektrinio duomenų rinkinio. VCA algoritmo išvestis yra medžiagų spektrų rinkinys, kuris vėliau naudojamas galimoms hiperspektrinio vaizdo tikroms klasėms pavaizduoti;
- norint pasirinkti geriausią klasę kiekvienam hiperspektrinio vaizdo pikseliui, skirtumai tarp kiekvieno VCA algoritmų generuojamo galutinio elemento ir hiperspektrinio vaizdo pikselių buvo apskaičiuoti naudojant RMSE. Šiame pavyzdyje sukurta 3177 x 1024 x medžiagų skaičiaus matrica, kai naudojama BFHUD vienaname kube;
- kiekvienam pikseliui medžiaga, turinti mažiausią RMSE reikšmę, buvo nustatyta kaip nauja tikroji klasė. Pavyzdžiui, jei pirmoji medžiaga reiškia dirvožemio klasę, kiekvienas pikselis, kuriame RMSE tarp pikselio ir pirmosios medžiagos buvo mažiausia, bus klasifikuojamas kaip pikselis su dirvožemio klase;
- norint pasirinkti optimaliausią generuojamų medžiagų skaičių, kuris yra mažiausias įmanomas skaičius, pakankamai paaiškinantis hiperspektrinio vaizdo duomenų kitimą, buvo apskaičiuotas visų mažiausių RMSE reikšmių vidurkis visuose hiperspektrinio vaizdo pikseliuose;
- kiekvienam medžiagų skaičiui nuo 3 iki 50 buvo apskaičiuota ir pavaizduota vidutinė mažiausia RMSE reikšmė. Grafike pateiktos nubrėžtos reikšmės.

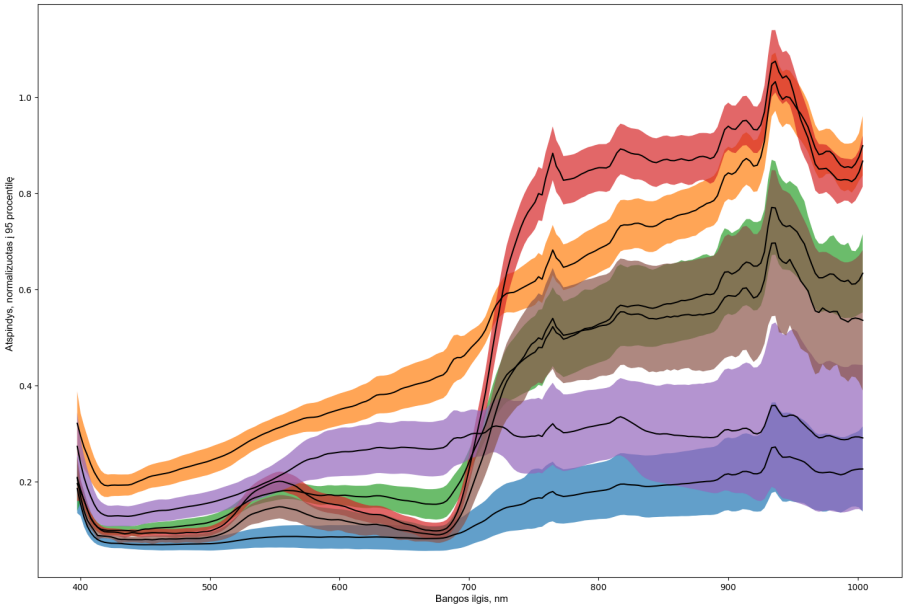
Medžiagų išgavimo eksperimento rezultatas yra klasių skaičius ir tai, kaip tiksliai jie gali apibūdinti pradinį hiperspektrinį vaizdą. Mažesnis bendras RMSE rodo geresnį pirminių duomenų atvaizdavimą. Tuo pačiu metu didesnis klasių skaičius padidina duomenų rinkinio sudėtingumą ir gali sukurti klases, kurios atspindi minimalų faktinių duomenų kiekį. Optimaliausias derinys sujungia mažiausią bendrą RMSE su mažiausiu klasių skaičiumi. Padaryta išvada, kad geriausias klasių skaičius buvo 6 arba 12. Iš pradžių buvo naudojamos abi konfigūracijos.



S.2 pav. Galinių elementų ištraukimo iš hiperspektrinio vaizdo diagrama naudojant VCA algoritmą BFHUD kubo klasifikavimui.

Tačiau, išanalizavus pikselių skaičių kiekvienoje klasėje buvo nustatyta, kad, naudojant 12 klasių, kai kurios klasės turi nedidelį mėginių kiekį (< 0,1 % pikselių), ir tai žymiai padidintų maišymo sunkumą. Todėl tolesniems tyrimams buvo panaudotos šešios išgautos medžiagos ir klasės.

Šešios išgautos medžiagos buvo pasirinktos kaip pagrindinės tiesos klasės BFHUD. Klasė buvo nustatyta kiekvienam pikseliui naudojant mažiausią RMSE reikšmę tarp pikselių spektrų ir kiekvienos medžiagos. Kai vaizdas buvo klasifikuojamas, visi pikselių duomenys buvo renkami kiekvienai klasei, todėl buvo sukurta panašių pikselių kolekcija. Norint pašalinti nukrypimus iš duomenų rinkinio, kiekvienas pikselis, kurio spektrinis pokytis bet kuriose juostose didesnis nei 2 σ, buvo pakeistas atsitiktiniu gretimu pikseliu, kuris atitiko šią spektro kitimo taisyklę. Pašalinus reikšmingiausius nuokrypius buvo sukurtas naujas



S.3 pav. Šešios medžiagos, pavaizduotos skirtingomis spalvomis ir standartiniais nuokrypiais kiekvienoje spektrinėje juostoje BFHUD.

filtruotas hiperspektrinis vaizdas. Paveiksle S.3 pateikta diagrama, kurioje parodytos kiekvienos klasės reikšmės ir jų standartiniai nuokrypiai kiekvienoje spektrinėje juostoje.

Neapdoroti klasių duomenys pateikti paveiksle S.4. Klasės atspindi šilauogių pasėlius, pliką dirvą, žolę, duomenis šešėlyje, vandenį ir visus kitus duomenis, sujungtus į kitą klasę.

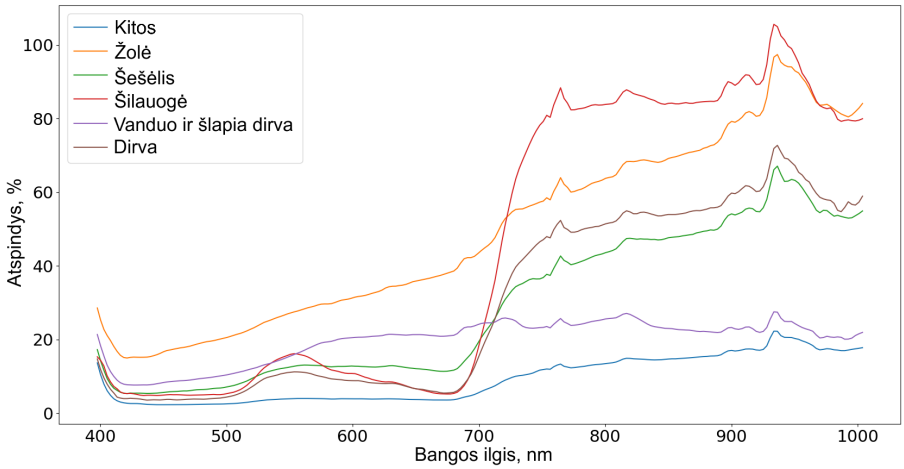
Klasių pasiskirstymas BFHUD 1 kube parodytas paveiksle S.5.

S.3.2. Duomenų rinkinio publikavimas

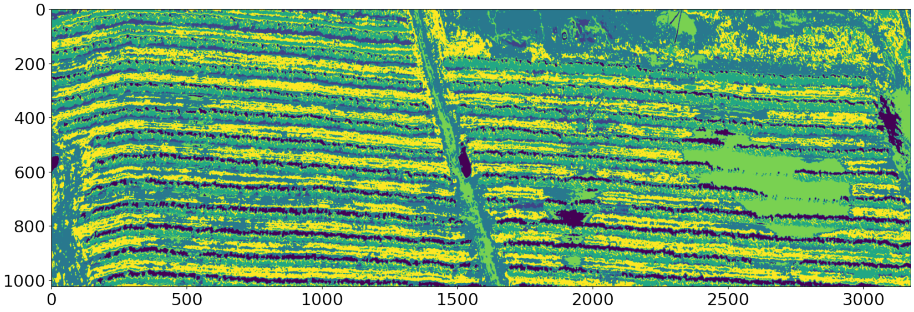
Išsamūs BFHUD ir klasifikuoti duomenys skelbiami kaip atviri duomenys, skirti naudoti kituose eksperimentuose bei hiperspektrinio medžiagų nustatymo ir klasifikavimo užduotyse. Duomenys prieinami atvirosios prieigos Zenodo platformoje: `https://doi.org/10.5281/zenodo.13856357`

S.4. Lyginamosios analizės metodika

Šiame skyriuje aprašoma HU lyginamosios analizės metodika, sukurta bandant standartizuoti HU algoritmo testavimą, nes nėra vieno būdo,



S.4 pav. Išskirtų klasių vidurkiai, naudojami kaip tiksli klasių informacija BFHUD kiekvienai iš šešių klasių.



S.5 pav. Klasių pasiskirstymas BFHUD 1 kube. Spalvų klasės vaizdavimas: geltona – plikas dirvožemis; žalia – šilauogės; mėlyna – žolė; tamsiai mėlyna – šešėliniai duomenys; šviesiai žalia – vanduo ir šlapia dirva; juoda – kiti duomenys.

kaip šie algoritmai buvo tikrinami atliekant literatūros analizę. Siekiant išbandyti skirtingus HU algoritmų aspektus, lyginamosios analizės eksperimentas suskirstytas į keturias pagrindines dalis:

1. Atsparumas medžiagos kiekiui. Taip patikrinama algoritmo galimybė apibendrinti ir bendras jo veikimas, kai pakeičiamas įvesties klasių skaičius. Šio tipo testas leidžia algoritmui patikrinti jo gebėjimą rasti medžiagas ir atkurti hiperspektrinius vaizdus, atsižvelgiant į tiriamos teritorijos sudėtingumą. Dėl kintančio klasių

skaičiaus buvo sukurtas naujas sintetinis duomenų rinkinys, kaip pagrindą pasitelkus IEEE GRSS [80] duomenų derinį, o spektrinei informacijai – USGS spektrinė biblioteka [48].

2. Atsparumas triukšmui. Šis eksperimentas vertina algoritmo gebėjimą tiksliai identifikuoti medžiagas iš hiperspektrinio vaizdo spektrų, kai prie vaizdo pridedamas skirtingas dirbtinio triukšmo lygis. Šis eksperimentas testuoja algoritmus, taikant skirtingo kiekio atsitiktinį triukšmą ir triukšmo profilį, sukurtą pagal realų scenarijų. Duomenų rinkinys, sukurtas medžiagų nustatymui išbandyti, buvo naudojamas kaip pagrindinis hiperspektrinių duomenų rinkinys, prie kurio buvo pridėtas dirbtinio triukšmo sluoksnis.

3. Įvesties vaizdo dydžių skirtumų įtaka. Nustačius skirtingus hiperspektrinių vaizdų dydžius keičiasi erdvinės ir spektrinės informacijos kiekis, kuris veikia bendrą algoritmų veikimą. Tai leidžia nustatyti optimalų vaizdo dydį, užtikrinantį geriausią medžiagų nustatymo tikslumo ir našumo derinį. Tai taip pat rodo duomenų kiekį, reikalingą algoritmams pasiekti geriausią tikslumą. Naudotas tas pats hiperspektrinių duomenų rinkinys, kuris vėliau buvo sumažintas pagal toliau aprašytą metodiką siekiant sukurti skirtingo erdvinio dydžio hiperspektrinius vaizdus.

S.4.1. Eksperimento, tikrinančio algoritmų atsparumą medžiagos kiekio pokyčiui, schema

Algoritmų atsparumo medžiagos kiekio pokyčiui testavimas atliekamas sukuriant dirbtinai sugeneruotų duomenų rinkinių grupę pagal taisyklių rinkinį:

- duomenų rinkiniai D_x (kur x – nustatytas klasių skaičius) sukuriami pasirenkant medžiagas iš USGS spektrinės bibliotekos duomenų;
- pasirinktas klasių skaičius x: nuo 3 iki 10 klasių su 1 žingsniu, nuo 10 iki 30 klasių su žingsniu 5 ir nuo 30 iki 100 klasių su 10 žingsniu;
- kiekvienam D_x viena gausos matrica A_{equal} sukuriama naudojant vienodą visų klasių x gausą;

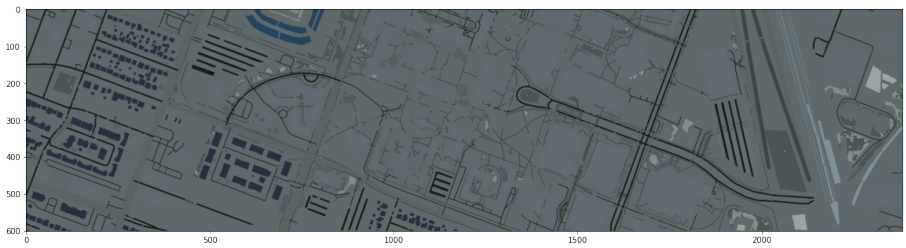
- kiekvienam D_x dešimt gausos matricų A_y sukuriama atsitiktinai generuojant klasių gausą y naudojant vienodą paskirstymą. y yra normalizuotas, kad atitiktų sumos į vienetą apribojimą (lygtis (S.6));

- dirbtinis hiperspektrinis vaizdas I_i (tokio vaizdo RGB atvaizdo pavyzdys parodytas S.6), kurio dydis – 150 x 100 pikselių, sugeneruojamas naudojant gausos matricą A_y ir medžiagas x. Vaizdo dydis pasirinktas taip, kad atspindėtų tikrovišką hiperspektrinį vaizdą, išlaikant jį mažą, kad būtų sumažintas skaičiavimo išteklių naudojimas.

Žemiau pateikta sumos į vienetą apribojimo formulė:

$$\hat{y_i} = \frac{y_i}{\sum_{i=1}^{x} y_i} \tag{S.6}$$

, kur $\hat{y_i}$ – normalizuotos gausos vertės, y_i – modelio sugeneruotos gausos vertės kiekvienai medžiagai, x – medžiagų skaičius.



S.6 pav. Dirbtinis hiperspektrinio vaizdo RGB vaizdavimas.

S.4.2. Triukšmo atsparumo eksperimento schema

Sukurta dirbtinių hiperspektrinių vaizdų kolekcija, skirta patikrinti algoritmo atsparumą triukšmui. Tada prie vaizdų pridėtas skirtingas triukšmo kiekis pagal šias nustatytas taisykles:

- sukurta keturių skirtingų duomenų rinkinių kolekcija su skirtingomis medžiagomis, taikat tokią pačią metodiką kaip ir atsparumo medžiagos kiekiui eksperimente S.4.1 poskyryje;

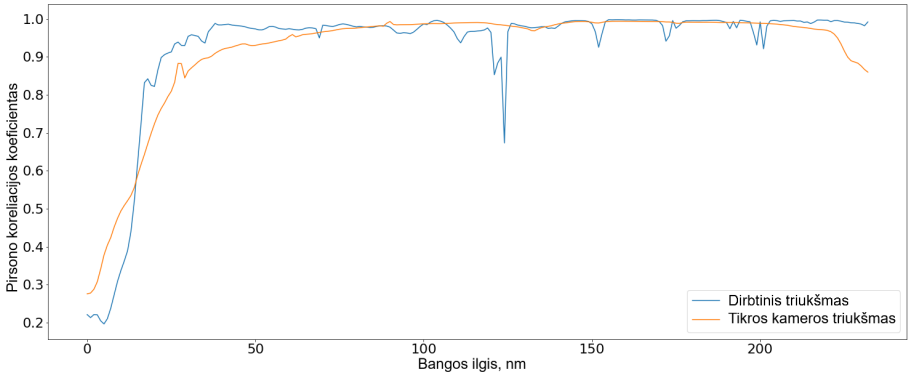
- kiekvienam iš keturių duomenų rinkinių pridedamas skirtingas dirbtinio triukšmo kiekis;

- sukurtas triukšmas matuojamas SNR dB, kur mažesnis skaičius reiškia didesnį baltojo triukšmo kiekį;
- atsitiktinis triukšmas, kurio vidutinė vertė yra 0, generuojamas su norimomis SNR dB reikšmėmis 20, 25, 30, 35, 40, 45 ir 50;
- tada šis triukšmas taikomas kiekvienam iš keturių duomenų rinkinių, kad būtų sukurti triukšmingi vaizdai.

Be atsitiktinio sugeneruoto triukšmo, iš hiperspektrinės vaizdo kameros, naudojamos tyrimams nekontroliuojamoje lauko aplinkoje, buvo išskirtas triukšmo parametrų rinkinys. Kamera buvo BaySpec OCI-F hiperspektrinis sensorius, skenuojantis matomame ir artimųjų infraraudonųjų spindulių elektromagnetiniame diapazone [7]. Siekiant išmatuoti kameros sukuriamą triukšmą kiekviename bangos ilgyje, apskaičiuotas Pirsono koreliacijos koeficientas. Kiekvienas gretimas bangos ilgis paimtas iš hiperspektrinio vaizdo. Apskaičiuota koreliacija tarp šių bangos ilgių verčių visame paveikslėlyje. Paveiksle S.7 (oranžinė linija) parodytas koreliacijos koeficientas bangos ilgio indekse x ir $x-1$. Tiksli Pirsono koreliacija apskaičiuota vienam iš sintetiniu būdu sukurtų hiperspektrinių vaizdų, naudojamų šiame eksperimente. Rezultatai parodyti S.7 paveiksle (mėlyna linija). Pirsono koreliacija tarp gretimų juostų tame pačiame paveiksle buvo apskaičiuota naudojant formulę (S.7), kur *r* – koreliacijos reikšmė, *x* – pirmasis reikšmių rinkinys (šiuo atveju tam tikro bangos ilgio reikšmės), *y* – antrasis reikšmių rinkinys (gretimo bangos ilgio reikšmės). Skaičiavimas atliekamas naudojant reikšmių porą iš to paties pikselio *i*, apskaičiuojant skirtumą nuo vidutinės kiekvieno rinkinio vertės ($\overline{x}$ ir $\overline{y}$), jas padauginant, ir gaunamą sumą padalinus iš šaknies, trauktos iš jų sumos kvadratu.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{S.7}$$

Dirbtinių triukšmo parametrų rinkinys nustatytas taikant gradiento nusileidimo minimizavimo algoritmą. Pirsono koreliacijos koeficientai, taikyti sintetiniu būdu sukurtam hiperspektriniam vaizdui, labai tiksliai atitiko realios kameros triukšmo specifikaciją. Šiuo atveju bangos ilgių skaičiui, siekiant apskaičiuoti skirtingų kintamųjų kiekius, buvo naudojamas daugiamatis optimizavimo algoritmas. Konkrečiau, evoliucinis



S.7 pav. Realaus ir dirbtinai sukurto triukšmo profilio Pirsono koreliacijos palyginimas.

algoritmas, vadinamas diferencine evoliucija iš Python bibliotekos *scipy*. Jis buvo pritaikytas siekiant sumažinti skirtumą tarp tikrosios ir dirbtinės koreliacijos. Paveiksle S.7 pateikti sukurti triukšmo specifikacijos rezultatai. Tada šis triukšmo profilis buvo pasitelktas siekiant sukurti tikroviškesnį kameros triukšmą. Siekta patikrinti, kaip algoritmai veikia pagal šį scenarijų.

S.4.3. Vaizdo dydžio skirtumo įtakos eksperimento metodika

Algoritmo veikimo testavimas pagal skirtingus vaizdo dydžius atliktas tokia seka:

- pasitelkus tikslią atsparumo medžiagos kiekiui metodiką, aprašytą poskyryje S.4.1., buvo sukurtas sintetiniu būdu sugeneruotas hiperspektrinio vaizdo duomenų rinkinys su skirtingu klasių skaičiumi;
- šie duomenų rinkiniai sumažinti naudojant vidutines reikšmes 2 x 2 pikselių srityje iki vienos reikšmės ir 3 x 3 pikselių iki vienos vertės, o žingsnio dydis lygus filtro sričiai, t. y. slankiojantis langas buvo naudojamas be persidengimo. Taip sukurti 4 ir 9 kartus mažesni vaizdai;
- tada RMSE ir SRE metrikos apskaičiuotos pagal minėtus tris duomenų rinkinių rinkinius, siekiant palyginti gautus rezultatus.

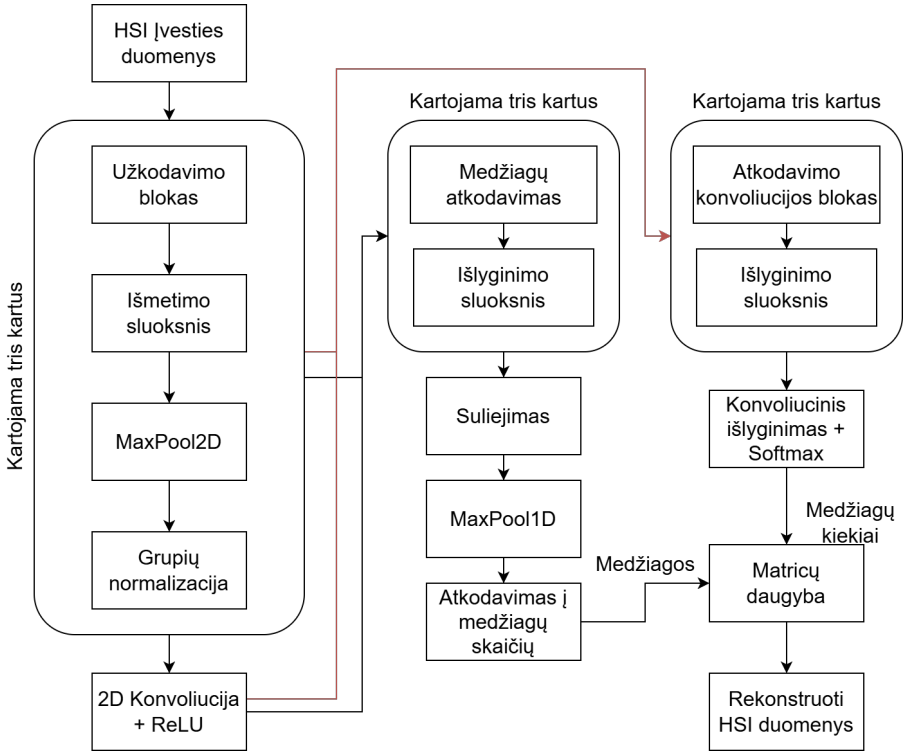
S.5. Siūlomas U-Net pagrįstas neuroninio tinklo modelis

Atliekant esamų medžiagų nustatymo iš hiperspektrinių vaizdų algoritmų tyrimus ir naudojant 1.2.7 skyriuje paminėtą modelį kaip eksperimentų pagrindą, sukurtas naujas medžiagų nustatymo iš hiperspektrinių vaizdų modelis. Remtasi architektūra, pagrįsta U-Net giliosios konvoliucinės automatinio kodavimo modeliu [85]. Naujai sukurtas HU modelis pavadintas Hyperspectral Unmixing U-NET (HUNET).

Transformatoriaus modelis, sukurtas Ghosh ir kt. [28], turėjo keletą trūkumų, išaiškintų atliekant bandymus:

- originalus modelis buvo naudojamas tik kvadratiniams vaizdams;
- visas vaizdas buvo naudojamas kaip įvestis modelyje, kuris nustatė modelio sluoksnių dydį. Didelis įvesties duomenų rinkinys turi būti suspaustas arba transformatoriaus latentinė erdvė turi būti sumažinta, nes pasiekiamos grafikos plokštės atminties ribos;
- kiekvienam darbe išbandytam duomenų rinkiniui buvo pateiktas rankiniu būdu suderintų hiperparametrų rinkinys. Norint naudoti naujus duomenų rinkinius reikia sugeneruoti naują parametrų rinkinį.

Pirmoji problema išspręsta tiesiogiai pradiniame modelio kode, pakeitus architektūrą iš naudojamų kvadratinių vaizdų į stačiakampius vaizdus. Tai padaryta tik duomenų nuskaitymo kodo dalyje, pačiame modelyje pakeitimų neatlikta. Optimizavimas ir modelio patobulinimai gali būti pasiekti naudojant stačiakampius duomenis kaip įvestį transformatoriaus pataisymo algoritmui modelyje. Paleidus šį modelį su BFHUD, buvo atlikta keletas mokymo iteracijų, siekiant rasti parametrų rinkinį, kuris užtikrintų tinkamus rezultatus nepralaužiant grafikos plokštės atminties apribojimų ar nesukeliant kitų klaidų (pvz., įvesties duomenų forma turi būti dalijama iš nurodyto pataisos dydžio). Antroji ir trečioji problema buvo išspręstos rankiniu būdu. Antroji problema, susijusi su per dideliu grafikos plokštės atminties naudojimu, išspręsta perpus sumažinus latentinės erdvės parametrą, palyginti su reikšme, kurią autoriai naudojo DC Mall duomenų rinkiniui. Siekiant išspręsti trečiąją problemą hiperparametrai derinti rankiniu būdu, kol algoritmas sėkmingai ištraukė galutinius narius iš BFHUD.



S.8 pav. HUNET modelio architektūros schema.

Siekiant išspręsti visas problemas buvo sukurtas naujas HU modelis, kuriame susijungia U-Net modelio architektūros idėjos ir transformatoriumi pagrįstas modelis, kurį sukūrė Ghosh ir kt. [28]. Segmentavimui naudotas originalus U-Net. Siekiant išgauti medžiagas ir gausos matricą bei atkurti vaizdą, prireikė modifikuoti architektūrą. Paveiksle S.8 pateikta naujoji HUNET modelio architektūros schema.

Pagrindiniai U-Net ir transformatorių modelių pakeitimai:

- Hiperspektrinio vaizdo padalijimas į mažesnius tokio paties dydžio vaizdus siekiant sumažinti bendrą modelio dydį, leidžiant įvesties duomenims naudoti papildymus (pvz., atspindėti ir pasukti), taip pat mokinti modelį, šias vaizdo pataisas pasirenkant atsitiktine tvarka.

- Suspaustų duomenų padalijimas į medžiagų ir gausos išgavimo potinklius.

- Modelio architektūra priklauso nuo mažiau hiperparametrų. Nau-

jasis modelis nenaudojo pleistro dydžio ir latentinės erdvės hiperparametrų, nes šie parametrai buvo transformatoriaus sluoksnių dalis.

- Kosinuso panašumo netikties pridėjimas buvo naudojamas siekiant paskatinti modelį išskirti ne tokias panašias medžiagas. Kosinuso panašumo netikties skaičiavimo funkcija yra priede E. Funkcija sukuria numatytų medžiagų vektorius į dvi matricas, kad funkcija *PyTorch cosine_similarity* galėtų apskaičiuoti visų galutinių narių porų panašumą. Tada apskaičiuotas rezultatas yra paverčiamas teigiamu ir apskaičiuojama vidutinė vertė. Gauta vertė padidinama iki diapazono nuo 0 iki 1, kad visos netiktys liktų panašiame diapazone ir sumažintų šios netikties įtaką modelio mokymui.

Galutinė HUNET modelio architektūra sudaryta iš šių modulių, pavaizduotų modelio architektūros diagramoje S.8 paveiksle:

- Encoding convolution blokas – grupės normalizavimo, 2D konvoliucijos ir ReLU sluoksnių rinkinys, kuris yra kodavimo tinklo dalies pagrindas;
- kodavimo blokas – sujungia du kodavimo konvoliucijos blokus ir po to prideda dropout sluoksnį;
- dekodavimo konvoliucijos blokas – sujungia du kodavimo konvoliucijos blokus su tiesinės interpoliacijos padidinimo algoritmu, kad vaizdas būtų padidintas du kartus;
- medžiagų dekodavimo sluoksnis – 2D konvoliucijos sluoksnis kartu su LeakyReLU aktyvinimo funkcija, kuri padidina spektrinį matmenį;
- 2D MaxPool sluoksniai – naudojami duomenims sumažinti du kartus dviem matmenimis ir naudojami kartu su dekoderio sluoksniais;
- linijinis dekodavimo sluoksnis su Sigmoid – naudojamas konvoliucijos kanalams rekonstruoti į spektrinius kanalus, skirtus medžiagoms išgauti;
- 2D konvoliucijos išlyginimo sluoksnis su Softmax – naudojamas gausos matricai iš dekoduotų duomenų išgauti.

Šiame skyriuje aprašomi atlikti HU eksperimentai, hiperspektrinių duomenų sintezės rezultatai, pasiūlytas etalonas HU algoritmo veikimui tikrinti. Algoritmai buvo išbandyti naudojant sukurtą etaloną, o siūlomo U-Net pagrindu sukurto išmaišymo modelio tikslumas palygintas su transformatorių naudojančio neuroninio tinklo modelio tikslumu. Skyrius pagrįstas rezultatais, publikuotais dokumentuose [A.1] ir [B.1]

S.6. Eksperimento planas

Apibendrinus skyriuje 1.2 pateiktus HU algoritmų tyrimus, pasiūlyta keletas įžvalgų ir iškeltos hipotezės:

- HU algoritmo našumas priklauso nuo galutinių narių skaičiaus, nes reikšmingesnis medžiagų skaičius nurodo mažesnį spektrinių duomenų kiekį kiekvienai medžiagai.
- Įvairūs HU algoritmai veikia blogiau, kai naudojami hiperspektriniai duomenys yra triukšmingi.
- Hiperspektrinių duomenų juostos, esančios arti viena kitos, pasižymi didele tarpusavio koreliacija ir gali būti apytiksliai sintezuojamos iš multispektrinių duomenų.

Eksperimento planas sukurtas remiantis šiomis hipotezėmis, taip pat siekiant patikrinti šiuos HU algoritmų aspektus. HU etalonas (paaiškintas skyriuje S.4) sukurtas siekiant patikrinti galutinį narį, kiekvieno galutinio nario duomenų kiekį ir algoritmo atsparumą triukšmui.

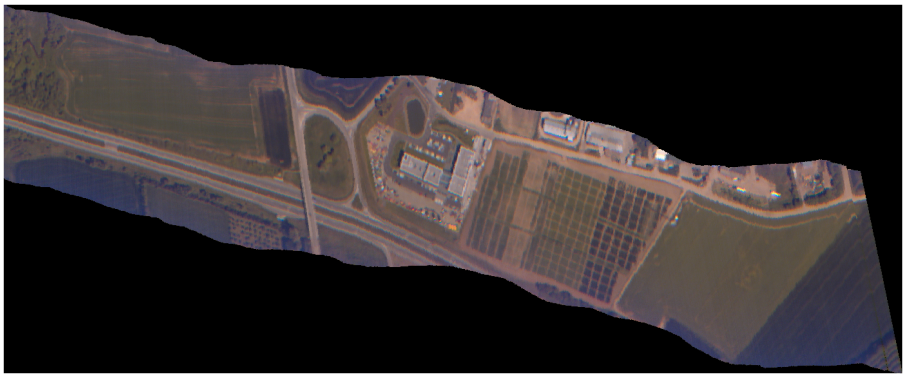
S.7. Hiperspektrinių duomenų sintezė

Pasitelkus Audbert ir kt. [4] GAN tinklą, buvo išplėstas antrasis duomenų sintezės metodas, aprašytas skyriuje 1.1.6. Jų pasiūlyta modelio architektūra ir įžvalgos buvo pritaikytos kuriant generatyvųjį modelį hiperspektriniams duomenims sintezuoti. Sukurtas modelis buvo naudojamas bandant sukurti 224 skirtingų spektrinių juostų hiperspektrinį vaizdą matomuose ir artimuose infraraudonųjų spindulių diapazonuose iš RGB vaizdo arba multispektrinių duomenų. Audbert ir kt. pasiūlytas GAN modelis buvo supaprastintas pritaikius šiuos pakeitimus:

- siekiant supaprastinti modelį pašalintas mokomojo diskriminatoriaus naudojimas, jis pakeistas rekonstrukcijos netiktimi, nes modelis generuoja hiperspektrinius duomenis ne nuo nulio, o iš esamos RGB arba multispektrinės informacijos;

- iš generatyvaus modelio pašalinus klasių informaciją, naujai sukurtas modelis visiškai jos nereikalauja. Pagrindinis šio modelio taikymo tikslas – papildyti esamus konkrečios srities hiperspektrinius duomenis naujai sugeneruotais duomenimis iš kitų šaltinių, pavyzdžiui, palydovinių vaizdų. Palyginti, Audbert ir kt. modelis buvo apmokytas generuoti specifinius klasės hiperspektrinius duomenis, atsižvelgiant į atsitiktinį triukšmą ir klasės etiketę.

Algoritmui kurti ir testuoti buvo pasirinkti trys hiperspektriniai vaizdai, surinkti naudojant BO. Pasitelkus CIE 1931 spalvų erdvės integravimo kreives, iš hiperspektrinių duomenų buvo sukurti RGB vaizdai. Jie sukurti siekiant imituoti geresnius RGB kamerų surinktus duomenis, o ne naudoti atskiras siauras juostas kiekvienai spalvai. RGB vaizdo pavyzdys pateiktas S.9 paveiksle viename iš testavimo duomenų kubų, naudojamų kuriant generatyvųjį modelį. RGB vaizdo kūrimas naudojant hiperspektrinių duomenų funkciją pateiktas priede C.



S.9 pav. RGB vaizdo, sugeneruoto iš vieno iš hiperspektrinių duomenų kubų, naudojamo kuriant generatyvųjį modelį, pavyzdys.

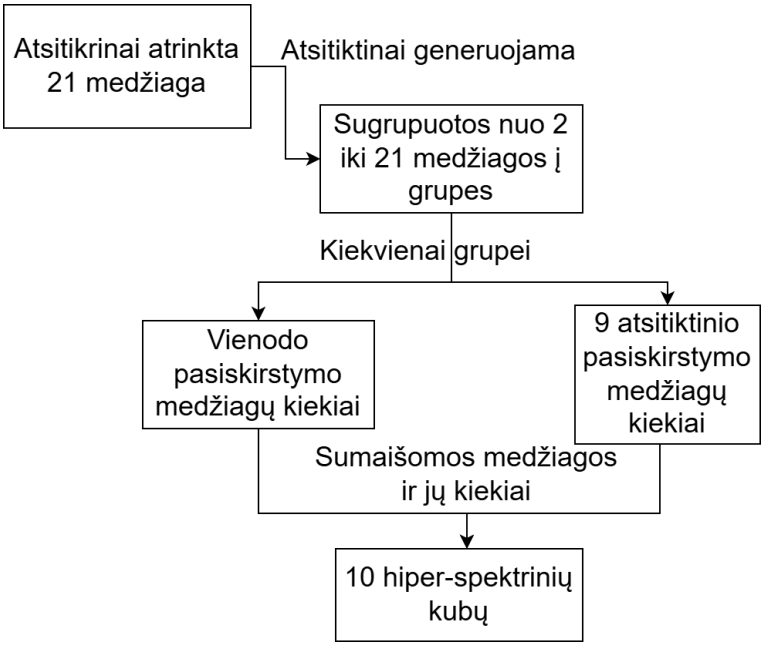
S.8. Lyginamosios analizės rezultatai

Šiame poskyryje pateikiami eksperimentų rezultatai, gauti taikant sukurtus algoritmus pagal numatytą metodiką. Kodas, naudotas kuriant ir vykdant šiuos etalonus, prieinamas adresu `https://github.com/VytautasPau/HUBenchmark.git`.

Atsparumas medžiagos kiekiui. Šis eksperimentas buvo atliktas naudojant dirbtinį duomenų rinkinį, sugeneruotą naudojant S.6 paveiksle pavaizduotą modelį. Šablonas turi klasių atitikmenį, to paties vaizdo apkarpymo versiją, tačiau vaizde pažymėta 20 skirtingų klasių (daugiau nei du medžiagų rinkiniai, priskirti kiekvienam pikseliui). Taikant šį klasifikavimo modelį, skirtingomis proporcijomis buvo sumaišyta 21 atsitiktinai atrinkta klasė. Pasirinkto gausumo nustatymas buvo atliktas pagal šiuos veiksmus, kurie taip pat pavaizduoti 3.5 paveikslo diagramoje:

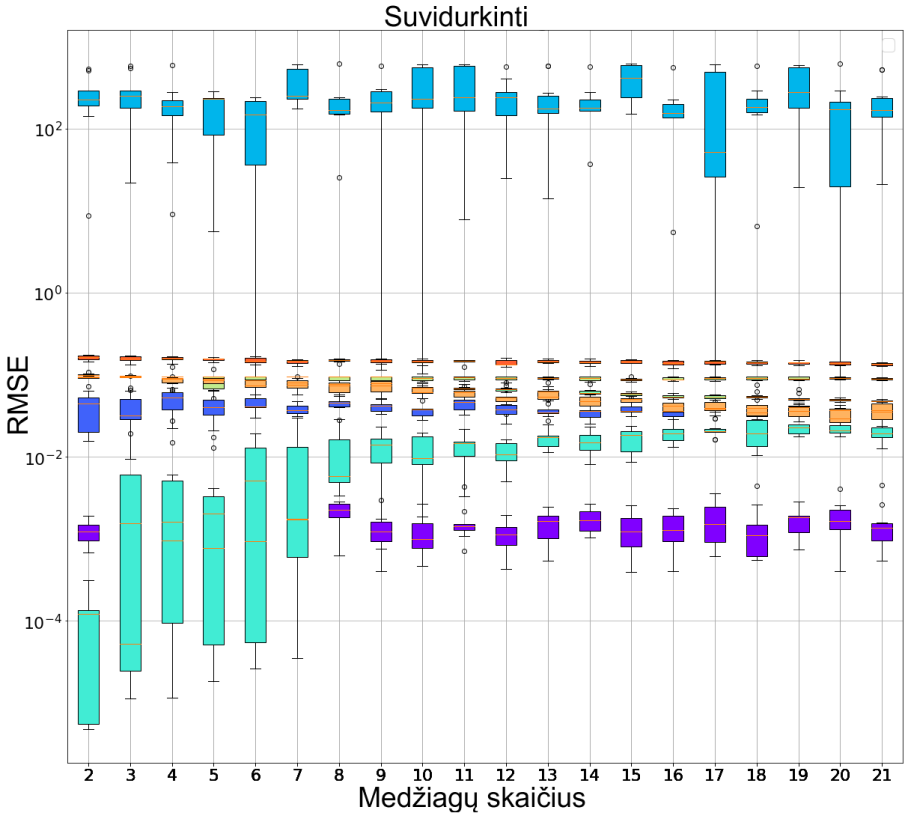
- dešimt skirtingų duomenų rinkinių buvo sukurti naudojant tas pačias medžiagas (21) siekiant pridėti statistinių skaičiavimų skirtumus;
- medžiagos atsitiktinai atrinktos į grupes, kad kiekvienai modelio klasei būtų sukurtas skirtingas galutinių narių skaičius nuo 2 iki 21;
- kiekvienai medžiagų grupei buvo sukurtos tolygiai paskirstytos gausos;
- kiti devyni gausumo variantai buvo atsitiktinai atrinkti ir sumaišyti į dešimt skirtingų hiperspektrinių vaizdų.

Paveiksle S.11 pateikti rezultatai, gauti taikant algoritmus ir skaičiuojant RMSE metriką tarp numatytų verčių ir sugeneruotų tikrųjų gausos reikšmių. 3.6 paveiksle beveik visi algoritmai, išskyrus RSNMF, demonstruoja nuoseklų RMSE su skirtingu klasių kiekiu vaizde, tačiau SGSNMF pasižymi didžiausiomis paklaidomis ir prasčiausiu našumu, o SUNSAL algoritmas pasiekė mažiausią paklaidą iš visų algoritmų. SGSNMF ir RSNMF algoritmai pasižymi reikšmingiausiais reikšmių pasiskirstymais. Mažesni skirstiniai rodo nuoseklesnius šių algoritmų rezultatus, o RSNMF yra nenuoseklus esant mažam galutinių narių kiekiui. Paryškintos reikšmės rodo geriausius kiekvieno algoritmo rezultatus, palyginti su galutinių narių skaičiumi.



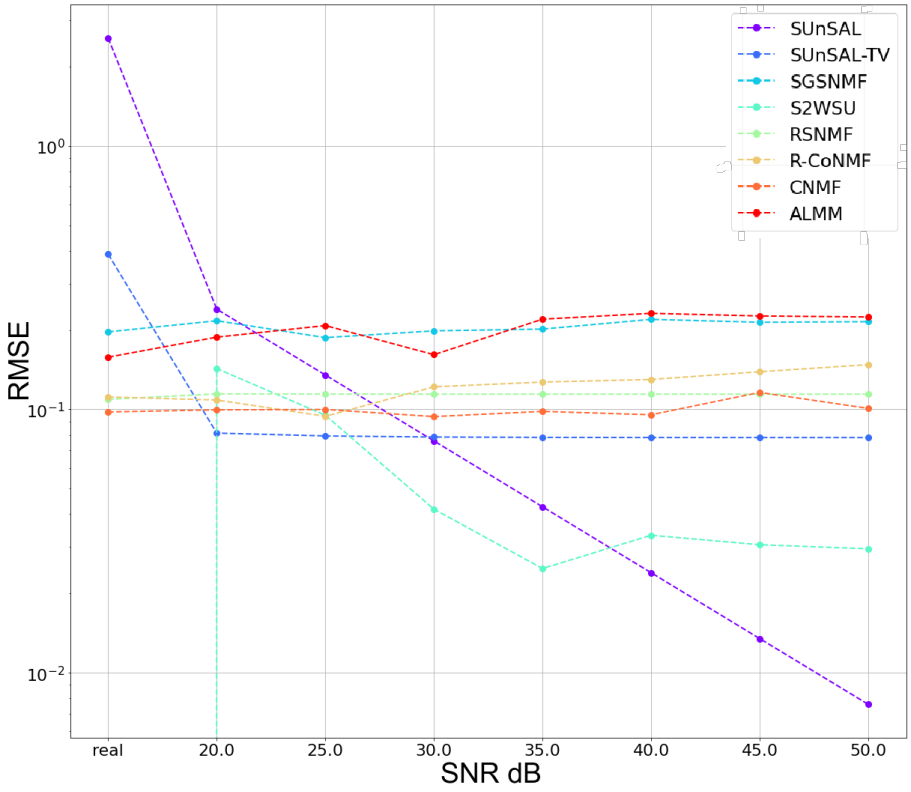
S.10 pav. Atsparumo medžiagos kiekiui eksperimento diagrama.

Atsparumas triukšmui. Hiperspektriniai vaizdai, kaip ir ankstesniame eksperimente, buvo sukurti naudojant S.6 pav. klasifikuotą paviršiaus vaizdą, kad būtų geriau atspindėtas medžiagų pasiskirstymas hiperspektriniame vaizde. Kaip aprašyta metodologijos skyriuje 3.3.1, buvo sukurtas vaizdas su penkiomis klasėmis ir pridėtas dirbtinis triukšmas. Paveiksle S.12 pateikta dešimt RMSE paleidimų, kiekvienam išbandytam algoritmui ir skirtingo pridėtinio dirbtinio triukšmo kiekiui, įtraukiant triukšmą, sukurtą iš tikros kameros triukšmo charakteristikų. Šis paveikslas atskleidžia, kad SUNSAL algoritmas pasižymi labai tiesine koreliacija tarp RMSE ir vaizdo triukšmo kiekio. SUNSAL yra blogiausias iš visų algoritmų, naudojančių tikrojo triukšmo charakteristiką. Kitų algoritmų rezultatai visais triukšmo lygiais beveik vienodi. RSNMF algoritmas pasižymi tiksliausiu bendru RMSE, kuris didesnis esant didesniam triukšmo lygiui. Tačiau jam būdingas tiksliausias medžiagų nustatymo rezultatas atliekant tikrus triukšmo eksperimentus.



S.11 pav. Atsparumo medžiagos kiekiui eksperimento rezultatas su kiekviena klasių grupe ir algoritmu. (Spalvos: violetinė – SUNSAL, tamsiai mėlyna – SUNSAL-TV, mėlyna – SGSNMF, šviesiai mėlyna – S2WSU, žydra – RSNMF, geltona – R-CoNMF, oranžinė – CNMF, raudona – ALMM). Kaip bandymo duomenys buvo naudojamas kombinuotas sintetinis IEEE GRSS ir USGS spektrinės bibliotekos duomenų rinkinys.

Vaizdo dydžio skirtumas. S.13 paveiksle pateikti devynis kartus sumažintų vaizdų RMSE rezultatai. Palyginti su keturis kartus sumažintais vaizdais, S.14 paveiksle pavaizduoti algoritmo veikimo rezultatai. Abu paveikslai atskleidžia panašius rezultatus, koreliuojančius su galutinio nario patikimumo eksperimento RMSE reikšmėmis, kai vaizdai nesumažinti.



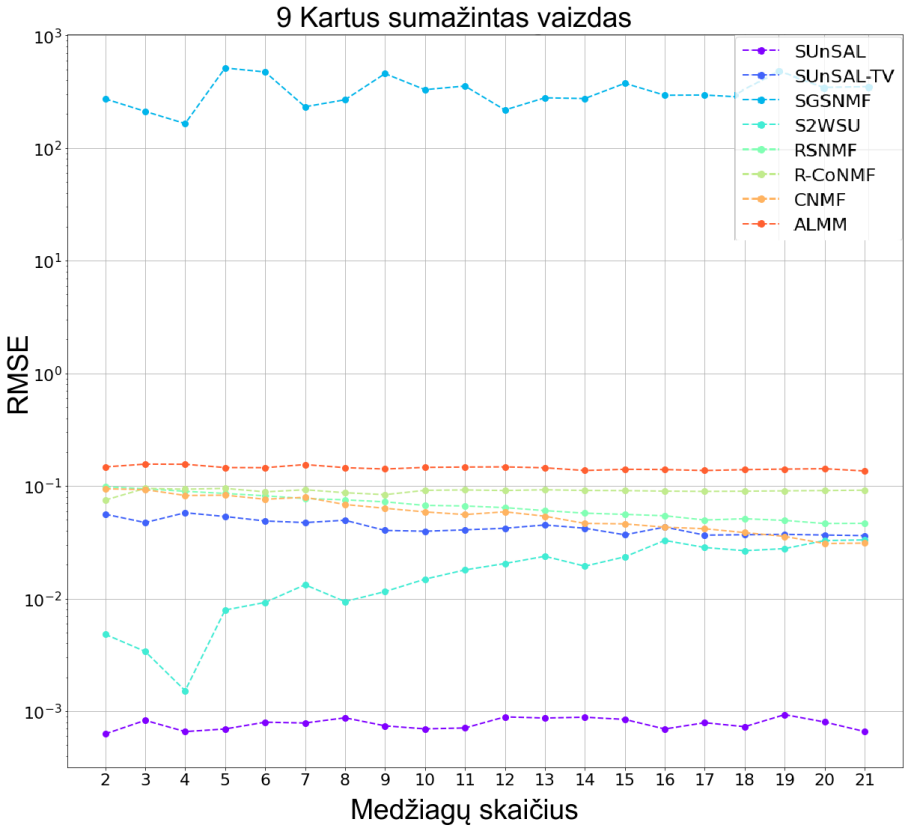
S.12 pav. Algoritmo atsparumo triukšmui eksperimento rezultatai. Kaip bandymo duomenys buvo naudojamas kombinuotas sintetinis IEEE GRSS ir USGS spektrinės bibliotekos duomenų rinkinys su pridėtu triukšmu. Rodomos reikšmės yra RMSE visų pikselių ir klasių vidurkiai.

S.9. HUNET modelio eksperimentai

Šiame skyriuje aprašomi eksperimentai, atlikti siekiant patikrinti naujai sukurto HUNET modelio veikimą, palyginti su moderniausiu transformatoriumi pagrįstu Hyperspectral Unmixing modeliu, taip pat modelio rezultatai, gauti taikant siūlomą HU lyginamosios analizės metodiką.

S.9.1. Naudoti duomenų rinkiniai

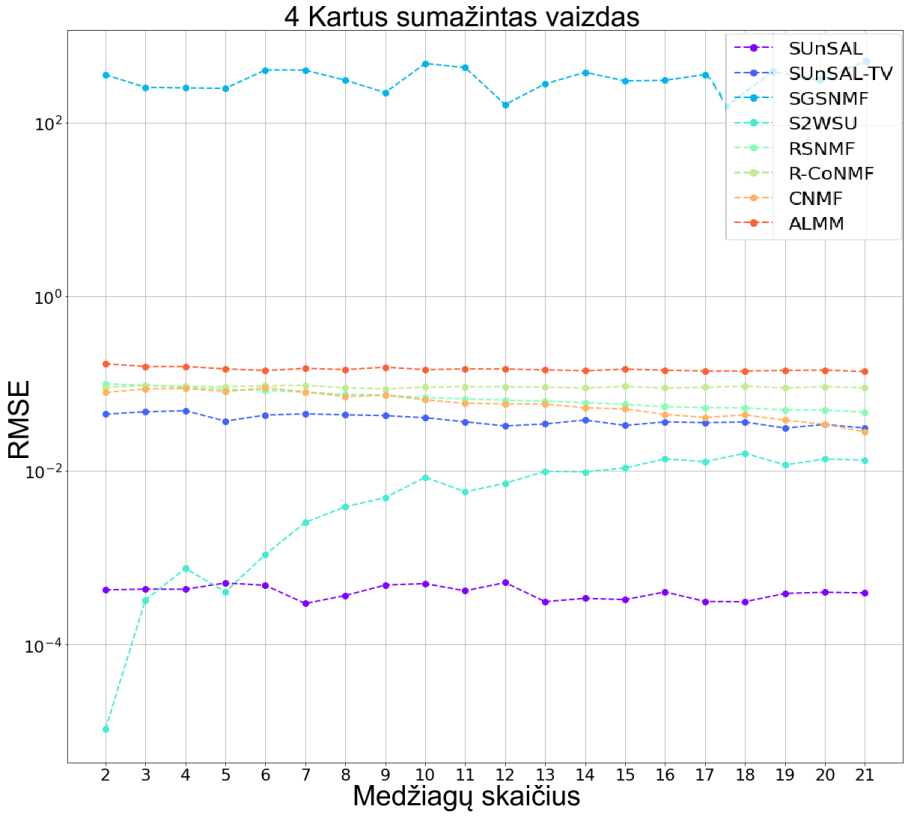
Šiame skyriuje aprašomi esamuose ir HUNET modeliuose naudojami duomenų rinkiniai ir jų vertinimai bei palyginimai. Analizuojami laisvai prieinami hiperspektrinių duomenų rinkiniai, kurie buvo atrinkti ir panaudoti algoritmo veikimo eksperimentuose. Taip pat įtraukiamas



S.13 pav. Algoritmo veikimas su devynis kartus sumažintais hiperspektriniais vaizdais. Buvo naudojamas kombinuotas sintetinis IEEE GRSS ir USGS spektrinės bibliotekos duomenų rinkinys, sumažintas devynis kartus. Rodomos reikšmės yra RMSE visų pikselių ir klasių vidurkiai.

naujai sukurtas HU duomenų rinkinys, aprašytas 2.2.1 skyriuje:

- *DC Mall* [75] [68]. Teritorija Vašingtone, DC, kurios dydis yra 1208 x 307 pikseliai ir 191 spektrinė juosta. Sukurta pagrindinė klasifikavimo tiesa turi šias klases: stogai, gatvės, takai, žolė, medžiai, vanduo ir šešėliai.;
- *Samson* [75] Hiperspektrinių duomenų kubas, iškirptas iki 95 x 95 pikselių dydžio su 156 spektro juostomis ir trimis skirtingomis klasėmis.
- *Apex* [75] Hiperspektriniai 110 x 110 pikselių duomenys su 285



S.14 pav. Algoritmo veikimas su keturis kartus sumažintais hiperspektriniais vaizdais. Buvo naudojamas kombinuotas sintetinis IEEE GRSS ir USGS spektrinės bibliotekos duomenų rinkinys, sumažintas keturis kartus. Rodomos reikšmės yra RMSE visų pikselių ir klasių vidurkiai.

spektro juostomis ir keturiomis skirtingomis klasėmis.

- BFHUD 1 kubo forma: 1024 pikselių pločio, 3177 pikselių ilgio su 224 spektrinio gylio juostomis.
- BFHUD 2 kubo forma: 1024 pikselių pločio, 3047 pikselių ilgio su 224 spektrinio gylio juostomis.
- BFHUD 3 kubo forma: 1024 pikselių pločio, 2815 pikselių ilgio su 224 spektrinio gylio juostomis.

S.9.2. Eksperimento rezultatai

Šiame skyriuje aprašomas eksperimentas, skirtas HUNET modeliui ir esamam transformatoriaus modeliui išbandyti naudojant BFHUD. Eksperimento bazė sukurta pasitekus Ghosh ir kt. [28] metodiką. Toliau pateikiami eksperimento žingsniai:

- kelių esamų duomenų rinkinių, naudojamų bandant transformatoriaus neuroninį tinklą, sukurtą Ghosh ir kt. , pasirinkimas. [28]. Tai apėmė *Samson* ir *Washington DC* duomenų rinkinius;
- BFHUD pritaikymas esamam modeliui ir kelių bandymų paleidimas siekiant surinkti hiperparametrų rinkinį;
- HUNET modelio pritaikymas esamam duomenų srautui, kurį sukūrė Ghosh ir kt., kad duomenų rinkinių ir modelių bandymai būtų nuoseklūs;
- tiek modelio mokymo atkūrimo klaidai (RE, žr. lygtį S.3), tiek spektrinio kampo atstumui (SAD, žr. lygtį S.4) pasitelktos šios netiktys. Prie HUNET modelio buvo pridėta papildoma kosinuso panašumo netiktis (žr. lygtį S.5);
- rezultatams patikrinti naudotos RMSE (žr. lygtį S.1) ir SAD metrikos.

RMSE, SAD ir RE rezultatai paleidžiant HUNET modelį ir transformatoriaus modelį, kurį sukūrė Ghosh ir kt. [28], su trimis pasirinktais duomenų rinkiniais pateikiami lentelėje S.1, kur BFHUD – naujai sukurtas duomenų rinkinys. Išskirtos medžiagos, palygintos su tikraisiais medžiagų kiekiais, pateikiamos kaip atskiri grafikai kiekvienam iš šešių dviejų modelių ir trijų išbandytų duomenų rinkinių derinių.

Bendrosios išvados

1. - HU algoritmai, tokie kaip SUnSAL, SUnSAL-TV, R-CoNMF ir ALMM, parodė beveik jokio skirtumo RMSE reikšmėse, kai medžiagų skaičius padidėjo nuo 2 iki 21, todėl jie priskiriami algoritmams, kurie nepriklauso nuo medžiagų skaičiaus. Kitų algoritmų, įskaitant naujai siūlomą HUNET modelį, veikimas priklausė nuo medžiagų skaičiaus.

Duomenų rinkinys	Modelis	RE	vid. RMSE	vid. SAD
BFHUD	Transformeris	0.3129	0.5054	**0.4135**
	HUNET	**0.0754**	**0.3625**	0.6632
DC Mall	Transformeris	**0.0253**	0.3853	0.3002
	HUNET	0.0451	**0.3832**	**0.1928**
Samson	Transformeris	0.1651	0.6027	**0.2411**
	HUNET	**0.0401**	**0.5215**	0.7880

S.1 lentelė: RMSE, SAD ir RE metrikos rezultatai iš siūlomų ir transformatorių modelių skirtingiems duomenų rinkiniams, įtraukiant BFHUD. Paryškintos reikšmės rodo geresnius kiekvieno duomenų rinkinio ir metrikos rezultatus tarp dviejų modelių.

- Eksperimentai su vaizdo dydžio skirtumais parodė, kad RSNMF, R-CoNMF, CNMF, ALMM, SUnSAL ir SUnSAL-TV algoritmų veikimas nesikeitė, o tai rodo, kad sumažinto dydžio hiperspektriniai vaizdai turėjo pakankamai informacijos medžiagų duomenims išgauti. Gautos RMSE reikšmės skyrėsi mažiau nei 10% (arba 0.002) tarp skirtingo dydžio hiperspektrinių vaizdų eksperimentų.
- Vykdant visą palyginimo testą, algoritmų skaičiavimo laikai buvo mažiausi SUnSAL ir R-CoNMF algoritmams, o greičiausias buvo S2WSU, kuris buvo apie 32 kartus lėtesnis nei SUnSAL.

2. - Testuojant algoritmų atsparumą triukšmui, anksčiau sukurti HU algoritmai, tokie kaip SUnSAL ir SUnSAL-TV, pateikė tikslesnius rezultatus ir mažiausias RMSE reikšmes (apie 0.008 SUnSAL ir 0.08 SUnSAL-TV), kai hiperspektriniams vaizdams buvo pridėtas Gauso triukšmas. Tačiau blogiausi rezultatai buvo, kai pridėtas triukšmas buvo pagrįstas realios hiperspektrinės kameros triukšmu. Naujas HUNET modelis pasiekė beveik identiškas RMSE reikšmes nepriklausomai nuo dirbtinio triukšmo tipo.
   - Tuo tarpu HU algoritmai, tokie kaip CNMF, SGSNMF ir RSNMF, veikė beveik vienodai nepriklausomai nuo pridėto triukšmo tipo ir kiekio, su RMSE reikšmėmis apie 0.1 CNMF ir

RSNMF bei 0.2 SGSNMF algoritmams.

3. Generuojant hiperspektrinius duomenis naudojant multispektrinius duomenis, buvo gautos daug mažesnės RMSE reikšmės (apie 0.034 vidutiniškai visiems bangos ilgiams), palyginti su hiperspektriniais duomenimis, generuotais iš RGB vaizdų (vidutinės RMSE reikšmės 0.1023 visiems bangos ilgiams), dėl didelio artimojo infraraudonojo spektro bangos ilgių koreliacijos.

4. - Naujai siūloma HUNET modelio architektūra sumažina reikalingų hiperparametrų skaičių ir pasiekia mažesnes HU RE ir RMSE reikšmes, palyginti su transformatoriniu medžiagų ir jų kiekių nustatymo algoritmu, atitinkamai vidutiniškai 0.12 ir 0.0732 visose duomenų aibėse.

5. - Naudojant HUNET modelio architektūrą, buvo pasiekti geresni RE (0.28) ir RMSE (0.14) rezultatai, palyginti su transformatoriniu medžiagų ir jų kiekių nustatymo algoritmu, UAV surinktoje hiperspektrinėje duomenų aibėje.
   - HUNET modelis pasiekė mažesnę vidutinę RMSE reikšmę (vidutiniškai 12% mažesnę) visose trijose testuotose hiperspektrinėse duomenų aibėse (BFHUD, DC Mall ir Samson), palyginti su transformatoriniu medžiagų ir jų kiekių nustatymo algoritmu. Priešingai, transformatorinis modelis pasiekė mažesnes vidutines SAD reikšmes (vidutiniškai 45% mažesnes) dviejose iš trijų duomenų aibių.

CURRICULUM VITAE

Vytautas Paura graduated from Simono Stanevičiaus Middle School in 2014 and with a bachelor's degree from Vilnius University Faculty of Mathematics and Informatics, "Information technologies" studies, in 2018. He graduated from Vilnius University Faculty of Mathematics and Informatics, "Computer Modelling" master studies, in 2020. In 2019, he started working in the "Beta Via" company as a data scientist and software engineer.

Vytautas Paura
Hyperspectral Unmixing of Hyperspectral Data Gathered Using an UAV
Doctoral Dissertation
Natural Sciences
Informatics (N 009)
Thesis Editor: Jamie Redfern

Vytautas Paura
Medžiagų ir jų kiekių nustatymas hiperspektriniuose duomenyse surinktuose naudojant bepiločius orlaivius
Daktaro disertacija
Gamtos mokslai
Informatika (N 009)
Santraukos redaktorė: Gabija Bankauskaitė

Vilniaus universiteto leidykla
Saulėtekio al. 9, III rūmai, LT-10222 Vilnius
El. p. info@leidykla.vu.lt, www.leidykla.vu.lt
bookshop.vu.lt, journals.vu.lt
Tiražas 20 egz.