

Hyperspectral Image Unmixing Using Residual-Convolutional Autoencoder in an Unsupervised Scenario



Project ID: Fall-23-25

Session: BSCS Fall 2020 to 2024

Project Advisor: Dr. Nadeem Iqbal

Submitted By

Amad Zaid	70109216
Waleed Baig	70109115

**Department of Computer Science & IT
The University of Lahore
Lahore, Pakistan**

Declaration

We have read the project guidelines and we understand the meaning of academic dishonesty, in particular plagiarism and collusion. We hereby declare that the work we submitted for our final year project, entitled **Hyperspectral Image Unmixing Using Residual-Convolutional Autoencoder in an Unsupervised Scenario** is original work and has not been printed, published or submitted before as final year project, research work, publication or any other documentation.

Group Member 1 Name: Amad Zaid

SAP No: 70109216

Signature:

Group Member 2 Name: Waleed Baig

SAP No: 70109115

Signature:

Statement of Submission

This is to certify that **Amad Zaid** 70109216, and **Waleed Baig** 70109115 have successfully submitted the final project named as: **Hyperspectral Image Unmixing Using Residual-Convolutional Autoencoder in an Unsupervised Scenario**, at Computer Science & IT Department, The University of Lahore, Lahore Pakistan, to fulfill the partial requirement of the degree of **BS in Computer Science**.

Supervisor Name: Dr. Nadeem Iqbal

Signature:

Date: 08-07-2024

Dedication

This thesis is dedicated to our families, whose unwavering support and encouragement have been a constant source of inspiration throughout this journey. Your belief in our abilities and understanding during moments of intense focus are the pillars upon which this work stands. To our esteemed supervisor, Dr. Nadeem Iqbal, your guidance and unwavering support have been invaluable. Your expertise has not only shaped the trajectory of our research but has also been a source of motivation and learning. This accomplishment would not have been possible without the collective support of our friends and mentors.

Acknowledgment

We truly acknowledge the cooperation and help make by Name of **Dr. Nadeem Iqbal, Assistant Professor at the University of Lahore, 1-Km Defence Road, near Bhuptian Chowk, Lahore, Punjab.** He has been a constant source of guidance throughout the course of this project. We would also like to thank **Dr. Nadeem Iqbal from Assistant Professor at the University of Lahore, 1-Km Defence Road, near Bhuptian Chowk, Lahore, Punjab** for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

Date: 08-07-2024

Abstract

In this study, we explore the development of a deep learning-based unmixing algorithm for hyperspectral data analysis. Hyperspectral imaging (HSI) captures data across numerous spectral bands, providing detailed information that surpasses traditional multispectral imaging. Hyperspectral unmixing aims to decompose the mixed pixel spectrum into its constituent pure spectral signatures, known as endmembers, and their corresponding abundance fractions. In this study, we proposed a deep learning-based algorithm for hyperspectral image unmixing (RCAE), employing an autoencoder architecture to decompose mixed pixel spectra into pure spectral signatures (endmembers) and their abundance fractions. Using a linear mixing model (LMM), the encoder compresses hyperspectral data into a lower-dimensional latent space, while the decoder reconstructs it back to the original dimensions. RCAE utilizes residual connection to enhance feature extraction and mitigate training challenges and to preserve the spatial structure of the HSI, we avoided the use of pooling layers and up sampling layers. The endmembers are extracted as the weights of the decoder layer and feature maps of hidden layer provides us with abundance maps. The experiments shows that the proposed RCAE model performs well on the real hyperspectral dataset particularly well when it comes to endmembers' extraction.

Area of the Project

The areas of the project include Remote sensing, Deep learning, Image processing, Computer vision, Pattern recognition, Image classification, and Hyperspectral Images.

Technologies used

Technologies used in our research project:

- TensorFlow & Keras.
- Numpy.
- Matplotlib.
- Scikit Learn.
- SciPy.
- MATLAB R2020a – Hypercube library is used to read Hyperspectral data.

List of Figures

Figure 1.1 RGB vs Hyperspectral Image of Plum fruit	1
Figure 1.2 A three-dimensional hyperspectral data cube.....	1
Figure 1.3 Reflectance curve of soil, water, and grass	2
Figure 1.4 (a) Linear Mixture Model. (b) Non-linear Mixture Model.....	3
Figure 1.5 A deep neural network for digit classification.....	6
Figure 1.6 Figure 1.6: Encoder-Decoder Architecture	7
Figure 1.7 Image classification with convolution neural	8
Figure 1.8 Convolutional autoencoder for hyperspectral unmixing	9
Figure 2.1 Steps involved in unmixing	12
Figure 2.2 Linear mixing.	13
Figure 2.3 (a) Illustration of linear mixing. (b) Illustration of nonlinear mixing	14
Figure 2.4 Categorization of geometrical based approach.....	15
Figure 2.5 Illustration of VCA.....	16
Figure 2.6 Concept of minimum simplex containing all the pixels	18
Figure 2.7 n-1 spectral vectors on each facet of data simplex.	19
Figure 2.8 Noisy data representation	19
Figure 2.9 Pure pixel, n-1 spectral vectors and no pure pixel scenerio	21
Figure 2.10 Illustration of CNN architecture	23
Figure 4.1 Proposed RCAE Network.....	31
Figure 4.2 Residual Connection.....	32
Figure 5.1 Samson dataset	33
Figure 5.2 Jasper-Ridge dataset	33
Figure 5.3 Urban dataset	34
Figure 5.4 Spectral signatures of 12-endmembers in cuprite dataset	35
Figure 6.1 Urban (four) dataset endmember signatures (Extracted and Ground Truth).....	37
Figure 6.2 Urban (four) dataset abundance maps (Extracted and Ground Truth)	38
Figure 6.3 Samson dataset endmember signatures (Extracted and Ground Truth)	38
Figure 6.4 Samson dataset abundance maps (Extracted and Ground Truth)	38
Figure 6.5 Jasper-Ridge dataset endmember signatures (Extracted and Ground Truth)	39
Figure 6.6 Jasper-Ridge abundance maps (Extracted and Ground Truth)	39
Figure 6.7 Urban (five) dataset endmember signatures (Extracted and Ground Truth)	40
Figure 6.8 Urban (five) dataset abundance maps (Extracted and Ground Truth).....	40

List of Tables

Table 4.1 Layers in the proposed encoder module	29
Table 5.1 (a) Characteristics of the Samson dataset	33
Table 5.1 (b) Shows a list of endmembers in Samson dataset	33
Table 5.2 (a) Characteristics of the Jasper-Ridge dataset	34
Table 5.2 (b) Shows a list of endmembers in Jasper-Ridge dataset.....	34
Table 5.3 (a) Characteristics of the Urban dataset	35
Table 5.3 (b) Shows a list of endmembers in Urban dataset.....	35
Table 5.4 (a) Characteristics of the Cuprite dataset	36
Table 5.4 (b) Shows a list of endmembers in Cuprite dataset.....	36
Table 7.1 SAD and average SAD scores for the Urban (four) dataset	41
Table 7.2 SAD and average SAD scores for the Samson dataset	41
Table 7.3 SAD and average SAD scores for the Jasper-Ridge dataset.....	41
Table 7.4 SAD and average SAD scores for the Urban (five) dataset.....	43
Table 7.5 RMSE scores for the Urban (four) dataset.....	43
Table 7.6 RMSE scores for the Samson dataset	43
Table 7.7 RMSE scores for the Jasper-Ridge dataset	43
Table 7.8 RMSE scores for the Urban (five) dataset	44
Table 7.9 Benchmark methods used for comparison in experiments	44
Table 7.10 Average SAD and RMSE obtained by different methods for the Urban.....	45
Table 7.11 Average SAD and RMSE obtained by different methods for the Samosn	45
Table 7.12 Average SAD and RMSE obtained by different methods for the Japser-Ridge.....	46

Table of Content

Declaration	ii
Statement of Submission	iii
Dedication	iv
Acknowledgment	v
Abstract	vi
List of Figures	ix
List of Tables	x
Chapter 1: Introduction to the Hyperspectral Image	1
1.1 Introduction.....	1
1.2 Background.....	2
1.3 Objectives	4
1.4 Motivation	4
1.5 Introduction to deep learning	5
1.5.1 What is Deep Learning?.....	5
1.5.2 Applications of Deep Learning.....	6
1.5.3 Deep Learning in Hyperspectral Image Unmixing	6
1.5.4 Deep Autoencoder for Hyperspectral Unmixing.....	7
1.5.5 Convolution Neural Networks	8
1.5.5.1 CNN in Computer Vision	8
1.5.5.2 Convolutional Autoencoders and Unmixing Problem	9
1.5.6 Benefits of Deep Learning Approach in HS Unmixing.....	9
1.6 Conclusion	10
Chapter 2: Literature Review	11
2.1 Introduction.....	11
2.2 Stages in Hyperspectral Unmixing	11
2.3 Linear and Nonlinear Mixing Models.....	12
2.3.1 Linear Mixing Model.....	13
2.3.2 Nonlinear Mixing Model.....	14

2.4	Categorization of Hyperspectral Unmixing Models.....	15
2.4.1	Geometrical-Based Approach for HSU	15
2.4.1.1	Pure Pixel (PP) Based Algorithms.....	15
2.4.1.1.1	Vertex Component Analysis (VCA) for HSU	16
2.4.1.1.2	N-FINDR for HSU	17
2.4.1.2	Minimum Volume (MV) Based Algorithms	18
2.4.2	Statistical Based Approach for HSU	19
2.4.3	Sparse Regression Based Approach for HSU	20
2.4.4	Summary of Traditional Algorithms	21
2.4.5	Deep Learning Based Algorithms for HSU	22
2.5	Performance Metrics for Quantitative Measurement of Algorithms	24
2.5.1	Spectral Angle Distance (SAD)	24
2.5.2	Root Mean Squared Error (RMSE)	24
2.5.3	Mean Squared Error (MSE).....	25
2.5.4	Spectral Information Divergence (SID)	25
2.6	Conclusion	25
Chapter 3:	Problem Statement for HS Unmixing	26
3.1	Introduction.....	26
3.2	Problem Definition and Notations.....	26
3.3	Physical Constraints on Abundance Matrix	26
3.3.1	Abundance Sum-to-One (ASC)	26
3.3.2	Abundance Nonnegatvie (ANC).....	27
3.4	Conclusion	27
Chapter 4:	The Proposed Method for Hyperspectral Image Unmixing	28
4.1	Introduction.....	28
4.2	Residual-Convolutional Autoencoder (RCAE)	28
4.2.1	Encoder Block.....	29
4.2.2	Decoder Block	31
4.2.3	Residual Connection	31
Chapter 5:	Hyperspectral Unmixing Datasets	32
5.1	Introduction	32
5.2	Hyperspectral Unmixing Datasets	32

5.2.1	Samson Dataset.....	32
5.2.2	Jasper-Ridge Dataset.....	33
5.2.3	Urban Dataset	34
5.2.4	Cuprite Dataset	35
5.3	Conclusion	36
Chapter 6:	Experimental Results	37
6.1	Introduction.....	37
6.2	Experiment Results.....	37
6.2.1	Experiment with Urban (four) dataset	37
6.2.2	Experiment with Samson dataset	38
6.2.3	Experiment with Jasper-Ridge dataset.....	39
6.2.4	Experiment with Urban (five) dataset	39
6.3	Conclusion	40
Chapter 7:	Validation and Comparative Analysis.....	41
7.1	Introduction	41
7.2	Loss Functions	41
7.3	Validation Results	42
7.3.1	Validating endmembers with SAD loss function	42
7.3.2	Validating abundance maps with RMSE loss function	43
7.4	Comparative Analysis	44
7.4.1	Comparison with Urban Dataset	44
7.4.2	Comparison with Samson Dataset	45
7.4.3	Comparison with Jasper-Ridge Dataset	45
7.5	Conclusion	46
Chapter 8:	Summary and Conclusion	47
Chapter 9:	Lesson Learnt and Future Work	49
References	50	
Appendix	54	
Appendix A.	Convolutional Neural Networks in Computer Vision.....	54

Chapter 1: Introduction to Hyperspectral Image

1.1 Introduction

In hyperspectral (HS), the word “hyper” means “an excess” or “overabundance” and it refers to the huge number of spectral wavelengths captured by HS cameras. When light interacts with an object, it gets absorbed, scattered, or reflected which is captured by hyperspectral imaging camera. While the human eye is only able to see 3 colors (Red, Green, and Blue), the HS imaging measures the continuous spectrum of light for each pixel present in the scene, not only in visible region but across the electromagnetic spectrum (see Figure 1.1). The image captured by a HS camera is referred to as “three-dimensional (x,y,z) data cube” where x and y represent the two spatial dimensions of the scene, and z represent the spectral dimension [1] as shown in Figure 1.2.

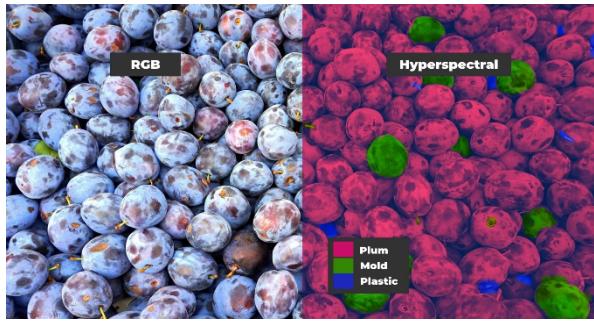


Figure 1.1: RGB vs Hyperspectral Image of Plum fruit

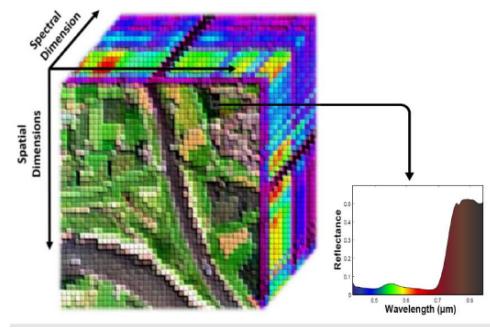


Figure 1.2: A three-dimensional hyperspectral data cube

Hyperspectral images find various applications in resource management, agriculture, mineral exploration, and environmental monitoring. The detailed analysis of an image allows us to distinguish between the materials present in the image based on their spectral signatures. These spectral signatures are unique for every material. It is estimated that three percent of the solar radiation that enters the planet is reflected into space by the land. The atmosphere either reflects the remainder or absorbs it and emits infrared energy back into space. Different amounts of energy are absorbed and reflected at various wavelengths by the various surface-level materials. A substance's spectral response pattern describes how much energy it absorbs, reflects, or emits over a range of spectrum of wavelengths [2]. Figure 1.3 illustrates the reflectance values of soil, water, and grass. For example, the reflectance values of water, soil and grass are between 0.3 and 0.6 micrometer. As depicted in Figure 1.3, at longer wavelengths, soil reflects more energy than grass. The majority of incoming radiation is absorbed by water over the entire wavelength range. It is possible to recognize materials and geological formations in remotely sensed pictures and evaluate their state by being aware of their typical spectral response characteristics.

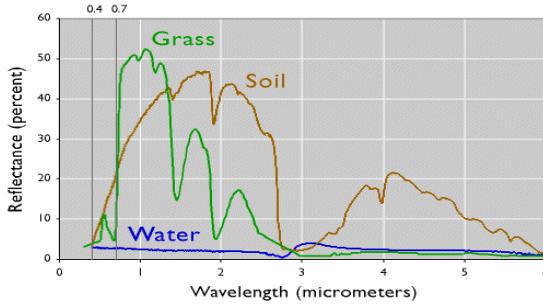


Figure 1.3: Reflectance curve of soil, water, and grass

Hyperspectral unmixing (HSU) is the process of identifying different materials in an image and their corresponding fractional abundances by decomposing each pixel into its pure spectral signatures called *endmembers*. The significance of HSU lies in its ability to extract valuable information from hyperspectral data, leading to better decision-making, resource optimization, and a deeper understanding of our environment. Some of the benefits of HSU are given below,

- A. Material Identification:** The HS sensors being able to capture huge number of bands allows us to get detailed analysis of the targeted material leading to enhanced material identification. This is extremely valuable in applications such as mineral detection, agriculture, and environmental monitoring.
- B. Quantitative Analysis:** HSU provides quantitative information about the abundance fractions of materials in each pixel of the image. This enables precise analysis and characterization of the spatial distribution and proportion of materials within a scene.
- C. Versatile:** The HS sensors are very versatile and are not bound to a single domain. The same sensor can be used for environmental monitoring, food inspection, and agriculture.
- D. Improves quality:** HSI is a great tool for quality assurance since the entire process can be automated to inspect object surfaces to identify anomalies or variations. It works on all types of surfaces including painted materials, wood, plastic, metal etc.

1.2 Background

Hyperspectral imaging is a powerful method that acquires data over a broad spectrum of wavelengths, allowing the collection of rich spectral data for each pixel in an image. Hyperspectral imaging offers a detailed spectral signature for each pixel, as compared with traditional imaging methods that only offer a few bands or channels, enabling a deeper understanding of the scene being studied. As a result, hyperspectral imaging has a variety of uses in remote sensing, agriculture, environmental monitoring, and mineral exploration.

In 1999, Maria Petrou suggested the 2 basic models for Unmixing in her book “Image Processing: The Fundamentals” [3]. Both the models had made a significant mark in the spectral mixture problem. In the linear mixing model (LMM), it is assumed that the light only interacts with one

material and the observed spectra can be represented as a linear combination of the pure spectral signatures of the endmembers. On the other hand, non-linear mixing models involve non-linear interaction. It acknowledges that the mixing process can involve nonlinear interactions such as multiple scattering, spectral variations, and atmospheric interactions etc. Figure 1.4 illustrates the interactions of both models.

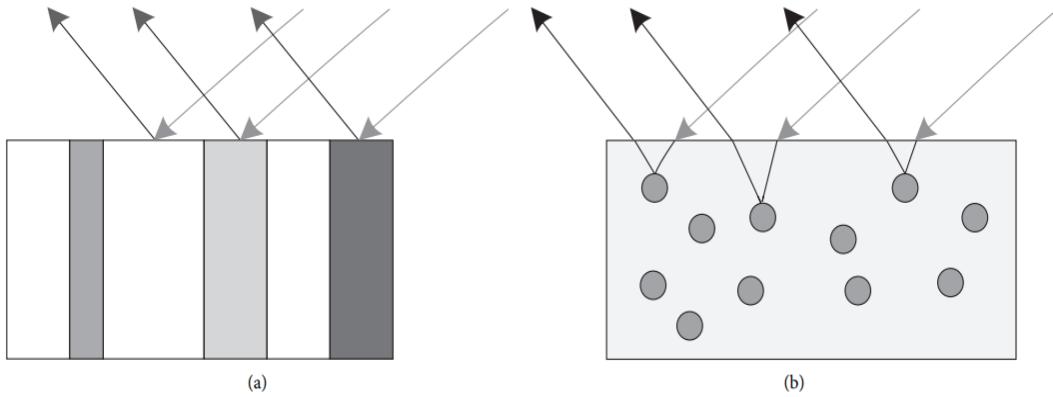


Figure 1.4: (a) Linear Mixture Model. (b) Non-linear Mixture Model

Hyperspectral unmixing methods used in the past have several phases. The pure spectral endmembers in the image are first identified via endmember extraction. After that, abundance estimation determines the ratios of each endmember inside each pixel. The methods in past used to perform unmixing are referred to as traditional methods. These methods are categorized into statistical, geometrical, and sparse regression-based methods depending on the approach that they used to solve the problem [1]. Sparse regression methods are one of the earliest methods used to address the problem of selecting a small set of endmembers from a large set of possible known candidates (spectral library). These methods aim to identify the most relevant endmembers that contribute significantly to the observed mixed pixel spectra while suppressing the influence of irrelevant or noise-related endmembers [4]–[7]. The second category of traditional methods are geometrical based that are used to estimate endmembers and abundance fractions by making use of geometrical structures and properties that are present in the hyperspectral data. These methods take advantage of the geometric correlations between the mixed pixel spectra and the endmember spectral signatures. Geometrical methods can be further categorized into pure pixel and no pure pixel depending on the fact that they rely on presence of pure pixel or not in the observed image. A well-known example of a method that relies on presence of pure pixel is vertex component analysis (VCA) [8], while Minimum volume simplex analysis (MVSA) [9] and a variable splitting augmented Lagrangian (SISAL) [10] is perfect example of no pure pixel scenario. The last category of this method redevelops the unmixing problem as an inference problem [11] – [13] and makes use of probability and statistical modeling to find the most likely parameters that explain the observed data.

The earlier methods often relied on linear mixing models, which might not accurately represent complex nonlinear mixing situations. Second, these techniques rely on pure spectral signatures, which are infrequent in real-world situations and can produce unmixing results that are unreliable. Furthermore, the "curse of dimensionality" could affect traditional techniques when dealing with high-dimensional data. Additionally, they are unable to effectively incorporate spatial information, which restricts their capacity to manage spatially complicated mixing scenarios. Finally, traditional methods use manual feature engineering techniques which are time consuming, require intensive labor and domain knowledge. Deep learning techniques, on the other hand, make use of the capabilities of neural networks to automatically learn the complex relationships between spectral signatures as well as abundance maps. Deep learning algorithms may capture complex spectrum variations and geographic associations by training on large-scale hyperspectral datasets, which leads to enhanced unmixing accuracy. Additionally, end-to-end learning made possible by deep learning eliminates the need for human feature extraction and allows the network to automatically pick up on and adjust to various hyperspectral scenarios. Section 1.5 discusses in detail the deep learning approach for hyperspectral unmixing.

1.3 Objectives

The objectives that are to be achieved by the end of this research project:

- 1- **Deep Learning based algorithm:** This model utilizes the convolution autoencoder architecture to unmix the hyperspectral data into pure spectra and their corresponding abundance on the surface. This algorithm will not require any prior knowledge of materials in the scene. It will be able to learn complex features from the HS data using the encoder-decoder mechanism.
- 2- **Accurate Material Estimation:** This algorithm will be able to extract materials achieving results that are comparable to state-of-the-art unmixing algorithms.
- 3- **Unsupervised Method:** This model will be fully unsupervised such that it does not rely on any labeled data for training. Moreover, in hyperspectral unmixing, unsupervised refers to identifying endmembers and their corresponding fraction. Whereas in the supervised method, the endmembers are assumed to be known and only abundance maps are extracted for each material.
- 4- **Incorporate Spectral-Spatial Information:** The model will not only rely on rich spectral information of the image but will also take into account the spatial information of the image. The way in which the materials are laid on the land and their local dependencies are crucial for the accurate estimation of endmembers.

1.4 Motivation

The hyperspectral unmixing problem is indeed interesting and important due to several reasons. Some of them are listed below:

Information-rich data: Hyperspectral images capture many narrow and contiguous spectral bands, providing detailed information about the electromagnetic spectrum. This rich spectral

information enables the identification and characterization of materials and their mixtures at a fine-grained level. HU techniques aim to exploit this wealth of data to extract valuable information and uncover hidden patterns in the observed scene.

Remote Sensing Applications: Hyperspectral imaging technology is widely used in remote sensing applications, such as satellite imagery and aerial photography. It allows us to capture detailed information about Earth's surface and monitor various phenomena. Hyperspectral unmixing helps in extracting valuable information, enabling accurate land cover classification, target detection, and environmental monitoring.

Material Identification and Quantification: Hyperspectral unmixing plays a crucial role in identifying and quantifying the materials present in a scene. By decomposing mixed pixel spectra into their constituent materials, it becomes possible to determine the abundance or proportion of each material. This information is valuable in numerous fields, such as geology, agriculture, mineralogy, and environmental science. For example, in mineral exploration, hyperspectral unmixing helps identify mineral compositions, which can aid in locating valuable mineral deposits.

Anomaly Detection: Hyperspectral unmixing techniques are also useful for anomaly detection. Anomalies can be caused by various factors like pollution, contamination, or the presence of hidden objects. By unmixing the hyperspectral data, it becomes easier to identify unusual or unexpected spectral signatures that may indicate the presence of anomalies. This is particularly important in applications like environmental monitoring, where the early detection of anomalies can help prevent or mitigate potential risks.

Scientific Research and Exploration: Hyperspectral unmixing is of great interest to the scientific community as it enables researchers to gain deeper insights into the composition and characteristics of various materials. By unmixing hyperspectral data, scientists can study the spatial distribution of different components within a scene, analyze their spectral properties, and understand their interactions. This knowledge contributes to advancements in fields like planetary science, environmental studies, and ecology.

1.5 Introduction to Deep Learning

1.5.1 What is Deep Learning?

Deep Learning (DL) is a subfield of machine learning that is inspired by the functioning of the human brain. The word “deep” does not refer to in depth knowledge acquired by this approach; instead, it reflects the mechanism of successive layers of representation followed by deep learning approach [14]. The number of layers in the network is the depth of the network. Deep learning is primarily based on artificial neural networks, which are used for the learning of these layered representations. The interconnected nodes, called neurons, that make up these networks are grouped in layers. Each neuron takes in information, processes it, and produces an output. The strength of these connections, known as weights, among neurons determines the significance of each neuron's contribution. Figure 1.5 shows a 4-layer deep neural network that processes an image to classify it. In deep neural network, each layer acts as filter which progressively refines the input as it passes through these

layers. The weights associated with each neuron determine the contribution of that specific neuron to the final output. The weights are just numbers (matrix) that are randomly initialized.

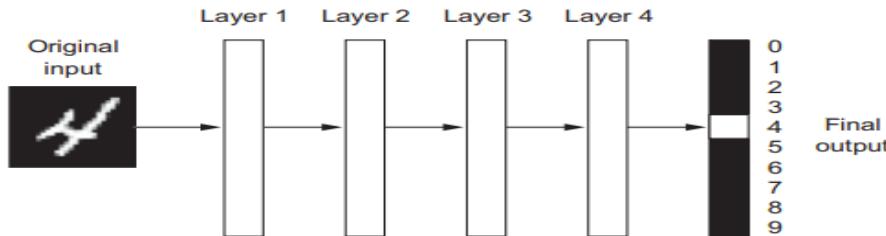


Figure 1.5: A deep neural network for digit classification

Here, learning refers to finding the right entries for the weight matrix of all layers for which the input given to the model can be correctly mapped to the desired output. The weights are sometimes also referred to as “parameters” of the network. To determine how far the output of the model is from the expected result, we use the *loss function*, also known as the *objective function*. This function receives two parameters, the output achieved and the expected result, and then computes a distance score indicating how well the model performed. The goal is to minimize this loss function. We will discuss in detail different loss functions, weight initialization techniques, and other concepts related to deep learning in later chapters.

1.5.2 Applications of Deep Learning

Although deep learning is an old field that dates to 1943, but it only gained importance in the early 2010s. In just a few years, it showed remarkable results in sensory and perceptual problems that once were considered impossible to be achieved by traditional machine learning models. Deep learning has found numerous applications across various fields due to its ability to learn and extract complex patterns from large datasets. Some notable applications of deep learning include:

- Image Classification.
- Speech recognition.
- Enhanced text to speech conversion.
- Autonomous Vehicles.
- Healthcare and Drugs Discovery.
- Be able to response to questions asked in natural language.
- Defeats the world’s best player of Superhuman Go playing.

These are just a few examples of the wide-ranging applications of deep learning. The full potential of what deep learning is capable of is still unknown and is being explored by researchers from all over the world. In short, the versatility and adaptability of deep learning

algorithms make them a powerful tool in solving complex problems across various domains.

1.5.3 Deep Learning in Hyperspectral Image Unmixing

As explained in Section 1.2, traditional methods for hyperspectral image unmixing have limitations such as assuming linearity, manual feature engineering, and it does not take into consideration the spatial context of an image. The traditional methods are sensitive to noise meaning that they struggle to handle noise sources, leading to a degraded performance in unmixing results. The recent advancements in computer technology and availability of data encouraged researchers to apply deep learning techniques to solve HS unmixing problem [15]. It was found that deep learning outperformed all the traditional methods in many aspects including the ability to include spatial information, and automatic feature engineering [16]. The three widely used deep learning architectures so far used to perform HSU are generative model, autoencoders and convolutional neural network. Since the most widely used model for HSU are deep autoencoder and more recently convolution neural networks, we will discuss the same.

1.5.4 Deep Autoencoder for Hyperspectral Unmixing

Deep autoencoders are a type of artificial neural network that can learn useful features from images. They consist of two main blocks: an encoder block and a decoder block. The encoder block takes the input image and compresses it to a dimension which is lower than the dimension of input image. This compressed representation is called the *latent space* or *bottleneck layer*. The decoder block then reconstructs the image from this latent space back to its original dimension [17]. Figure 1.6 illustrates the concept of encoder-decoder architecture.

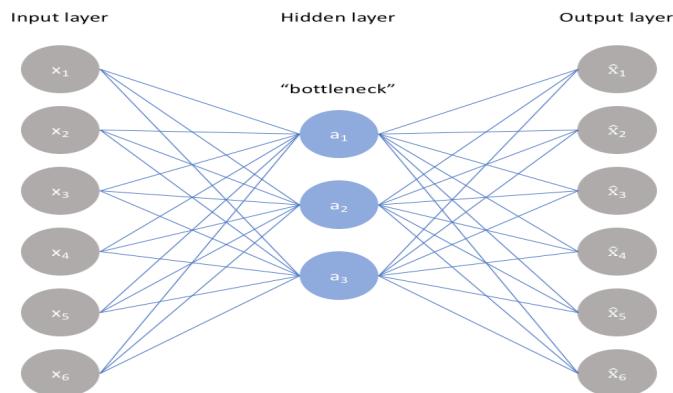


Figure 1.6: Encoder-Decoder Architecture

During training, the deep autoencoder aims to minimize the difference between the original input image and the reconstructed image. By doing so, the autoencoder learns to capture the most important and distinguishing features of the image (hyperspectral data). Once the deep

autoencoder is trained, the latent space representation can be used for HSU. The compressed representation learned by the autoencoder can be considered as a set of meaningful and compact features that capture the underlying characteristics of the hyperspectral image. To perform HSU, the latent space representation is fed into subsequent unmixing algorithms or models. These algorithms estimate the materials (endmembers) spectra (the pure spectral signatures) and their corresponding abundances (the relative proportions of each endmember in each pixel) from the latent space representation.

1.5.5 Convolution Neural Networks

1.5.5.1 CNN in Computer Vision

When it comes to solving computer vision-related tasks such as image classification, object detection, text parsing, and facial recognition, etc. CNN has made remarkable achievements (see Appendix A). Vanilla neural networks can handle datasets of small sizes, for example, if the image size is $28 \times 28 \times 1 = 784$ (MNIST dataset), this can be manageable. Now, if we have a real hyperspectral dataset like the Urban dataset which consists of 307×307 pixels (height*width) and 168 color channels, by simple multiplication we get 15833832 neurons in the input layer. It becomes impractical to deal with such a large number and comes with an intense computational burden.

Convolution neural networks (CNN) on the other hand solve the computer vision problem [18] on larger datasets by extracting the features of an image and transforming it into a lower dimensional representation while preserving the characteristics present in the original image. CNN is composed of three types of layers: a convolution layer, pooling layers, and a fully connected (FC) layer. A convolution layer is the core part of the network which is responsible for capturing the features of the image with the help of a convolution filter. This convolution filter, also known as *kernel* acts as a scanner and will move across the receptive fields of the image capturing the features along the way. Pooling layers are used to reduce the spatial dimensions of the input while retaining important features, which helps in reducing computational complexity and overfitting. A convolution layer can be followed by additional convolution layers or pooling layers whereas the FC layer is always placed at the end since this layer is responsible for giving out the final predictions. Figure 1.7 shows an image of a bird fed to a convolution neural network for the image classification task.

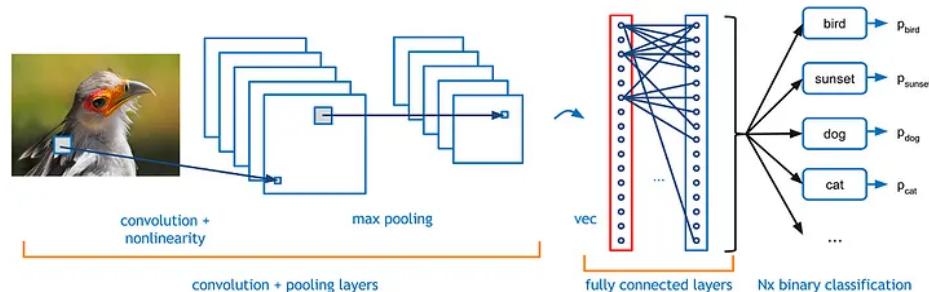


Figure 1.7: Image classification with convolution neural networks

In addition to the layers mentioned earlier, we have size of kernel, striders, sampling layers, activation functions, loss functions, kernel regularizer and kernel initializer which all contribute to the accuracy of the model. The choice of selecting these parameters depends on the nature of the problem. A detailed explanation of these parameters and their selection criteria will be discussed in Chapter 4.

1.5.5.2 Convolutional Autoencoders and Unmixing Problem

Convolutional autoencoders (CAE) [19] use the same approach of image compression and decompression as discussed in Section 1.5.4 and they are architecturally similar. However, CAE utilizes convolution neural networks for both encoder and decoder block instead of simple neural networks. Convolutional autoencoders had shown promising results in the field of remote sensing and hyperspectral unmixing task [20] – [23]. CAE is effective in capturing the spatial correlation of underlying HS image which was completely ignored in conventional autoencoder. When the autoencoders were used to solve the unmixing problem, it was found that the abundance maps naturally arise as feature maps of the bottleneck layer. Similarly, the endmembers can be extracted from the decoder layer's weight matrix given that it is a linear layer. This concept is important and will be further used in our research. Figure 1.8 shows spectral unmixing with convolutional autoencoder.

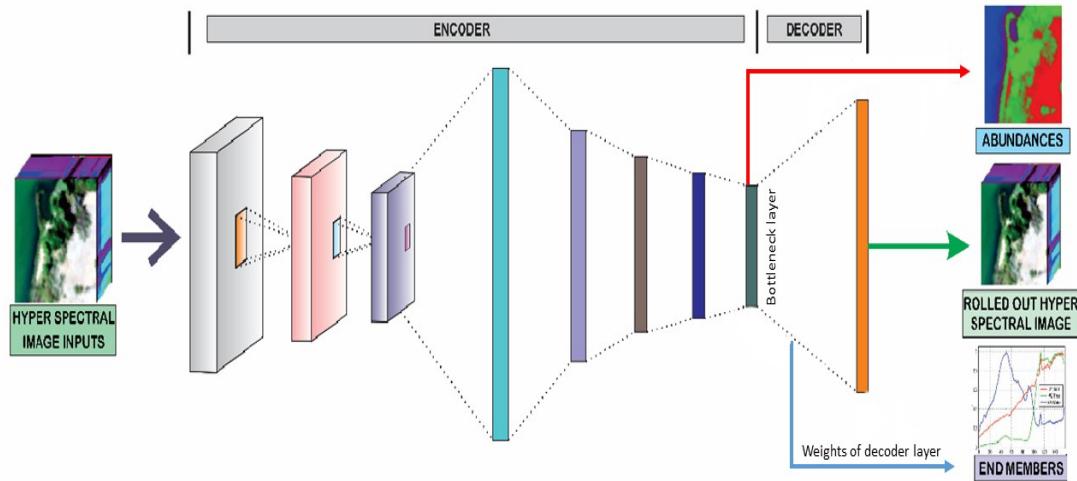


Figure 1.8: Convolutional autoencoder for hyperspectral unmixing

1.5.6 Benefits of Deep Learning Approach in HS Unmixing

Deep Learning has showed remarkable performance in HS unmixing task. The results achieved from deep learning technique verified the superiority of deep learning over the traditional methods. The advantages of using this technique are listed below:

- 1- **No Manual Feature Learning:** Deep learning is capable of automatic feature learning unlike traditional methods that rely on handcrafted feature learning model. The network can train and adjust its weights during training all by itself. It does so to minimize that objective function.
- 2- **Nonlinear Modeling:** Deep Learning can learn nonlinear relationships in hyperspectral data. Unlike traditional methods that often assume the observed data to be linearly mixed which may not accurately represent the real-world data. Deep learning can capture these nonlinearities leading to a better unmixing result.
- 3- **Spatial Information Exploiting:** The traditional methods do not take into consideration the spatial context of an image and only focus on the rich spectral information. However, spatial information is crucial for accurate estimation as the local dependencies present in the HS data can give valuable clues about the materials present in the scene.
- 4- **Robust to Noise:** Hyperspectral data is often subject to various types of noise, such as sensor noise and atmospheric effects, which can degrade the quality of unmixing results. Deep learning models, when properly trained and regularized, can show robustness to noise in the data.

1.6 Conclusion

This chapter gives an overview of hyperspectral images, their applications, and the unmixing problem. We have discussed the traditional approaches that were used to solve the unmixing problem. A brief overview of deep learning, convolution neural networks, and autoencoders for hyperspectral unmixing is presented. At this moment, A challenge that is faced when applying autoencoders for hyperspectral unmixing is developing methods that demonstrate consistent and strong performance across diverse datasets. While some methods may excel on one dataset, they might only yield average results on another. Ensuring the robustness of autoencoder-based techniques remains a critical challenge, and addressing endmember variability is an essential aspect of this pursuit. The fusion of HU and DL is a dynamic and rapidly evolving research field. More autoencoder-based methods are being introduced annually, capitalizing on the latest advancements in DL. This convergence holds promising potential for advancing HU techniques and achieving significant breakthroughs.

Chapter 2: Literature Review

2.1 Introduction

Hyperspectral Cameras (HSCs) provide high spectral resolution of the image which allows material identification. However, the spatial resolution of HSCs is low which causes the spectra observed by HSC to be mixture of several material spectra present in the scene [1]. Therefore, accurate identification requires unmixing. Researchers have developed and examined several models in an effort to find unmixing algorithms that are reliable, stable, robust, and accurate. This chapter gives an overview of unmixing models from the earliest methods of Mustard and Keshava [24] to the present deep learning-based models.

2.2 Stages in Hyperspectral Unmixing

Before we discuss the methods of unmixing, it is better to have some knowledge of the steps involved in unmixing. The traditional methods of HS unmixing must pass through several stages before they can give valuable insights about the data [1]. Figure 2.1 shows steps involved in unmixing: atmospheric correction, dimensionality reduction, unmixing, and inversion. Below is a brief description of each step:

- 1) **Atmospheric Correction:** Due to light attenuation and scattering, the environment significantly affects the measured radiance at the sensor. The procedure of atmospheric correction is used to take these atmospheric factors into consideration. With this adjustment, the radiance data is converted into reflectance, which indicates a fundamental characteristic of the materials in the picture. Note that atmospheric correction is not necessary when performing linear unmixing directly on the radiance data.
- 2) **Dimensionality Reduction:** The space covered by the spectral information in hyperspectral images typically has a much lower dimensionality than the number of bands available. We can efficiently reduce the dimensionality by locating appropriate subspaces, which will improve algorithm performance, simplify the problem, and allow for more effective data storage. Additionally, when the data are accurately modeled by the linear mixture model, the dimension of the signal subspace is one less than the total number of endmembers. This specific dimension is crucial to the hyperspectral unmixing process because it helps to characterize the underlying mixing processes and locate the pure spectral signatures that are present in the scene.
- 3) **Unmixing:** In this step, spectral unmixing will be performed which is to identify the endmembers and their abundance fraction at each pixel. We will discuss three traditional approaches for unmixing in the later section. These include geometrical, statistical, and sparse regression approaches. Additionally, deep learning-based methods will also be covered.
- 4) **Inversion:** In the process of HS unmixing, the inversion step involves working with observed spectral vectors and identified endmembers. The main objective here is to solve a constrained optimization problem to minimize the difference between the observed spectra

and the spectra that we have extracted. This optimization process aims to determine the most suitable fractional abundances for each endmember.

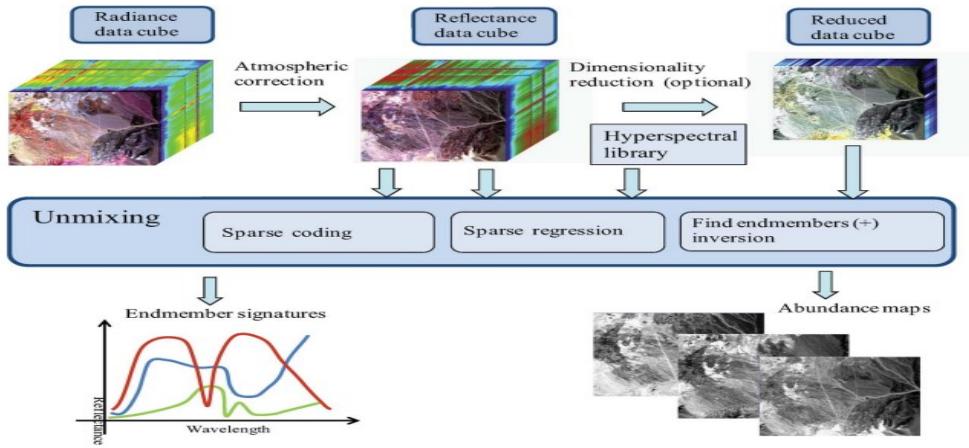


Figure 2.1: Steps involved in unmixing.

To ensure meaningful and physically valid outcomes, fractional abundances are typically subjected to certain constraints. Commonly, they are constrained to be abundance non-negative (ANC), ensuring that negative abundance values are avoided. Additionally, these fractional abundances are constrained to sum to one (ASC), indicating that each pixel's composition is a complete combination of the identified endmembers. While some hyperspectral unmixing techniques separate the endmember determination and inversion into distinct steps, there are also approaches where these steps are simultaneously implemented. In such combined methodologies, the endmembers and fractional abundances are iteratively refined until an acceptable solution is obtained, leading to a comprehensive and accurate unmixing outcome for the hyperspectral image.

It is important to note that these steps are only applicable to traditional approaches and not to deep learning-based algorithms. Deep learning algorithms do need dimensionality reduction and atmospheric correction, but it does not rely on separate mechanisms. They utilize the power of neural networks to implement these steps which will be discussed later.

2.3 Linear and Nonlinear Mixing Models

Hyperspectral unmixing (HU) involves the decomposition of pixel spectra within a hyperspectral image into endmembers, representing pure materials, and abundances, which indicate the proportions of each endmember in a pixel. The selection of endmembers depends on the specific application, introducing variability. Additionally, accurately estimating proportions can be complex, especially in nonlinear cases where highly reflective, small objects may dominate larger, darker ones in a pixel. Nonetheless, studies in hyperspectral research have substantiated the

practical utility of current models over the last decade. Unmixing algorithms are characterized into linear mixing model and nonlinear mixing model [3] as discussed in section 1.2 of chapter 1.

2.3.1 Linear Mixing Model

The linear mixing model (LMM) is a fundamental assumption in HU that underlies many unmixing techniques. It states that the spectral signature of each mixed pixel in a HS image can be represented as a linear combination of the spectral signatures of pure materials, each multiplied by its corresponding abundance or proportion in the pixel. The linear mixing model assumes that the light interaction with the materials in the pixel is additive and that there are no interactions or nonlinearities between the materials. In this case, the mixing phenomenon happens in the instrument's sensor itself and not due to the physical interactions of the material. Figure 2.2 depicts the linear mixing scenario: m_1 , m_2 and m_3 are the materials with corresponding abundances α_1 , α_2 and α_3 respectively. The observed spectra y is the linear combination of the spectra of the materials present in the scene. In other words, the radiance of y can be given as:

$$\text{Radiance}(y) = \text{Radiance}(m1) + \text{Radiance}(m2) + \text{Radiance}(m3)$$

While the linear mixing model is a useful approximation for many scenarios, it may not fully capture all complexities in real-world situations, especially when the interaction of materials is nonlinear or when atmospheric and illumination effects are significant. In such cases, more advanced unmixing techniques, such as nonlinear unmixing or physics-based approaches, may be employed to address these challenges.

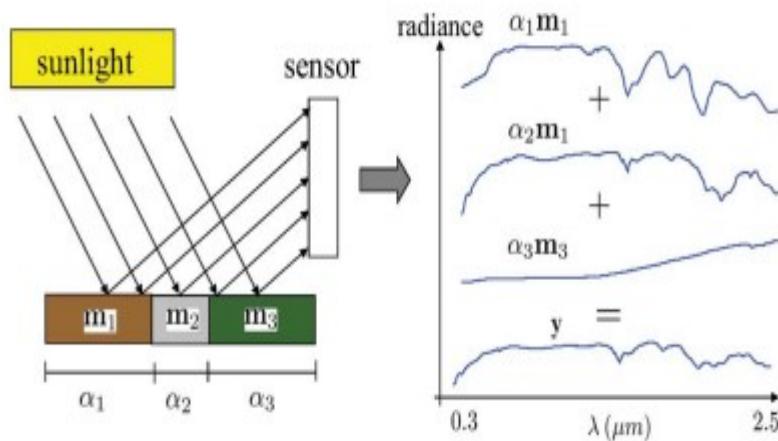


Figure 2.2: Linear mixing. The observed spectra are weighted average of radiances of the materials.

2.3.2 Nonlinear Mixing Model

In hyperspectral unmixing, the nonlinear mixing model (NLMM) is an extension of the linear mixing model (LMM) that considers more intricate dependencies and interactions between the elements in a mixed pixel. Contrary to the linear model, the NLMM assumes that nonlinear combinations of endmembers and their associated abundances produce the observable spectral signature of a mixed pixel [24]. Nonlinear mixing is often caused by physical interactions [1] between the light dispersed by various materials present in the scene. Figure 2.3 shows the interaction of light with materials in the LMM and NLMM scenario.

Nonlinear unmixing techniques aim to estimate both the endmembers and their corresponding abundances while considering the nonlinear relationship between them. These methods often rely on optimization algorithms and advanced mathematical models to infer the most likely nonlinear mixing processes occurring in the observed data.

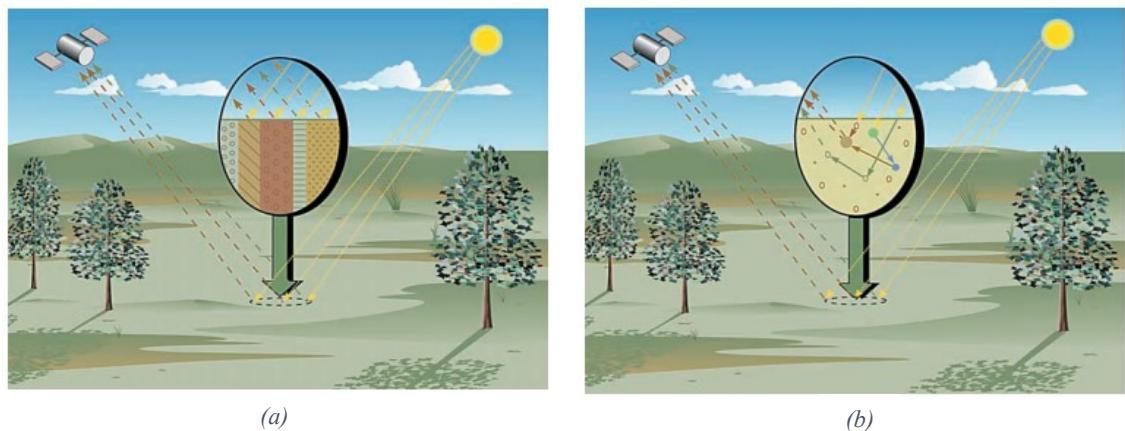


Figure 2.3: (a) illustration of linear mixing. (b) illustration of nonlinear mixing.

It is unclear whether linear or nonlinear processes predominate in the spectral signatures of mixed pixels, an issue that is probably impacted by a number of variables and scene settings. While the linear technique is useful for analyzing the variability of remote sensing data and producing data products on material abundance, its strict validity is only applicable to situations where endmembers are discretely and individually dispersed over the surface, which is uncommon in natural settings. In research involving the earth sciences, relevant elements frequently coexist in soils or are closely related on smaller scales. Finally, LMM is widely used to represent a mixed pixel using distinct combination of endmembers as it is simple and yet powerful model to formulate the unmixing problem. Many state-of-the-art algorithms rely on linear mixing assumption which will be discussed in section 2.4.

2.4 Categorization of Hyperspectral Unmixing Models

Methods for hyperspectral unmixing are essential for obtaining useful information from hyperspectral data. The authors of [1] divided unmixing techniques into three main categories: *statistical*, *geometrical*, and *sparse regression*. Recently, deep learning-based algorithms have shown remarkable results and outperformed all the prior methods of unmixing. Each category offers a unique approach for overcoming the difficulties of separating mixed pixels and figuring out the underlying materials present in HS images. We will examine the specifics of these four HSU classifications in this section as well as the state-of-the-art algorithms of each category.

2.4.1 Geometrical Based Approach for HSU

The geometrical approach is further categorized into two main categories: Pure Pixel (PP) and No Pure Pixel (NPP) which is also referred to as Minimum Volume (MV) based (see Figure 2.4). PP based algorithms rely on the assumption that there is at least one pure pixel present in the observed image. If the pixel contains only one type of material, then we say that the pixel is pure. On the other hand, NPP does not rely on the presence of pure pixel which is common in real world as there are multiple materials closely placed together.

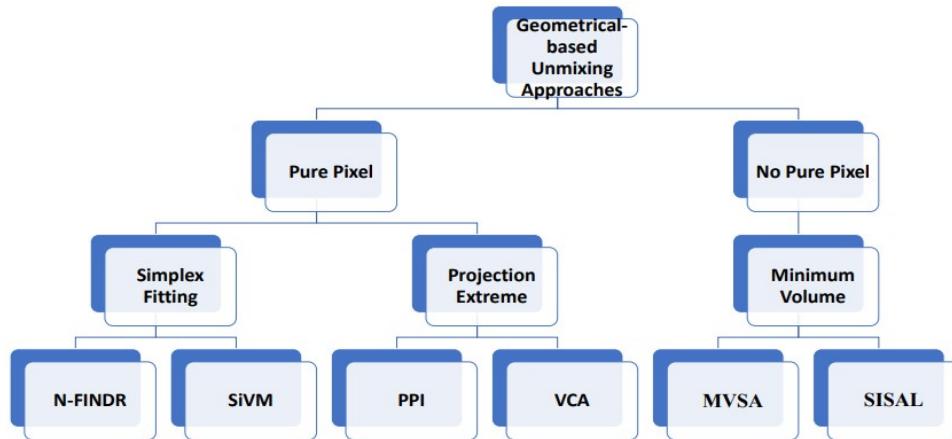


Figure 2.4: Categorization of geometrical based approach.

In such cases, the geometrical approach relies on identifying "virtual" or "partially" pure pixels, which are combinations of endmembers that are unlikely to occur naturally but provide critical reference points for unmixing. These virtual pure pixels are selected to create the convex hull, and the mixed pixels are projected onto it to estimate the endmember abundances.

2.4.1.1 Pure Pixel (PP) Based Algorithms

A few representative algorithms of this class are N-FINDR [25], Vertex Component Analysis (VCA) [8], Iterative Error Analysis (IEA) [26], and Pixel Purity Index (PPI) [27,28]. Here, we will only discuss VCA and N-FINDR as these algorithms are sufficient to give a good overview of the approach followed by PP based algorithm.

2.4.1.1.1 Vertex Component Analysis (VCA) for HSU

This algorithm relies on the following assumption: first the light only interacts with the one material and hence linear mixture model is assumed. Second, there is at least one pure pixel present in the image which purely belongs to a single class of material. This method works by exploiting two facts: 1) the vertices of the simplex represent pure spectra and 2) simplex when transformed preserves collinearity so the affine transformation of a simplex remains a simplex. In HSU, a simplex refers to the convex hull formed by a set of endmembers and their corresponding data points in the hyperspectral feature space. The convex hull represents the smallest convex region that encloses all the data points. The algorithm first determines the initial set of endmembers by random selection, manual selection, or by using algorithms like Spectral Angle Mapper (SAM) [29], Maximum Spectral Response Selection [30], and Endmember Purity Measures [31] etc.

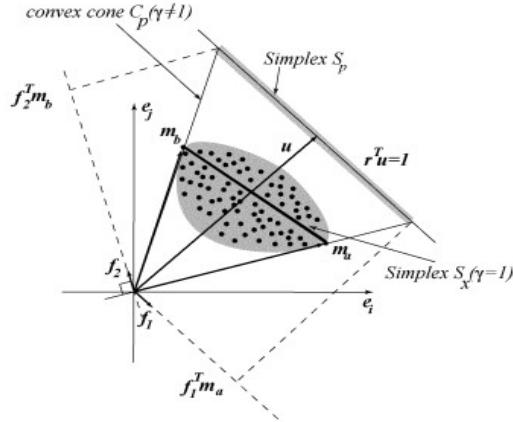


Figure 2.5: Illustration of VCA.

After endmember selection, the algorithm projects the remaining data points onto a direction that is orthogonal (perpendicular) to the subspace spanned by the identified endmembers in each iteration. This direction is calculated to maximize the distance of the data points from the subspace, ensuring that the projected data point represents a new endmember candidate. The endmember candidate corresponding to the extreme point of the projection (i.e., the point farthest from the subspace) is identified as the new endmember signature. The algorithm continues this iterative process until all the pure spectra (endmembers) are identified exhaustively, meaning that the projection process

no longer yields new endmember candidates. Figure 2.5 illustrates the concept of vertex component analysis. After the pure materials are identified using VCA. Now is the time to determine abundance fractions of each material.

VCA does not directly determine the abundance fraction, instead it uses Dirichlet distribution for estimating the abundances (see [8] for details). To check the performance of algorithm, a comparison of endmembers extracted using VCA is done with laboratory spectra [32]. Spectral Angle Distance (SAD) metric is used to evaluate the performance (see Section 2.5). The experiment was performed against 12 endmembers by the authors in [8] but we will only mention 3 of them namely Alunite, Nontronite, and Sphene. The SAD value achieved for Alunite = 4.1, Nontronite = 3.9 and Sphene = 3.1 as described in the original paper [8]. Although VCA is a very effective algorithm for unmixing and provides valuable information about the scene, it relies on the presence of pure pixel which may not be true in real world scenarios. Nevertheless, this algorithm is still employed in recent work as an initial stage to determine the number of endmembers.

2.4.1.1.2 N-FINDR Algorithm for HSU

This algorithm also follows the same assumption as followed by VCA (described in section 2.4.1.1.1). The first step of this algorithm is a preprocessing step which is to reduce the dimensionality of HS image to $n-1$ where n is the number of endmembers in the scene. This is done via an orthogonal subspace projection as described in [25]. The endmember extraction process begins with the random selection of a set of pixels as initial endmembers. To refine the endmember estimates, each pixel in the image is evaluated to determine its likelihood of being a pure or nearly pure pixel. This evaluation involves calculating the volume with each pixel in place of each endmember. If replacing an endmember with a pixel result in an increase in volume, the pixel replaces the endmember. The process continues iteratively until no further replacements of endmembers occur. After identifying endmembers, their spectral signatures are used to unmix the original hyperspectral image. This process generates a set of images, each representing the contribution of a specific endmember for every pixel in the original image. To determine the endmember contributions during unmixing, two common inversion procedures are employed: 1) Least Squares Inversion [33] and 2) Non-Negatively Constrained Inversion [34]. These algorithms are too complex to detail here, but their description is provided in [33] and [34].

This algorithm works well in case of presence of pure pixel, but if there is no pure pixel in the image which is common in real world data, then the basic assumption of this algorithm is breached. This results in mixed pixel selected as pure pixel. Moreover, this algorithm selects the mixed pixel with the highest brightness as pure pixel instead of the actual pure pixel with lower brightness. This defect was somewhat mitigated by the authors by enforcing the requirement that in order for a pixel to be selected as pure pixel, its brightness must be increased as the spectral angle between it and the true

endmember increases. Finally, to evaluate the performance of N-FINDR, the authors of [8] also performed experiments on the same dataset with 12 endmembers used for VCA. We will consider the same 3 endmembers (Alunite, Nontronite, Sphene) for the sake of comparison and fairness. The experimental results using SAD are as follows: Alunite = 4.0, Nontronite = 4.0 and Sphene = 2.9 as provided in [8]. Despite the difficulties faced by the algorithm mentioned above, it is still an effective tool to unmix the image and has shown robustness when fed with noisy data.

2.4.1.2 Minimum Volume (MV) Based Algorithm

The algorithms of this class do not rely on the presence of pure pixel assumption but do assume linear mixing. The concept of MV refers to the idea of finding the smallest possible simplex that contains all the pixels of HS image (see Figure 2.6). We consider Minimum Volume Simplex Analysis (MVSA) [9] the state-of-the-art algorithm representing this class.

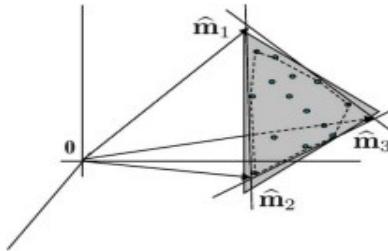


Figure 2.6: Concept of minimum simplex containing all the pixels.

The algorithm of this class seeks to solve the below optimization problem which is given as:

$$\min_{M,A} \|Y - MA\|_F^2 + \lambda V(M) \quad (1)$$

$$\text{subject to: } A \geq 0, 1_p^T A = 1_n, \quad (2)$$

The optimization problem in Eq (1) consists of two parts: the data term (loss function) which is responsible for minimizing the reconstruction error between the ground truth and extracted endmembers, and the volume term aims to provide mixing matrices of minimum volume with λ controlling their relative weight. The Eq (2) are physical constraints that must be enforced on the abundance maps. These are abundance sum-to-one (ASC) constraint and abundance non-negative constraint (ANC). The constraints given in Eq (2) also apply to algorithms based on pure pixel assumption (VCA and NFINDR).

If the HS data does not have pure pixels, then the foundation of pure pixel algorithms [8,25] is violated and the algorithms fail to extract any valuable information from the HS data. However, if there are at least $n-1$ spectral vectors on each facet of the data simplex as shown

in Figure 2.7, then we might be able to fit a MV simplex to the data. The MVSA [9] algorithm works by solving the optimization similar to Eq (1) while relaxing the physical constraints given in Eq (2). The actual optimization equation solved by MVSA can be found in [8]. The existence of noise or other sources of perturbations can lead to spectral vectors being positioned outside the true data simplex [1]. To grasp the significance of this violation, noisy data is depicted in Figure 2.8. The dashed line is the estimated minimum simplex when the constraints are not violated which is far from the original.

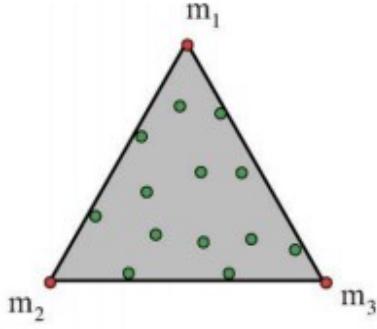


Figure 2.7: $n-1$ spectral vectors on each facet of data simplex.

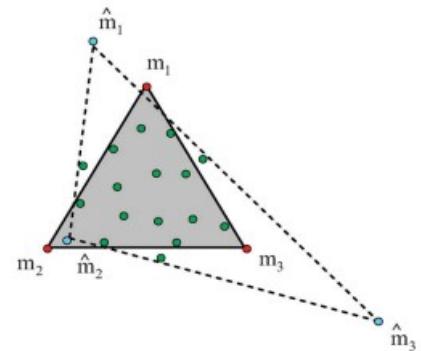


Figure 2.8: Noisy data. Dashed line represents the estimation when physical constraints are enforced.

Therefore, to make an accurate estimation, the MVSA algorithm allows violation of constraints yielding a simplex very close to the original. The hard constraint is replaced with soft constraint called the hinge function ($\text{hinge}(x) = 0 \text{ if } x \geq 0 \text{ and } -x \text{ if } x < 0$). The authors of [9] however, addressed only the hard constraint as their objective was to develop fast algorithm that could provide not the best but “good” or suboptimal results. To evaluate the performance of the algorithm, spectral angle distance (SAD) is used as the performance metric. The experiment is performed on a well-known Cuprite dataset provided by the AVIRIS laboratory. A 250×190 sub-image is used containing 224 spectral bands and 14 types of minerals. The results achieved by this method were similar to state-of-the-art algorithms of this class but in a more computationally efficient manner. The time (in seconds) taken by this algorithm to converge is 149.93s while MVES [35] and MVC-NMF [36] took 24909s and 2896s respectively.

Despite yielding suboptimal solutions, MVSA offers a powerful and efficient approach to hyperspectral unmixing. By iteratively identifying endmembers and refining abundance fractions, MVSA can accurately estimate the materials present in a hyperspectral image. Its ability to find the smallest simplex volume that encapsulates the data points leads to robust and compact representations.

2.4.2 Statistical-Based Approach for HSU

In the presence of pure pixels, algorithms like VCA [8] and NFINDR [25] are applicable. If there are no pure pixels in the data but enough spectral vectors on each facet of the

simplex, then algorithms like MVSA [9], MVC-NMF [36], and MVES [35] can be employed. But if the data is highly mixed i.e., no pure pixels and not enough spectral vectors on the simplex facet, then all the mentioned algorithms fail to provide any valuable information about the data. In such cases, the statistical framework is an effective alternative, which, when compared to geometrical methods requires more computations. This method treats the unmixing problem as a statistical inference problem [37] usually adopting the Bayesian view. The goal is to estimate the posterior density of the random objects of interest, which in this case are the endmember matrix and the abundance fractions. The Bayesian approach, particularly demonstrated by the Dependent Component Analysis (DECA) algorithm [38], showcases its effectiveness in handling highly mixed data sets in hyperspectral unmixing. The Bayesian inference engine of DECA considers both the likelihood of observing the data given the model and the prior knowledge about the endmembers and abundance fractions. By incorporating prior information about the abundance fractions, such as non-negativity and constant sum constraints, DECA can automatically enforce physically meaningful solutions during the unmixing process. Furthermore, DECA handles the inherent uncertainties and noise in the data by estimating not only the endmembers and abundance fractions but also the noise covariance matrix. This additional step allows DECA to provide more accurate and reliable unmixing results, even in the presence of noise and highly mixed data.

The statistical-based algorithms are very complex when compared with other approaches for unmixing. This is because of high-dimensional data, uncertainty modeling, prior selection, iterative Bayesian inference, and model complexity. The work [12] and [39] – [41] all offer different flavors of Bayesian approach. We have not included any mathematical equations of this approach for the sake of simplicity, but its detail can be found in the work [1].

2.4.3 Sparse Regression Based Approach for HSU

This approach assumes that the observed mixed spectra is a linear combination of already known spectral signatures from the spectral library [4], [42] – [43]. This approach aims to find a number of spectra from the spectral library that can best represent the observed mixed spectra. The spectral library provided by National Aeronautics and Space Administration (NASA) [2] contains a huge collection of pure spectral signatures of materials found on Earth.

Let's assume the endmembers that were originally retrieved or created from the hyperspectral data are no longer needed to address the mixture problem. They are instead chosen from a pre-existing spectral library (A) which has a high number of spectral samples (m) that were previously available. Finding the best selection of samples from the library to represent each mixed pixel in the hyperspectral scene is the goal of the unmixing procedure in this case. An underdetermined linear problem results when there are more samples in the library (m) than there are bands in the data (B). Now, let "x" be the fractional

abundance vector with respect to library A. With these definitions, the sparse regression problem for hyperspectral unmixing can be formulated as follows:

$$\min \|x\|_0 \text{ subject to } \|y - Ax\|_2 \leq \delta, x \geq 0, \quad (3)$$

in Eq (3), $\|x\|_0$ represents the number of non-zero components in the vector x , and $\delta \geq 0$ is an error tolerance parameter that accounts for noise and modeling errors. The Eq (3) is known to be NP-hard problem [44], making it computationally challenging to find an exact solution directly. However, there are alternative approaches that offer computationally feasible solutions to obtain the sparsest representation. Greedy algorithms, such as Orthogonal Basis Pursuit (OMP) [45], Basis Pursuit (BP), and Basis Pursuit Denoising (BPDN) [46], are three commonly used techniques to approximate the sparse solution efficiently. In both BP and BPDN, the original ℓ_0 norm in Eq (3) is replaced with the ℓ_1 norm given as:

$$\min \|x\|_1 \text{ subject to } \|y - Ax\|_2 \leq \delta, x \geq 0, \quad (4)$$

Now, Eq (4) is a convex optimization task that enables the use of efficient numerical optimization techniques to find approximate sparse solutions. This approach has two limitations: first the observed mixed spectra usually contain 3 to 4 spectral signatures which is a very small number when compared to huge spectral library containing thousands of signatures of distinct materials. This makes it harder to find the right subset of endmembers from spectral library. Secondly, the observed mixed spectra contain noise which may alter the reflectance curve of a material making it represent a different material's spectra.

Unmixing using sparse regression is a very active research field as more and more materials are being added to the already vast spectral library. In recent literature, researchers have combined sparse regression approach with neural networks [47] to yield useful unmixing results.

2.4.4 Summary of Traditional-Based Approaches

Before moving onto a deep learning-based approach, let us summarize all the traditional methods that we have discussed in the previous sections. The traditional approach is categorized into geometrical, statistical, or sparse regression-based approaches depending on the nature of the data. The geometrical approach is further categorized into pure pixel or no pure pixel (minimum volume). The algorithms are categorized due to the following four situations (see Figure 2.9) that may occur:

- 1) There is at least one pure pixel on each vertex of data simplex (pure pixel assumption).
- 2) There are no pure pixels but at least contains $n-1$ spectral vectors on each facet of data simplex (minimum volume simplex).
- 3) The data is highly mixed i.e., no pure pixel and insufficient spectral vectors on each facet. The problem is then formulated as a statistical inference problem.
- 4) The data may be represented as a linear combination of the already known endmembers from the spectral library.

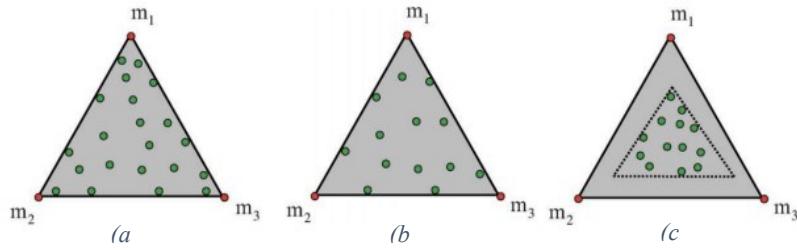


Figure 2.9 (a) Pure pixel scenario. (b) $n-1$ spectral vectors on each facet. (c) No pure pixel and insufficient spectral vectors on each facet

2.4.5 Deep Learning Based Algorithms in HSU

In recent years, deep learning approaches have shown promising results in various image processing tasks, including hyperspectral unmixing. Deep learning utilizes the power of neural networks to learn complex features from HS image. In the early stages of using neural networks for spectral unmixing, a two-stage approach was commonly employed. This approach involves separating the process into two distinct steps: feature extraction and unmixing [16]. An early attempt was made [48] using an auto-associative network for feature extraction and a multilayer perceptron (MLP) for abundance estimation and pixel-based fuzzy classification in HS unmixing. A compressed representation of the input was produced by training the auto-associative network to extract relevant features from the HS data. The abundance fractions of endmembers in each pixel were then estimated by the MLP using these extracted features. The network was trained using information that was already available, such as the number of endmembers and their signatures (reflectance curve). The method showed promising results when estimating abundance maps using the INTA-AHS, CHRIS-PROBA, and AVIRIS Cuprite datasets. Despite the early success of this method, there have since been additional developments in the field, such as more complex end-to-end deep learning models for hyperspectral unmixing.

As discussed already in section 1.5.3 of chapter 1, three widely used deep learning architectures used to perform HSU are generative model, autoencoders and convolutional neural network. This section will focus on state-of-the-art algorithms that belong to the autoencoder class since it is the most explored architecture among all.

An autoencoder is composed of an encoder block, and a decoder block, Encoder block f_E aims to compress the original HS image \mathbf{X} into a latent space v (bottleneck layer) where $v = f_E(\mathbf{X})$. The decoder block f_D receives the output of encoder as input and decompresses it to reconstruct the original image $\hat{\mathbf{X}}$ where $\hat{\mathbf{X}} = f_D(v)$. The network is trained in a way that it minimizes the loss function $Loss(\mathbf{X}, \hat{\mathbf{X}})$ between the input image \mathbf{X} and reconstructed $\hat{\mathbf{X}}$. The weight matrix of decoder block usually represents the pure spectra while the output of bottleneck layer provides abundance estimation. Initial work using autoencoder [49] used autoencoder cascade for unmixing. The method consists of marginalized denoising autoencoder (mDA) and non-negative sparse autoencoder. The mDa, unlike conventional

denoising autoencoders [50] aims to improve the encoding process by effectively handling noise in the data. It achieves this by integrating out the noise distribution and determining optimal weights for a linear transformation to reconstruct the original pure signals. The output of mDA is given as input to non-negative sparse autoencoder for unmixing. The experiment was performed on synthetic as well as real hyperspectral dataset and has shown great results. The work was later improved in [51] by exhibiting three distinct characteristics of the proposed autoencoder. Firstly, the network's encoder and decoder operate independently, with only the decoder needing to satisfy nonnegativity constraints. Secondly, rather than using a denoising layer as a preprocessing step, a denoising constraint is directly imposed on the decoder of the network to reduce the reconstruction error. Third, l_{21} -norm is imposed to accurately determine the number of endmembers by removing the redundant rows from the endmember matrix. The approach has shown robustness against extremely noisy data, i.e., SNR up to 1 dB. A sparse autoencoder is primarily used for endmember extraction in [52]. The autoencoder network has two stages, more deep layers, and a better loss function that incorporates the Kullback-Leibler divergence function. In comparison to state-of-the-art methods, the authors reported competitive results. On the other hand, in [53], the authors created deep autoencoder networks specifically to deal with the problem of outliers in hyperspectral data during blind linear spectral unmixing. The technique makes use of preliminary estimations of spectral signatures discovered by the stacked autoencoder. Following that, a variational autoencoder uses these estimates to jointly calculate endmembers and abundances. The variational autoencoder makes sure that the estimated abundances comply with physical restrictions like nonnegativity and sum-to-one constraints.

The next deep learning architecture is convolution neural networks (CNN) for spectral unmixing. CNN has already shown great results in computer vision problems such as image classification, text parsing and facial recognition [18]. As depicted in Figure 2.10, a deep CNN is composed of multiple convolutional layers and nonlinear logistic function. The input data is sequentially passed through the layers, where the output of the previous layer serves as the input to the subsequent layer in the sequence. A deep Convolutional Neural Network (CNN) can learn the most intrinsic features of the data through its hierarchical layers, and it provides a more refined decision in the last fully connected layer (dense layer).

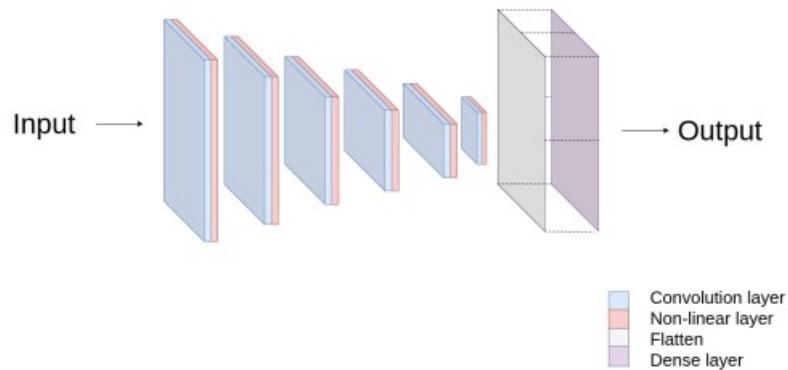


Figure 2.10: Illustration of CNN architecture.

The authors of [23] proposed an end-to-end Deep Convolution Neural Network (DCAE) for linear HS unmixing. The DCAE model is made up of two components: a non-linear vanilla decoder that maps the extracted features to the abundances, and a linear deep encoder that extracts a deep and non-redundant feature vector. They used a threshold layer and normalization layer to enforce abundance non-negativity (ANC) and sum-to-one constrain (ASC) respectively. Once the network is entirely trained, DCAE can effectively encode and decode the HS image. The performance is evaluated on synthetic as well as real HS data. Also, the experiment is performed using four different loss functions which will be discussed in the next section (Section 2.5). The HS image provides rich spectral information along spatial information which is crucial for achieving accurate unmixing results. All the above-mentioned algorithms did not consider spatial information and only relied on spectral information for unmixing. This issue was addressed recently by the authors of [54] by strictly using convolutional blocks for both the encoder and decoder. Moreover, they refrain from using any pooling layers or upsampling layers to preserve the spatial structure of the data. Finally, they utilized patch-wise learning instead of pixel-wise learning to retain the spectral-spatial nature of the data and make the training process more efficient. This method achieved results superior to all previously mentioned algorithms on four real hyperspectral datasets setting a new standard for comparison.

2.5 Performance Metrics for Quantitative Measurement of Algorithms

To evaluate the effectiveness of an algorithm and the accuracy of a model in solving a specific problem, quantitative measure called the performance metric is used. In hyperspectral unmixing, several performance metrics are commonly employed to assess the quality of unmixing results. This section will discuss common performance metrics used by the researchers for the model evaluation.

2.5.1 Spectral Angle Distance (SAD)

SAD is a common metric used in hyperspectral unmixing to quantify the similarity between two spectral signatures. It measures the angle between two spectral vectors in the hyperspectral data space. It is given by the following equation:

$$SAD(X, \hat{X}) = \cos^{-1} \left(\frac{\hat{x}^T x}{\|\hat{x}\| \|\hat{x}\|} \right) \quad (5)$$

In eq (5), x and \hat{x} are ground-truth and extracted endmembers respectively.

2.5.2 Root Mean Squared Error (RMSE)

In HSU, RMSE is commonly used to assess the quality of abundance maps, which represent the estimated fractions of endmembers in each pixel of the hyperspectral image. RMSE is calculated by the following formula:

$$RMSE(s, \hat{s}) = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{s}_i - s_i)^2} \quad (6)$$

In eq (6), s_i is the ground-truth abundances of j^{th} endmember and \hat{s}_i is the extracted abundances.

2.5.3 Mean Squared Error (MSE)

The MSE loss function is similar to RMSE function given in eq (6). It is computed by the formula:

$$MSE = \frac{1}{N} \sum_{j=1}^N (\hat{x}_j - x_i)^2 \quad (7)$$

In the above equation, \hat{x}_j is the reconstructed pixel and x_i is the original pixel.

2.5.4 Spectral Information Divergence (SID)

SID is based on the Kullback-Leibler divergence, also known as relative entropy, which is a mathematical concept used to compare two probability distributions. It is given as:

$$SID = \frac{1}{\rho} \sum_{\rho=1}^{\rho} \sum_{n=1}^B \rho_n \log \left(\frac{\rho_n}{q_n} \right) + \sum_{n=1}^B q_n \log \left(\frac{q_n}{\rho_n} \right) \quad (8)$$

Where

$$\rho_n = \frac{x_{i,n}}{\sum_{k=1}^M x_{i,k}}, q_n = \frac{\hat{x}_{i,n}}{\sum_{k=1}^M \hat{x}_{i,k}} \quad (9)$$

where ρ_n and q_n are the probability vectors for the spectral signatures $x_{i,n}$ and $\hat{x}_{i,n}$ respectively.

2.6 Conclusion

In this chapter, we discussed the steps involved in spectral unmixing. A detailed explanation of linear mixing and non-linear mixing model is presented. We also discussed the three main categories of traditional models along with the full explanation of state-of-the-art algorithm of each category. A review of deep learning based autoencoders is presented. Finally, we discussed the commonly used performance metrics for both the endmember matrix and abundance matrix. The collection of methods described in this study showcases how the field is becoming more advanced and sophisticated. It is a rapidly growing area that brings together various disciplines, such as signal and image processing, physics, linear algebra, and computer science. In the future we will conduct an intensive experimental comparison of some of the techniques discussed in this work.

Chapter 3: Problem Statement for HS Unmixing

3.1 Introduction

In image processing, an image consists of a collection of square pixels, organized in a grid-like structure of rows and columns. In simple words, an image is represented by a matrix, or an array. Each entry of an image matrix is a pixel representing a tiny portion of a picture, contributing to the overall visual representation. The pixel value ranges from 0 to 255 where 0 is taken for black and 255 is taken for white. A 2-dimensional matrix is used to denote a grayscale image while 3-dimensional is used for an RGB image. The hyperspectral (HS) image is also a 3-dimensional matrix but has a greater number of channels than any other type of image.

3.2 Problem Definition and Notations

Given a HS image, the goal of unmixing is to identify the materials (endmembers) present in the image along with their fractional abundances. This is achieved by finding the right entries for both the endmember matrix and the abundance matrix i.e., when dot operation is performed between the matrices, it yields the original HS image. In our work, we assume that each pixel of the observed HS image is a linear combination of endmembers (y_j), i.e, incident light only interacts with just one material. The LMM is given by the following equation:

$$X = Ya + n, \quad (1)$$

The notations are as follows, $X \in \mathbb{R}^{B \times N}$ is the observed hyperspectral image having B bands and $N = w \times h$ represents the total number of pixels present in HSI, $\mathbf{Y} = \{y_j : j = 1, \dots, P\} \in \mathbb{R}^{B \times P}$ where P denotes the number of endmembers in each column and matrix $\mathbf{a} = \{\alpha_{n,j} : j = 1, \dots, P\} \in \mathbb{R}^{P \times N}$ is the abundance fraction of j^{th} endmember at the n^{th} pixel, where $n = 1, \dots, N$. Finally, \mathbf{n} is the noise matrix that is added to account for the noise (sensor noise, atmospheric effects, etc.) present in the observed data. At a given pixel, the a_j represents the fractional area occupied by the j th endmember. Therefore, the abundance matrix is subject to **Abundance Sum-to-One Constraint (ASC)** and **Abundance Nonnegative Constraint (ANC)**.

3.3 Physical Constraints on Abundance Matrix

3.3.1 Abundance Sum-to-One (ASC)

The sum-to-one constraint reflects the physical nature of matter that each material's contribution when combined will result in one. In LMM, we assume that each pixel of an image is a linear combination of endmembers. Therefore, their sum must be equal to one. This constraint must be held for each pixel in the image. Mathematically, it is given as:

$$\text{Abundance Sum-to-One Constraint (ASC)} \quad \sum_{j=1}^P a_{n,j} = 1 \quad (2)$$

The constraint ensures that the estimated abundances are interpreted as proportions or fractions of the pixel's composition. This aligns with the idea that the pixel is a mixture of different materials.

3.3.2 Abundance Nonnegative (ANC)

The nonnegativity is another constraint that is imposed on abundance fractions. This constraint is also derived from the physical nature of matter that abundance fraction of a material cannot be negative as negative value would imply “absence of material” which is physically meaningless. In simple words, it enforces the presence of at least one material in the image. This constraint must be held for each pixel in the image. Mathematically, it is given as:

$$\text{Abundance Nonnegativity Constraint (ANC)} \quad a_{n,j} \geq 0 \quad (3)$$

In short, the constraint aligns with the notion that abundance values represent proportions or fractions of endmembers in the mixed pixel. Negative abundances would lack a physical interpretation.

3.4 Conclusion

In this chapter, a brief introduction of image representation in image processing is given. The mathematical equation of LMM is given along with the necessary notations. The description of physically enforceable constraints namely the abundance sum-to-one constraint (ASC) and abundance nonnegative constraint (ANC) is presented. These constraints are crucial for meaningful interpretation of abundance fractions. If the constraints are not enforced, it will result in physical inconsistency, algorithm instability, meaningless outcomes, and unreliable interpretation. Hence, it is important that every pixel in the HS image satisfies the aforementioned constraints.

Chapter 4: The Proposed Method for Hyperspectral Image Unmixing

4.1 Introduction

The proposed method for HS unmixing operates within the realm of deep learning. The motivation for opting for a deep learning approach in our work lies in the unparalleled capabilities demonstrated by deep neural networks. Despite being simple, deep learning has proven effective in automatically capturing intricate patterns and non-linear relationships within HS data. In this work, we assume linear mixing model and the equation of LMM is given in Section 3.2. The model is unsupervised such that both the endmembers and abundance maps are to be found. Please note that “unsupervised” in the domain of unmixing does not refer to any kind of unlabeled data, but it refers to extraction of both the endmembers and their abundance. Lastly, the model performs blind unmixing such that the endmembers matrix and abundance maps will be extracted simultaneously.

4.2 Residual-Convolutional Autoencoder (RCAE)

The proposed Residual Convolution Autoencoder is composed of an encoder block, a decoder block, and residual connections which are also referred to as skip connection. Encoder block f_E aims to compress the original HS image X into a lower-dimensional space called the latent space v where $v = f_E(X)$. The decoder block f_D receives the output of encoder as input and decompresses it to reconstruct the original image \hat{X} where $\hat{X} = f_D(v)$. The network is trained in a way that it minimizes the loss function $Loss(X, \hat{X})$ between the input image X and reconstructed image \hat{X} . In this section, we discuss the proposed (RCAE) for HS unmixing as depicted in Figure 4.1.

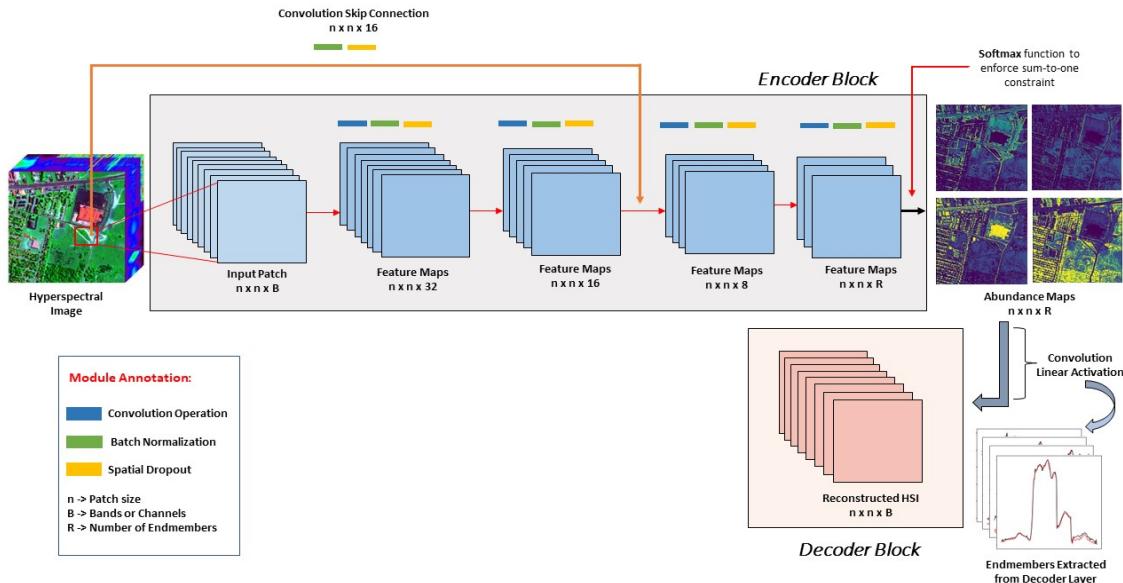


Figure 4.1: Proposed RCAE Network

Once the entire network is trained on real hyperspectral data, it can effectively encode and decode the input data. Given linear activation for decoder part, we can interpret the latent space v (output of encoder) as the estimated abundance maps and the weight matrix of the decoder layer (last layer) as endmembers matrix.

4.2.1 Encoder Block

The hyperspectral dataset is divided into K patches, $\beta_i = \{x_1, \dots, x_{n \times n}\}, i = 1 \dots, K$ and $n \times n$ is the patch size. We have adopted a deep network layer for encoder module consisting of one input layer, four 2D convolution layers, four batch normalization layers, 4 spatial dropout layers, and one convolutional skip connection. As shown in Table 1, the

Table 4.1. Layers in the proposed encoder module

<i>Layer Number</i>	<i>Layer Type</i>	<i>Activation Function</i>	<i>Kernel Size</i>	<i>Output Feature Maps</i>
1	Input	-	-	-
2	2D Convolution	LeakyRelu	3×3	32
3	Batch Normalization	-	-	-
4	Spatial Dropout	-	-	-
5	2D Convolution	LeakyRelu	3×3	16
6	Batch Normalization	-	-	-
7	Spatial Dropout	-	-	-
8	2D Convolution	LeakyRelu	1×1	8
9	Batch Normalization	-	-	-
10	Spatial Dropout	-	-	-
11	2D Convolution	softmax	1×1	P
12	Batch Normalization	-	-	-
13	Spatial Dropout	-	-	-
14	2D Convolution Skip	LeakyRelu	1×1	16

first layer of the module is the input layer which receives an input patch $\beta_i \in \mathbb{R}^{n \times n \times B}$ from the HS image X where \mathbb{R} is real number. The second layer is a convolutional 2D layer (C1) having 32 feature maps, kernel size of 3×3 and LeakyRelu activation function. After C1 layer, we apply batch normalization to improve the training process and spatial dropout layer to avoid overfitting of the model. The dropout layer is set to 0.2 meaning 20% of entire feature maps are randomly set to zero. Spatial dropout helps introduce diversity during training by temporarily removing certain spatial activations, preventing the network from relying too heavily on specific features and improving its ability to generalize data. The next layer is another convolutional 2D layer (C2) with 16 feature maps, kernel size of 3×3 and LeakyRelu activation. After this layer, we again apply batch normalization layer

and spatial dropout layer. The next convolutional 2D layer (C3) is composed of 8 feature maps, 1×1 filter size and LeakyRelu activation, which is again followed by batch normalization and spatial dropout. Finally, the last convolutional layer of encoder block is the bottleneck layer having feature maps set equal to number of endmembers P present in the input HS image and the filter size is set to 1×1 . Batch normalization and spatial dropout layer is applied for the last layer. As previously mentioned, this bottleneck layer's output will be used to extract abundance maps. Therefore, we need to enforce physical constraints discussed in the previous chapter (see Section 3.3) which are abundance sum-to-one (ASC) and abundance non-negativity constraint (ANC). To enforce the ASC, we have applied softmax activation function to the output of bottleneck layer which results in P feature maps having values ranging from 0 to 1 which can be interpreted as the percentage (proportion) in which each material is present on the surface. Moreover, batch normalization in addition to overcoming covariate shift problem, also enforces ASC and the layer's output can be defined as the following:

$$y_i = \gamma \bar{x}_i + \beta \quad (1)$$

where

$$\bar{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma^2_\beta + \epsilon}} \quad (2)$$

$$\mu_\beta = \frac{1}{m} \sum_{i=1}^m x_i \quad (3)$$

$$\sigma^2_\beta = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_\beta)^2 \quad (4)$$

The Equation (2) normalizes the layer inputs, Equation (3) is the mean of batch with m samples, and Equation (4) is the batch variance. The γ and β are parameters learned during training along with the original parameters of the network and ϵ is a very small number added to avoid division by zero.

4.2.2 Decoder Block

The last layer of the network is the decoder layer. We have used yet another 2D convolutional layer for the decoder part with linear activation function. This convolutional layer with B feature maps where B corresponds to number of bands in HS image and $s \times s$ filter size where s is an odd number will reconstruct the input patch passed from the previous ASC enforcing layer.

4.2.3 Residual Connection

Residual connections involve additional links within a neural network that connect certain layers to others that are not immediately adjacent. In a standard convolutional neural network (CNN), each layer typically receives input solely from the preceding layer and transmits output exclusively to the subsequent layer. However, with the introduction of residual connections, specific layers can both receive input from and send output to layers located at non-sequential positions. This architectural modification creates a parallel pathway for information flow, allowing data to bypass intermediate layers through an alternative route within the network as shown in Figure 4.2.

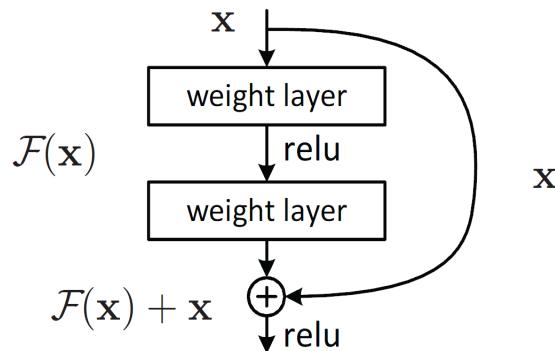


Figure 4.2: Residual Connection

In our work, the residual connection is a 2D convolutional layer (C-Skip) that directly connects the input patch with the output of the second convolutional layer (C2) and we perform addition operation between the two matrices. The output feature map number is set to 16 to match the dimensions of output matrix of layer C2 and kernel size is set to 1×1 . The C-Skip is first passed through batch normalization layer and spatial dropout layer before it connects with the subsequent layer.

Chapter 5: Hyperspectral Unmixing Datasets

5.1 Introduction

In this chapter, we will explore openly accessible hyperspectral datasets. The availability of high-quality datasets is crucial for advancing research and innovation in the field. We will introduce and discuss the unique characteristics of these datasets along with their endmembers and ground truth (GT) abundances.

The hyperspectral dataset used in our research is an open-source dataset provided in [55] and it is classified into two types: HS datasets for classification and HS datasets for unmixing. Our focus will be on unmixing datasets only.

5.2 Hyperspectral Unmixing Datasets

There are four accessible hyperspectral unmixing datasets from the source [55] namely: (1) Samson, (2) Jasper-Ridge, (3) Urban, and (4) Cuprite. In the next few sub-sections, we will go through each of the datasets in detail.

5.2.1 Samson Dataset

The Samson dataset is a relatively simple dataset. The image comprises of 952×952 pixels, with each pixel capturing data across 156 channels, spanning wavelengths from 401 nm to 889 nm. The spectral resolution is finely detailed, reaching up to 3.13 nm. Due to the computational cost associated with the original image's size, a region of 95×95 pixels is selected starting from the 252nd to 332nd pixel in the original image. The dataset is summarized in Table 2 where the column “Remaining bands” corresponds to bands remaining after preprocessing step (applies to all datasets). Notably, this subset maintains the integrity of the data, avoiding degradation from blank or noisy channels. The image showcases three distinct endmembers: *Soil*, *Tree*, and *Water*. Figure 5.1 shows Samson dataset along with its GT abundances and GT endmembers.

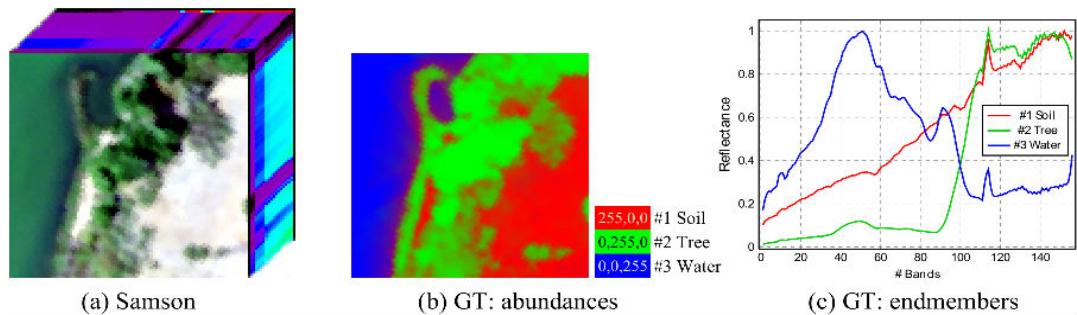


Figure 5.1: Samson dataset. (a) Image cube, (b) Ground truth abundances, (c) Ground truth spectral signatures of endmembers

Table 5.1 Summary of samson dataset. (a) shows characteristics of the dataset. (b) shows list of endmembers

Samson Dataset	
Spatial Dimension	95×95
Number of bands	156
Number of endmembers	3
Remaining bands	162
Bands Range (nm)	401 – 889

(a)

List of endmembers	
Endmember #	Endmember Name
1	Asphalt
2	Grass
3	Tree

(b)

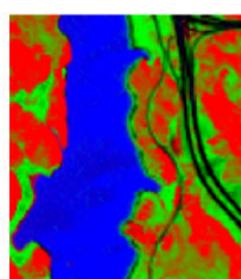
5.2.2 Jasper-Ridge Dataset

The dataset under consideration consists of 512×614 pixels, with each pixel capturing data across 224 channels covering the wavelength ranging from 380 nm to 2500 nm. The spectral resolution for each channel is finely detailed, reaching up to 9.46 nm. Due to the complexity of this hyperspectral image, obtaining GT for the entire image proves challenging. As a practical approach, we focus on a sub image measuring 100×100 pixels, starting from the 105th to 269th pixel in the original image. Table 3 shows a brief summarization of Jasper-Ridge dataset.

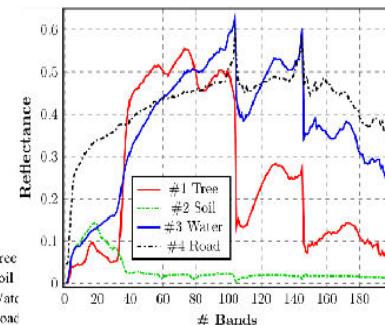
To simplify the data and address atmospheric effects, a preprocessing step is performed on the dataset. Channels 1-3, 108-112, 154-166, and 220-224 are removed due to dense water vapor and atmospheric influences. Post this preprocessing, 198 channels are retained, a common practice in hyperspectral unmixing analyses. The latent features within this dataset comprise four endmembers: *Road*, *Soil*, *Water*, and *Tree*. Figure 5.2 shows Jasper-Ridge dataset and its GT abundances and GT endmembers.



(a) Jasper-Ridge



(b) GT: abundances



(c) GT: endmembers

Figure 5.2: Jasper-Ridge dataset. (a) Image cube, (b) Ground truth abundances, (c) Ground truth spectral signatures of endmembers

Table 5.2 Summary of jasper-ridge dataset. (a) shows characteristics of the dataset. (b) shows list of endmembers

Jaspe-Ridge Dataset		List of endmembers	
Spatial Dimension		Endmember #	Endmember Name
Spatial Dimension	100 × 100	1	Asphalt
Number of bands	224	2	Grass
Number of endmembers	4	3	Tree
Remaining bands	198	4	Roof
Bands Range (nm)	380 – 2500		

(a)

(b)

5.2.3 Urban Dataset

The Urban dataset holds a prominent position in hyperspectral unmixing studies due to its widespread use. Featuring an image grid of 307×307 pixels, where each pixel corresponds to a $2 \times 2 \text{ m}^2$ area, this dataset provides a comprehensive view of urban environments. The image encompasses 210 wavelengths ranging from 400 nm to 2500 nm, resulting in a spectral resolution of 10 nm. Summary for Urban dataset can be found in Table 4.

In the preprocessing step of this dataset, channels 1-4, 76, 87, 101-111, 136-153, and 198-210 are selectively removed to address issues related to dense water vapor and atmospheric effects. This preprocessing step results in 162 channels, aligning with common practices in hyperspectral unmixing studies.

One distinctive feature of the Urban dataset is the availability of three versions of ground truth. These versions incorporate 4, 5, and 6 endmembers, providing varied perspectives that enhance the dataset's utility for detailed and nuanced analyses in hyperspectral unmixing. The four-endmember version is commonly used consisting of *Asphalt*, *Grass*, *Tree*, and *Roof*. Figure 5.3 shows Urban dataset and its GT abundances and GT endmembers.

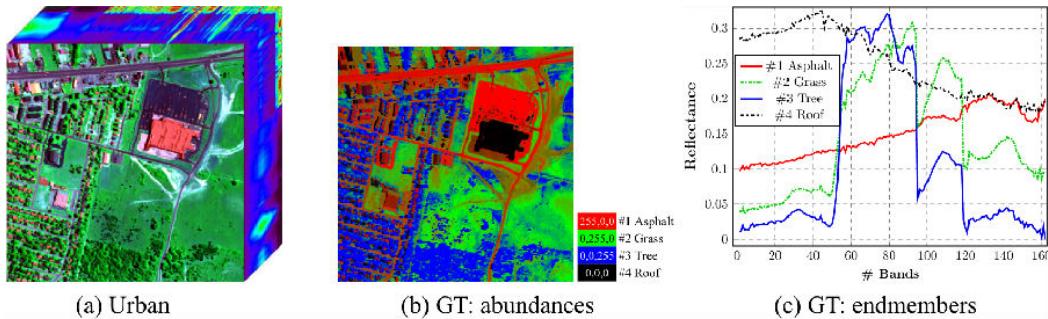


Figure 5.3: Urban dataset. (a) Image cube, (b) Ground truth abundances, (c) Ground truth spectral signatures of endmembers

Table 5.3 Summary of Urban dataset. (a) shows characteristics of the dataset. (b) shows list of endmembers

Urban Dataset	
Spatial Dimension	307×307
Number of bands	210
Number of endmembers	4
Remaining bands	162
Bands Range (nm)	400 – 2500

(a)

List of endmembers	
Endmember #	Endmember Name
1	Asphalt
2	Grass
3	Tree
4	Roof

(b)

5.2.4 Cuprite Dataset

The Cuprite dataset as summarized in Table 5, is the most complex dataset out of all and stands as a benchmark in hyperspectral unmixing research, capturing the Cuprite region in Las Vegas, NV, U.S. This dataset comprises 224 channels across the spectral range from 370 nm to 2480 nm. To enhance data quality, channels 1-2 and 221-224 (noisy channels) and water absorption channels 104–113 and 148–167 are removed and the dataset is refined resulting in 188 channels.

A specific region measuring 250×190 pixels is selected for analysis, featuring a diverse array of 14 mineral types. Recognizing minor differences between variants of similar minerals, the number of endmembers is strategically reduced to 12 (see Table 4-b for list of endmembers). Figure 5.4 depicts pure spectral signatures of endmembers present in Cuprite dataset.

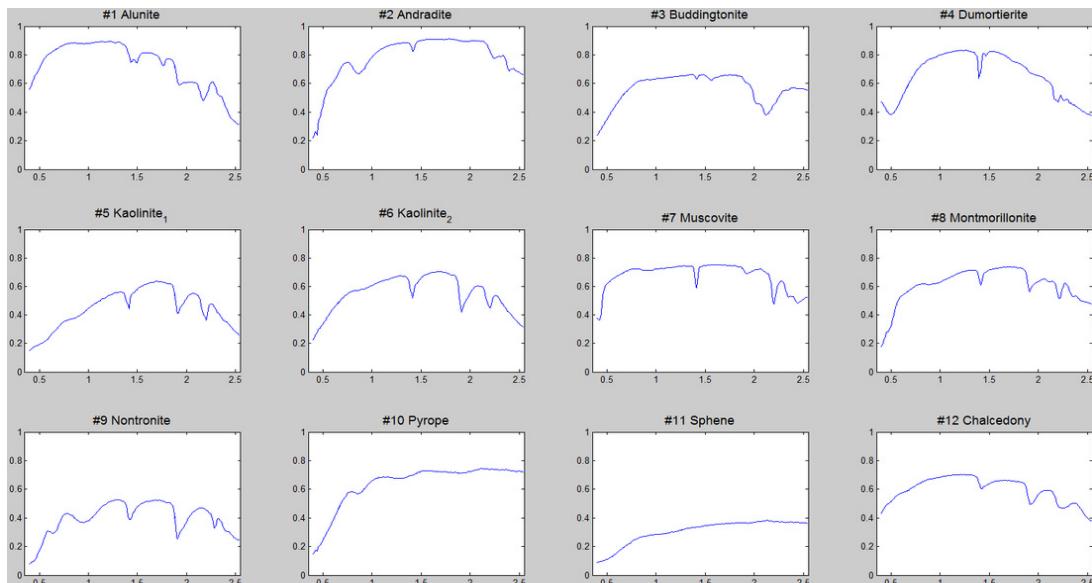


Figure 5.4: Spectral signatures of 12-endmembers in cuprite dataset

Table 5.4 Summary of cuprite dataset. (a) shows characteristics of the dataset. (b) shows list of endmembers

<i>Cuprite Dataset</i>		<i>List of endmembers</i>	
Spatial Dimension	250×190	Endmember #	Endmember Name
Number of bands	224	1	Alunite
Number of endmembers	12	2	Andradite
Remaining bands (nm)	188	3	Buddingtonite
Bands Range (nm)	$370 - 2480$	4	Dumortierite
(a)		5	Kaolinite1
(b)		6	Kaolinite2
(b)		7	Muscovite
(b)		8	Montmorillonite
(b)		9	Nontronite
(b)		10	Pyrope
(b)		11	Sphene
(b)		12	Chalcedony

The Cuprite dataset is provided in two variants, featuring images with 224 bands and 188 bands, respectively. Researchers can choose between these configurations based on their specific hyperspectral unmixing study requirements.

5.3 Conclusion

In this chapter, we have discussed several hyperspectral unmixing datasets and their characteristics including number of bands before and after preprocessing step, spatial dimensions, number of endmembers, their spectral signatures, and ground truth abundances.

It is to be noted that all the data preparation and preprocessing discussed has been already performed by the owners of source [55]. Lastly, the cuprite dataset featuring 12 distinct endmembers is a highly complex dataset, therefore, it will be excluded from our experimental phase and remaining 3 datasets will be used. In the next chapter, we will run experiments on our proposed RCAE model and comparisons will be made with some of the recent deep learning-based models.

Chapter 6: Experimental Results

6.1 Introduction

In this chapter, we will utilize our proposed RCAE model to conduct experiments on real HS unmixing datasets, as outlined in Chapter 5. We aim to see how well our model works in extracting useful information from HS images. The goal is to understand how effective the model is in practical situations and to learn more about its strengths and limitations. Through these experiments, we aim to contribute to the improvement of HS image unmixing techniques.

6.2 Experimental results

For any unsupervised unmixing model, two key outcomes are expected: the spectral signature curve of endmembers and their corresponding abundance maps. These outcomes represent essential components in analyzing HS images, providing insights about the image. The datasets used in our experiments are the following: Urban, Samson, and Jasper Ridge. We will utilize two different variants of the Urban dataset, each varying in the number of endmembers present in it (Urban 4 and Urban 5).

6.2.1 Experiment with Urban (four) dataset

Urban is a highly utilized hyperspectral dataset, featuring four distinct endmembers: Asphalt, Grass, Tree, and Roof. Following preprocessing, the dataset comprises 162 channels as discussed in detail in Chapter 5. Figures 6.1 and 6.2 display the spectral signatures of the endmembers and their corresponding abundance maps, respectively.

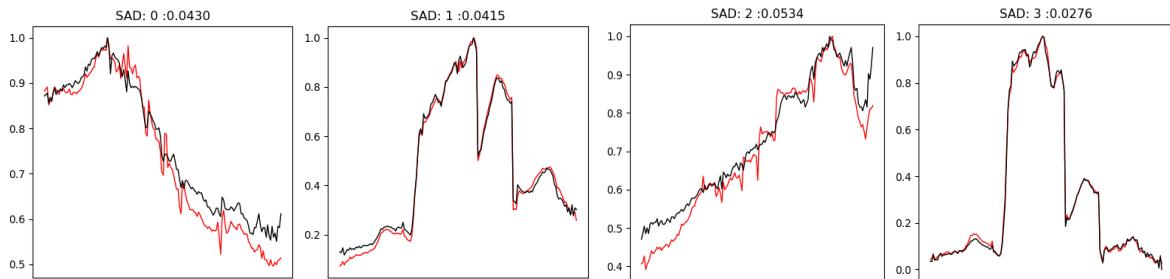


Figure 6.1: Urban dataset endmember signatures. The black lines are the ground truth endmembers and lines in red are endmembers obtained from RCAE

Both figures demonstrate the remarkable similarity between the endmembers estimated by the proposed RCAE model and the ground truth. Furthermore, the abundance maps offer enhanced clarity in identifying various materials present in the scene. The Urban dataset due to its grid size is a computationally expensive dataset to be used in unmixing, as a result, most of the research papers do not include this dataset in the experiment phase.

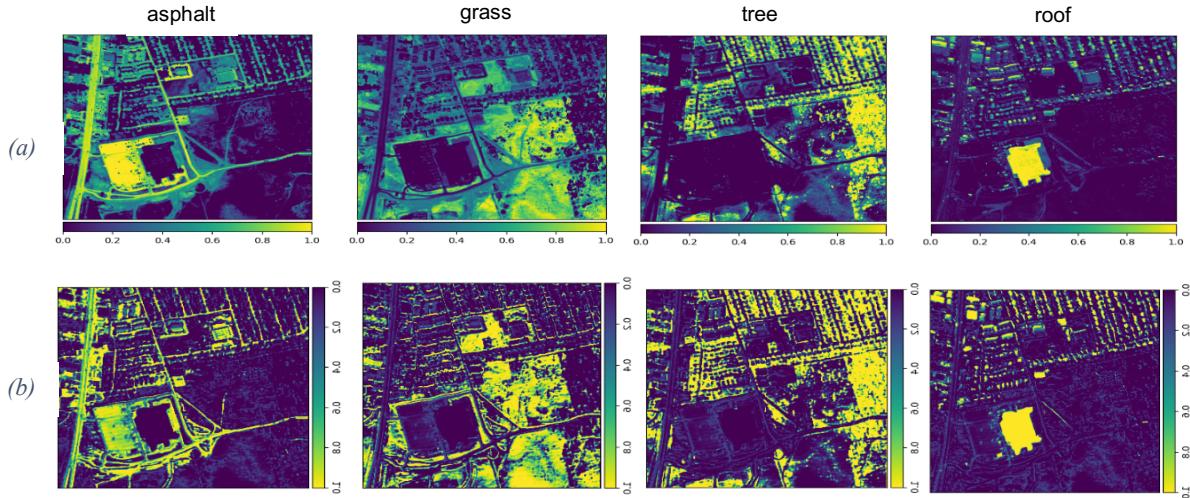


Figure 6.2: Urban-four dataset abundance maps. (a) Ground truth abundance maps (b) Extracted abundance maps

6.2.2 Experiment with Samson dataset

Samson is a well-known hyperspectral dataset containing three endmembers: Soil, Tree, and Water, and 156 bands after the removal of noisy bands. Figure 6.3 shows the GT and extracted spectral signatures of the Samson dataset while Figure 6.4 shows the abundances.

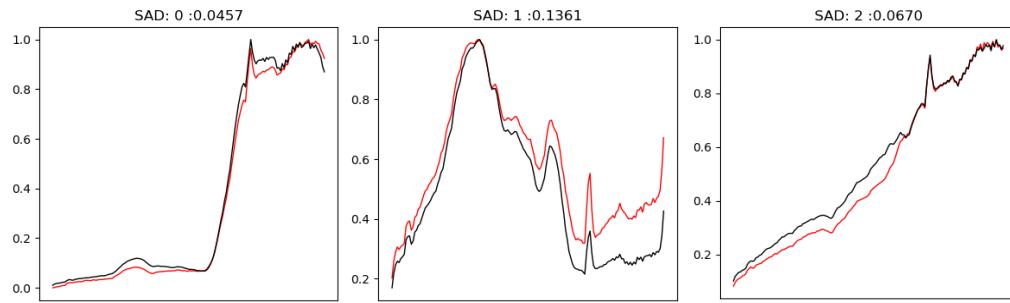


Figure 6.3: Samson dataset endmember signatures. The black lines are the ground truth endmembers and lines in red are endmembers obtained from RCAE

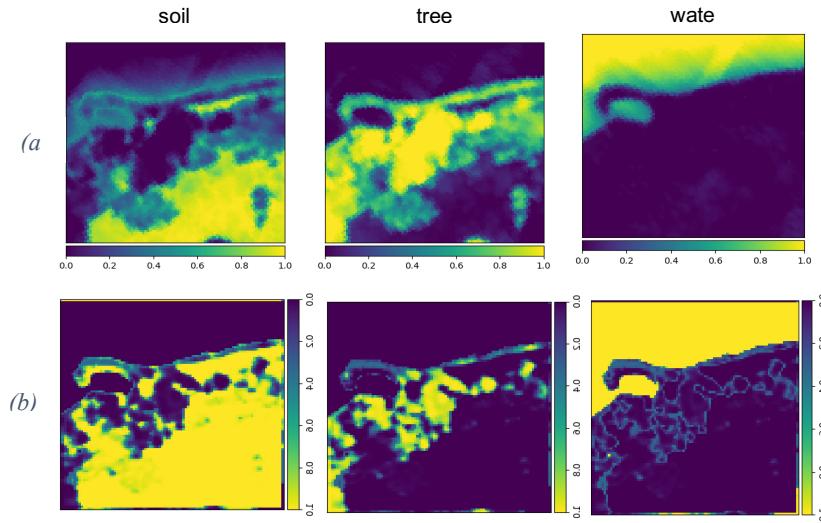


Figure 6.4: Samson dataset abundance maps. (a) Ground truth abundance maps (b) Extracted abundance

6.2.3 Experiment with Jasper Ridge dataset

Jasper Ridge is an often-tested dataset in remote sensing literature. The image comprises 198 bands and offers 4 distinct endmembers: Tree, Water, Dirt, and Road. The experimental results for the Jasper-Ridge dataset are given in Figure 6.5 and Figure 6.6. As can be seen in Figures 6.5 and 6.6, the model suffers from correctly identifying the “Road” endmember. This is because the Jasper Ridge dataset contains a very small proportion of “Road” material and hence leads to poor training. The model extracts decent results for the remaining endmembers.

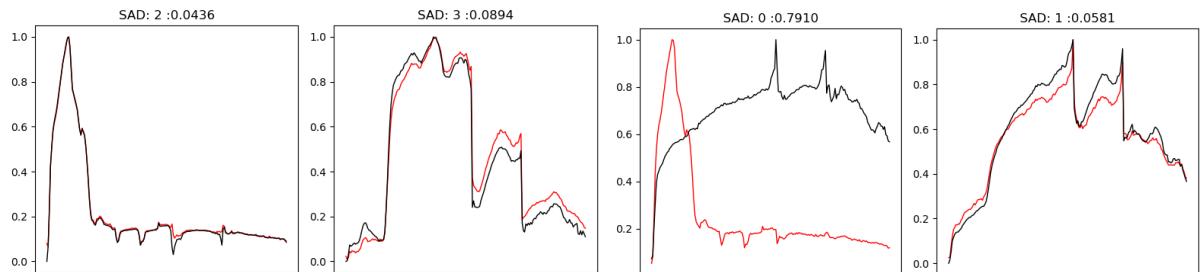


Figure 6.5: Jasper Ridge dataset endmember signatures. The black lines are the ground truth endmembers and lines in red are endmembers obtained from RCAE

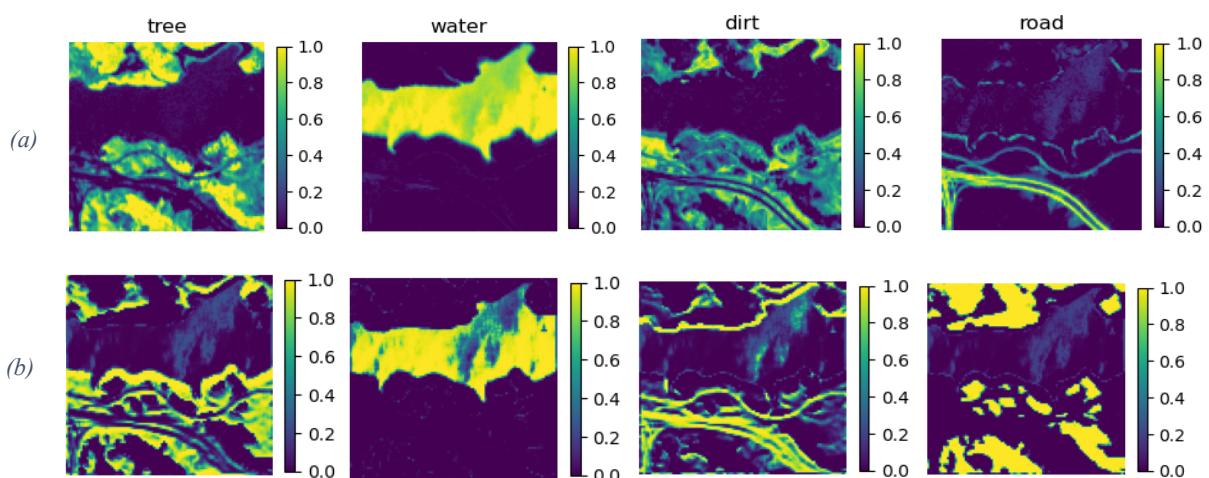


Figure 6.6: Jasper Ridge dataset abundance maps. (a) Ground truth abundance maps (b) Extracted abundance maps

6.2.4 Experiment with Urban (five) dataset

Urban-five is another variant of the Urban dataset offering one additional endmember “Dirt”. The remaining endmembers are same as in the Urban four dataset (Asphalt, Grass, Tree, and Roof). As discussed earlier, this dataset is computationally expensive dataset and often omitted from experiments. Figures 6.7 and 6.8 show the endmembers and abundances respectively. It can be seen that the RCAE model successfully identifies all the endmembers and gives decent abundance maps.

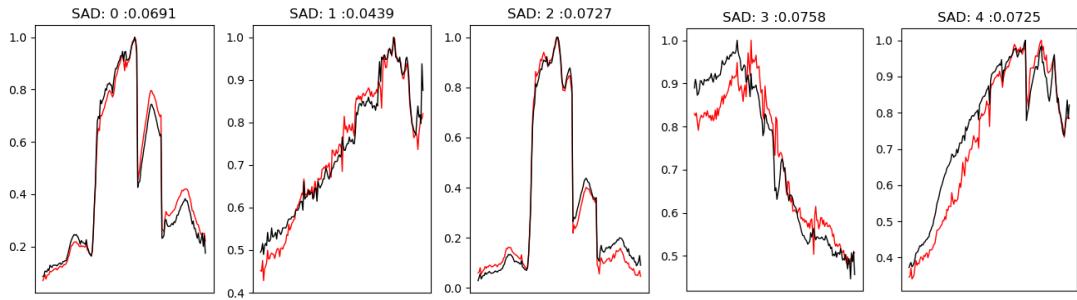


Figure 6.7: Urban-five dataset endmember signatures. The black lines are the ground truth endmembers and lines in red are endmembers obtained from RCAE

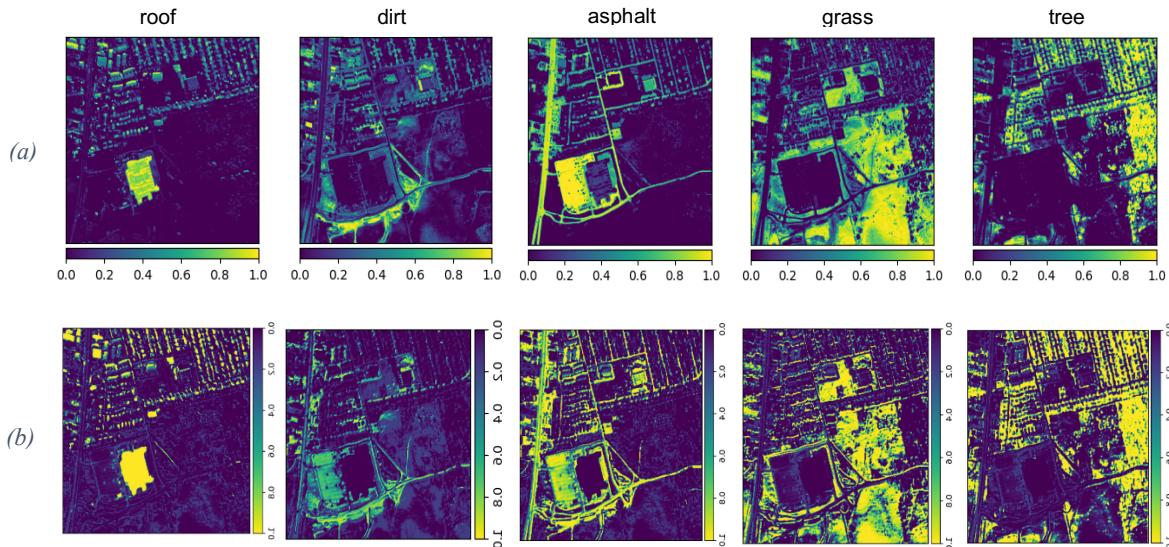


Figure 6.8 Urban-five dataset abundance maps. (a) Ground truth abundance maps (b) Extracted abundance maps

6.3 Conclusion

In this chapter, we conducted experiments using real hyperspectral datasets namely Urban with four endmembers, Samson, Jasper-Ridge, and Urban with five endmembers. It can be seen that our proposed RCAE model can accurately identify endmembers and yields average abundance maps. The quality of extracted endmembers weighs more heavily in our evaluation of the method than the quality of the abundance maps. The abundance maps extracted by RCAE model are very intense looking (bright), the reason is that abundance maps produced by our method tend to be very binary, that is abundances are either very low or very high, almost like classification maps. This may have happened because of the use of the softmax activation function to enforce the sum-to-one constraint and batch normalization layers. Finally, Figures 6.5 and 6.6 show another limitation of our model, that is the model cannot accurately identify the material if it is present in less proportion in the HS image which is demonstrated by the “road” endmember.

Chapter 7: Validation and Comparative Analysis

7.1 Introduction

Following the completion of our experiments, the next step involves utilizing validation metrics to quantitatively measure the difference between the ground truth data and the endmembers and abundance maps extracted by our RCAE model. Furthermore, we aim to broaden our understanding by conducting a comprehensive comparative analysis. In this regard, we will not only evaluate the performance of our RCAE model but also compare it against other state-of-the-art unmixing methods. This comparative study will allow us to gain deeper insights into the strengths and weaknesses of different methodologies, thereby guiding future research directions in the field of spectral unmixing.

7.2 Loss Functions

In this paper, we tested three loss functions: Spectral Angle Distance (SAD), and two variations of Root Mean Square Error (RMSE). The SAD function is used to quantify the endmembers while the RMSE metrics will quantify abundance maps. The mathematical formulation of loss functions is given below:

The SAD is given by the following equation:

$$mSAD = \frac{1}{P} \sum_{i=1}^P \cos^{-1} \left(\frac{\hat{x}_i^T x_i}{\|\hat{x}_i\| \|x_i\|} \right)$$

where x and \hat{x} are ground-truth and extracted endmembers respectively and P represents the number of endmembers. The lower the mSAD, the higher the similarity.

The RMSE_1 is calculated by the formula:

$$mRMSE_1 = \frac{1}{P} \sum_{i=1}^P \sqrt{(s_i - \hat{s}_i)^2}$$

where s is the ground-truth abundance and \hat{s} is the extracted abundance.

The RMSE_2 (class-wise) is computed as:

$$RMSE_2 = \sum_{i=1}^B \sqrt{\frac{(s_i - \hat{s}_i)^2}{w \times h}}$$

This RMSE_2 metric gives the scores for each endmember in the HS data.

The mean RMSE_2 is calculated as:

$$mRMSE_2 = \sqrt{\sum \frac{(s - \hat{s})^2}{w \times h \times B}}$$

where w , h and B represents the width, height, and number of bands respectively.

7.3 Validation Results

In this section, we will use the extracted results of all datasets by the RCAE model and apply the loss functions to each of them. As aforementioned, the SAD metrics will be used to quantify the endmember signatures and RMSE variations will quantify abundance maps. Lower values for metrics imply a higher similarity between true and achieved results.

7.3.1 Validating endmembers with SAD loss function

The datasets that we used to conduct experiments were Urban (four), Samson, Jasper, and Urban (five). Table 7.1 shows SAD values for the Urban (four) dataset and the last column shows the average SAD score for all endmembers.

	roof	grass	asphalt	tree	mSAD
SAD	0.0430	0.0415	0.0534	0.0276	0.041

Table 7.1 SAD and average SAD scores for the Urban (four) dataset

Table 7.2 shows SAD scores for the Samson dataset.

	Tree	Water	Soil	mSAD
SAD	0.0457	0.1361	0.0670	0.0830

Table 7.2 SAD and average SAD scores for the Samson dataset

Table 7.3 shows SAD scores for the Jasper-Ridge dataset.

	dirt	tree	road	water	mSAD
SAD	0.0436	0.0894	0.7910	0.0581	0.2450

Table 7.3 SAD and average SAD scores for the Jasper-Ridge dataset

Table 7.4 shows SAD scores for the Urban (five) dataset.

	grass	asphalt	tree	roof	dirt	mSAD
SAD	0.0691	0.0439	0.0727	0.0758	0.0725	0.0670

Table 7.4 SAD and average SAD scores for the Urban (five) dataset

7.3.2 Validating abundance maps with RMSE loss functions

Now, we use two variations of the RMSE function: RMSE_1 and RMSE_2 to test our abundance results. As already discussed in the previous chapter, the abundance maps obtained by the RCAE model are very intense-looking due to their binary nature. Therefore, the RMSE values of abundance maps tend to be high which indicates that the model gives not the best but decent maps. Table 7.5 shows the average RMSE_1 and RMSE_2 (class-wise and average) for the Urban (four) dataset.

	roof	grass	asphalt	tree	mRMSE_2
RMSE_2	0.4702	0.4932	0.4681	0.4399	0.4682
mRMSE_1				143.7560	

Table 7.5 RMSE scores for the Urban (four) dataset

Table 7.6 shows mean RMSE_1 and RMSE_2 scores of the Samson dataset.

	tree	water	soil	mRMSE_2
RMSE_2	0.5310	0.5467	0.5218	0.5333
mRMSE_1		43.8766		

Table 7.6 RMSE scores for the Samson dataset

Table 7.7 shows RMSE_1 and RMSE_2 scores of the Jasper-Ridge dataset.

	dirt	tree	road	water	mRMSE_2
RMSE_2	0.5306	0.4594	0.5019	0.4938	0.4971
mRMSE_1			49.7135		

Table 7.7 RMSE scores for the Jasper-Ridge dataset

Table 7.8 shows mean RMSE_1 and RMSE_2 scores of the Urban (five) dataset.

	grass	asphalt	tree	roof	dirt	mRMSE_2
RMSE_2	0.4651	0.5087	0.4706	0.3901	0.4432	0.4555
mRMSE_1				152.1445		

Table 7.8 RMSE scores for the Urban (five) dataset

7.4 Comparative Analysis

In this section, we compare the results of the RCAE model with other HS unmixing models to understand how well our model performs among its competitors. The models we selected for comparison are VCA, N-FINDR, MVSA, CAE, and Autoencoder (Palsson et al. 2018). The selected models conducted experiments on the subset of the same datasets that we used in our research. The comparison methods are listed in Table 7.9 along with their descriptions in the second column. The Urban (five) dataset is not included in this comparison section as none of the methods (from Table 7.9) utilized this dataset during the experimental phase. Moreover, the CAE method did not conduct experiments on the Urban dataset, therefore, it is left empty in Table 7.10.

METHOD	DESCRIPTION
VCA	Vertex Component Analysis [8]. A geometrical-based approach.
N-FINDR	N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data [25]. A geometrical-based approach.
MVSA	Minimum Volume Simplex Analysis [9]. A geometrical-based approach.
CAE	Convolutional Autoencoder: A deep learning-based Approach [19].
AUTOENCODER (PALSSON ET AL. 2018)	Palsson Burkni, et al. “Hyperspectral unmixing using a neural network Autoencoder.” [17]

Table 7.9 Benchmark methods used for comparison in experiments

7.4.1 Comparison with Urban Dataset

The obtained results by the aforementioned methods in terms of SAD and RMSE are shown in Table 7.10 for the Urban (four) dataset. It can be observed that the proposed RCAE model obtained the highest value for the SAD metric. The second-best result for SAD and the highest value of RMSE are obtained by the Autoencoder (Palsson et al) method which proves the superiority of deep-learning methods over the traditional methods. As discussed earlier, our method focuses more on extracting high-quality endmembers rather than

focusing on improved abundance maps. Therefore, the RMSE value achieved by the RCAE method is relatively high compared to other methods. VCA obtained the worst results for SAD and the second worst for RMSE.

	VCA	N-FINDR	MVSA	CAE	Autoencoder (Palsson et al)	RCAE (Proposed)
SAD	0.5552	0.3610	0.4161	N/A	<u>0.1590</u>	0.041
RMSE	0.4151	<u>0.2154</u>	0.3457	N/A	0.0360	0.4682

Table 7.10 Average SAD and RMSE obtained by different methods for the Urban dataset, where the best results are in bold and the second-best results are underlined

7.4.2 Comparison with Samson Dataset

The obtained results by the aforementioned methods in terms of SAD and RMSE are shown in Table 7.11 for the Samson dataset. It can be seen that the Autoencoder (Palsson et al) method of unmixing outperforms all the unmixing methods in terms of both SAD and RMSE. The RCAE model got the second-best result for the SAD metric but obtained worst result for the RMSE metric. CAE model obtained poor results for both RMSE and SAD, this may have happened due to the simplicity of the model and complexity of the dataset.

	VCA	N-FINDR	MVSA	CAE	Autoencoder (Palsson et al)	RCAE (Proposed)
SAD	0.1947	0.0914	0.1045	0.3196	0.0361	<u>0.0830</u>
RMSE	0.0992	0.0857	<u>0.0812</u>	0.2479	0.0166	0.5333

Table 7.11 Average SAD and RMSE obtained by different methods for the Samson dataset, where the best results are in bold and the second-best results are underlined.

7.4.3 Comparison with Jasper-Ridge Dataset

The obtained results by the aforementioned methods in terms of SAD and RMSE are shown in Table 7.12 for the Japser-Ridge dataset. The Autoencoder (Palsson et al) method again obtained the best results for both SAD and RMSE metrics followed by the second-best for the SAD metric obtained by our proposed model. In Table 7.3, it can be seen the model effectively identifies all the endmembers except the “road” endmember, which is responsible for degrading the overall SAD score of the Japser-Ridge dataset.

	VCA	N-FINDR	MVSA	CAE	Autoencoder (Palsson et al)	RCAE (Proposed)
SAD	0.364	0.5540	0.558	0.4351	0.0752	0.2450
RMSE	0.433	0.2540	0.287	<u>0.1681</u>	0.0199	0.4971

Table 7.12 Average SAD and RMSE obtained by different methods for the Japser-Ridge dataset, where the best results are in bold and the second-best results are underlined.

NFINDR and MVSA both gave almost identical results, CAE obtained the second-best result in terms of RMSE.

7.5 Conclusion

In this chapter, the performance of the proposed model was evaluated using SAD and RMSE performance metrics. In addition to this evaluation, the model was also compared with other unmixing benchmark methods from both geometrical and deep learning-based classes. Results clearly show the superiority of deep learning-based methods over geometrical-based methods. The method Autoencoder (Palsson et al) obtained the best results for almost all datasets followed by our proposed RCAE model. RCAE model can successfully identify endmembers from an HS image but suffers from correctly identifying the abundance fractions. This limitation as already discussed comes from the binary nature of abundance maps. Overall, the proposed method achieves the best or second-best results for the SAD metric and can be used in cases where only identification of endmembers is crucial and estimating abundances is not required.

Chapter 8: Summary and Conclusion

This chapter serves as a summary and conclusion of our research on the hyperspectral (HS) unmixing problem. This research project began with an introduction to HS imaging and deep learning, exploring their applications in computer vision. We provided a comprehensive overview of the background of HS unmixing, discussing two primary models: Linear Mixing Model (LMM) and Non-Linear Mixing Model (NLMM). We also explained the three categories of HS unmixing algorithms: geometrical, statistical, and sparse regression-based, alongside deep learning-based algorithms. State-of-the-art algorithms within each category were scrutinized to understand their methodologies.

Furthermore, we explained the problem statement of HS unmixing, emphasizing the importance of enforcing physical constraints. Subsequently, we provided detailed insights into HS unmixing datasets such as Urban, Samson, and Jasper-Ridge. Characteristics like the number of channels, dimensions, remaining channels after preprocessing, and the number of endmembers were discussed for each dataset.

Our proposed solution revolves around a deep-learning-based unmixing model known as Residual Convolutional Autoencoder (RCAE). This architecture incorporates skip connections, enhancing the model's ability to capture complex spectral information. The encoder block of the RCAE model comprises four 2D convolutional layers with a kernel size of 3×3 for the first two layers and a kernel size of 1×1 for the remaining layers, including the skip connection layer. All layers of the encoder module utilize the Leaky ReLU activation function, except the last layer, which uses the softmax activation function to enforce the abundance sum-to-one (ASC) and abundance non-negativity constraint (ANC). The decoder module uses a single 2D convolutional layer to reconstruct the image. The RCAE model receives HS image as input in patches. It then encodes the input patch into a lower dimensional image which is then passed to decoder. The decoder block takes this compressed image patch and reconstructs it back into its original dimension. The skip connection short-circuits the information flow from the initial input patch and adds it with the output of the second convolutional layer. The compressed representation of HS image (output of encoder module) serves as abundance maps, and endmember signatures can be extracted from decoder's weight matrix.

We conducted experiments using real HS datasets and employed Root Mean Squared Error (RMSE) and Spectral Angle Distance (SAD) loss functions for quantitative validation. However, our experiments revealed two limitations of the proposed RCAE model. Firstly, the model struggles to accurately identify endmembers when their fractional presence is insufficient. Secondly, the abundance maps produced by RCAE appear overly intense, which may not be practical for scenarios requiring precise abundance mapping.

Finally, we performed a comparative analysis of our RCAE model with existing unmixing models, highlighting its strengths and weaknesses. Overall, while the RCAE model shows promise in HS unmixing, further refinement is necessary to address its limitations and enhance its practical applicability.

Chapter 9: Lessons Learnt and Future Work

Our research on hyperspectral (HS) unmixing using the Residual Convolutional Autoencoder (RCAE) has provided several valuable insights. Firstly, the importance of balancing model complexity and computational efficiency became evident. The RCAE's skip connections improved its capacity to capture complex spectral information, but the performance suggested that simpler architectures might still be effective, or that additional complexity could be advantageous. Secondly, we encountered challenges in accurately identifying endmembers, particularly when their fractional presence was minimal. This underscores the need for more sophisticated mechanisms within models to detect subtle spectral signatures effectively. Additionally, the RCAE's tendency to produce overly intense abundance maps revealed that while the model is highly sensitive to spectral features, it might benefit from additional constraints or regularization techniques to enhance its practical applicability.

Moving forward, several areas warrant further investigation and development. To address the challenge of endmember detection, future models could incorporate advanced techniques such as attention mechanisms or multi-scale feature extraction to improve sensitivity to minor spectral components. Additionally, refining the RCAE architecture to include better regularization methods or hybrid approaches that combine deep learning with traditional unmixing techniques might mitigate the issue of overly intense abundance maps. Exploring alternative loss functions that better capture the nuances of HS data could also enhance model performance. Finally, expanding the model's validation to include a broader range of HS datasets and real-world applications will be crucial in assessing its generalizability and robustness. These efforts will contribute to the continued advancement of HS unmixing methodologies, pushing towards more accurate and practical solutions in the field.

References

- [1] Bioucas-Dias, José M., et al. "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches." *IEEE journal of selected topics in applied earth observations and remote sensing* 5.2 (2012): 354-379.
- [2] <https://speclib.jpl.nasa.gov/>
- [3] Maria Petrou and Panagiota Bosdogianni. 1999. *Image Processing: The Fundamentals* (1st. ed.). John Wiley & Sons, Inc., USA.
- [4] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.
- [5] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4484–4502, Nov. 2012.
- [6] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.
- [7] M. D. Iordache, J. M. Bioucas-Dias, A. Plaza, and B. Somers, "MUSICCSR: Hyperspectral unmixing via multiple signal classification and collaborative sparse regression," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 4364–4382, Jul. 2014.
- [8] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.
- [9] Li, J., & Bioucas-Dias, J. M. (2008, July). Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data. In *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium* (Vol. 3, pp. III-250). IEEE
- [10] Bioucas-Dias, J. M. (2009, August). A variable splitting augmented Lagrangian approach to linear spectral unmixing. In *2009 First workshop on hyperspectral image and signal processing: Evolution in remote sensing* (pp. 1-4). IEEE.
- [11] J. M. P. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of Dirichlet components," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 863–878, Mar. 2012.
- [12] N. Dobigeon, S. Moussaoui, M. Coulon, J. Y. Tourneret, and A. O.Hero, "Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4355–4368, Nov. 2009.
- [13] O. Eches, N. Dobigeon, C. Mailhes, and J. Y. Tourneret, "Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1403–1413, Jun. 2010.
- [14] Chollet, Francois. *Deep Learning with Python*. Manning Publications, 2017.

- [15] L. Zhang et al., "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, June 2016.
- [16] J. S. Bhatt and M. V. Joshi, "Deep Learning in Hyperspectral Unmixing: A Review," *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, Waikoloa, HI, USA, 2020, pp. 2189-2192, doi: 10.1109/IGARSS39084.2020.9324546.
- [17] Palsson, Burkni, et al. "Hyperspectral unmixing using a neural network autoencoder." *IEEE Access* 6 (2018): 25646-25656.
- [18] Khan, S., Rahmani, H., Shah, S. A. A., Bennamoun, M., Medioni, G., & Dickinson, S. (2018). *A guide to convolutional neural networks for computer vision* (Vol. 8, No. 1, pp. 1-207). San Rafael: Morgan & Claypool Publishers.
- [19] J. Masci, U. Meier, D. Cire, san, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59
- [20] Ranasinghe, Y., Herath, S., Weerasooriya, K., Ekanayake, M., Godaliyadda, R., Ekanayake, P., & Herath, V. (2020, November). Convolutional autoencoder for blind hyperspectral image unmixing. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)* (pp. 174-179). IEEE.
- [21] Palsson, B., Ulfarsson, M. O., & Sveinsson, J. R. (2020). Convolutional autoencoder for spectral–spatial hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1), 535-549.
- [22] Khajehrayeni, F., & Ghassemian, H. (2020). Hyperspectral unmixing using deep convolutional autoencoders in a supervised scenario. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 567-576.
- [23] Elkholy, M. M., Mostafa, M., Ebied, H. M., & Tolba, M. F. (2020). Hyperspectral unmixing using deep convolutional autoencoder. *International Journal of Remote Sensing*, 41(12), 4799-4819.
- [24] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, 2002
- [25] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Proc. SPIE Image Spectrometry V*, 1999, vol. 3753, pp. 266–277.
- [26] R. A. Neville, K. Staenz, T. Szeregi, J. Lefebvre, and P. Hauff, "Automatic endmember extraction from hyperspectral data for mineral exploration," in *Proc. Canadian Symp. Remote Sens.*, 1999, pp. 21–24.
- [27] J. Boardman, "Automating spectral unmixing of AVIRIS data using convex geometry concepts," in *Proc. Ann. JPL Airborne Geosci. Workshop*, 1993, vol. 1, pp. 11–14.
- [28] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Proc. JPL Airborne Earth Sci. Workshop*, 1995, pp. 23–26.
- [29] R. H. Yuhas, A. F. H. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the Spectral Angle Mapper (SAM) algorithm,"

- in Summaries 3rd Annu. JPL Airborne Geoscience Workshop, vol. 1, R. O. Green, Ed., 1992, JPL Publ. 92-14, pp. 147–149.
- [30] J. J. Settle, “On the relationship between spectral unmixing and subspace projection,” IEEE Trans. Geosci. Remote Sens., vol. 34, no. 4, pp. 1045–1046, Jul. 1996.
 - [31] Dennison, P. E., & Roberts, D. A. (2003). Endmember selection for multiple endmember spectral mixture analysis using endmember average RMSE. *Remote sensing of environment*, 87(2-3), 123-135.
 - [32] R. N. Clark, G. A. Swayze, A. Gallagher, T. V. King, and W. M. Calvin, “The U.S. Geological Survey digital spectral library: Version 1: 0.2 to 3.0 m,” U.S. Geol. Surv., Denver, CO, Open File Rep. 93-592, 1993.
 - [33] Lines, L. R., & Treitel, S. (1984). A review of least-squares inversion and its application to geophysical problems. *Geophysical prospecting*, 32(2), 159-186.
 - [34] Lawson, C. L., & Hanson, R. J. (1995). *Solving least squares problems*. Society for Industrial and Applied Mathematics.
 - [35] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, “A convex analysis based minimum-volume enclosing simplex algorithm for hyperspectral unmixing,” IEEE Trans. Signal Process., vol. 57, no. 11, pp. 4418–4432, Nov. 2009.
 - [36] L. Miao and H. Qi, “Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization,” IEEE Trans. Geosci. Remote Sens., vol. 45, no. 3, pp. 765–777, Mar. 2007.
 - [37] Bioucas-Dias, J. M., & Plaza, A. (2010, October). Hyperspectral unmixing: Geometrical, statistical, and sparse regression-based approaches. In *Image and signal processing for remote sensing XVI* (Vol. 7830, pp. 79-93). SPIE.
 - [38] Stögbauer, H., Kraskov, A., Astakhov, S. A., & Grassberger, P. (2004). Least-dependent-component analysis based on mutual information. *Physical Review E*, 70(6), 066123.
 - [39] S. Moussaoui, C. Carteret, D. Brie, and A. Mohammad-Djafari, “Bayesian analysis of spectral mixture data using Markov chain Monte Carlo methods,” *Chemometr. Intell. Lab. Syst.*, vol. 81, no. 2, pp. 137–148, 2006.
 - [40] N. Dobigeon, S. Moussaoui, J.-Y. Tourneret, and C. Carteret, “Bayesian separation of spectral sources under non-negativity and full additivity constraints,” *Signal Process.*, vol. 89, no. 12, pp. 2657–2669, Dec. 2009.
 - [41] M. Arngren, M. N. Schmidt, and J. Larsen, “Unmixing of hyperspectral images using Bayesian nonnegative matrix factorization with volume prior,” *J. Signal Process. Syst.*, vol. 65, no. 3, pp. 479–496, Nov. 2011.
 - [42] D. M. Rogge, B. Rivard, J. Zhang, and J. Feng, “Iterative spectral unmixing for optimizing per-pixel endmember sets,” IEEE Trans. Geosci. Remote Sens., vol. 44, no. 12, pp. 3725–3736, 2006
 - [43] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, “On the use of spectral libraries to perform sparse unmixing of hyperspectral data,” in Proc. IEEE GRSS Workshop Hyperspectral Image Signal Process.: Evolution in Remote Sens. (WHISPERS), 2010, vol. 1, pp. 1–4.
 - [44] B. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227234, 1995.

- [45] Y. C. Pati, R. Rezahfar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," Proceedings of the 27th Annual Asilomar Conference on Signals, Systems and Computers, Los Alamitos, CA, USA, 2003.
- [46] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit" SIAM Review, vol. 43, no. 1, pp. 129159, 2001.
- [47] Lin Qi, Jie Li, Ying Wang, Mingyu Lei, Xinbo Gao, Deep spectral convolution network for hyperspectral image unmixing with spectral library, Signal Processing, Volume 176, 2020, 107672, ISSN 0165-1684.
- [48] G.A. Licciardi and F. Del Frate, "Pixel unmixing in hyperspectral data by means of neural networks," IEEE Trans. Geosci. Remote Sens., vol. 49, no. 11, pp. 4163–4172, 2011.
- [49] R. Guo, W. Wang, and H. Qi, "Hyperspectral image unmixing using autoencoder cascade," in 2015 7th Workshop on Hyperspectral Image and Signal Processing Evolution in Remote Sensing (WHISPERS), June 2015, pp. 1–4.
- [50] Kyunghyun Cho, "Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images," in Proceedings of ICML, May 2013, vol. 28, pp. 432–440.
- [51] Y. Qu and H. Qi, "udas: An untied denoising autoencoder with sparsity for spectral unmixing," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 3, pp. 1698–1712, March 2019.
- [52] S. Ozkan, B. Kaya, and G. B. Akar, "Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 1, pp. 482–496, Jan 2019.
- [53] Y. Su et al., "Daen: Deep autoencoder networks for hyperspectral unmixing," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 7, pp. 4309–4321, July 2019.
- [54] B. Palsson, M. O. Ulfarsson and J. R. Sveinsson, "Convolutional Autoencoder for Spectral–Spatial Hyperspectral Unmixing," in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 1, pp. 535–549, Jan. 2021, doi: 10.1109/TGRS.2020.2992743.
- [55] <http://lesun.weebly.com/hyperspectral-data-set.html>

Appendix

Appendix A

Convolutional Neural Networks in Computer Vision

Convolutional Neural Networks (CNNs) have notably improved computer vision, particularly excelling in image classification, where models like AlexNet and ResNet achieved breakthroughs in accuracy, outperforming traditional methods. In object detection, architectures such as Faster R-CNN and YOLO demonstrated remarkable capabilities, enabling the identification and localization of multiple objects within images, contributing to applications in autonomous vehicles and surveillance. Semantic segmentation tasks benefited from CNNs like U-Net, allowing precise pixel-level classification, with applications ranging from medical image analysis to scene understanding. Transfer learning, data augmentation, and attention mechanisms have further boosted CNN performance, allowing models to adapt to specific tasks, generalize better, and focus on relevant image regions.

The impact of CNNs extends into real-time applications, including facial recognition and augmented reality, where their rapid processing capabilities are instrumental. However, challenges like adversarial attacks underscore the need for ongoing research to enhance the robustness of CNNs. In video analysis, 3D CNNs have been instrumental in capturing spatial and temporal information, advancing action recognition. Overall, CNNs have become foundational in computer vision, powering breakthroughs across diverse tasks and applications, albeit with the recognition of ongoing challenges in their deployment.