REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical and Electronics Engineering

INCREASING SPECTRAL RESOLUTION OF HYPERSPECTRAL IMAGES WHILE DECREASING SPECTRAL VARIABILITY USING DEEP GENERATIVE MODELS

Master Thesis

Khalid BOUNAIL

Supervisor

Asst. Prof. Dr. Sevcan KAHRAMAN

Istanbul – 2023

THESIS INTRODUCTION FORM

Name and Surname	**:**	Khalid BOUNAIL
Language of the Thesis	**:**	English
Name of the Thesis	**:**	Increasing spectral resolution of hyperspectral images while decreasing spectral variability using Deep. Generative Modeling
Institute	**:**	Istanbul Gelisim University Institute of Graduate Studies
Department	**:**	Electrical and Electronics Engineering
Thesis Type	**:**	Master of Science
Date of the Thesis	**:**	02.06.2020
Page Number	**:**	85
Thesis Supervisors	**:**	Asst. Prof. Dr. Sevcan KAHRAMAN
Index Terms	**:**	Hyperspectal imaging, deep learning, deep neural network, deep generative modeling, spectral unmixing, spectral resolution, spectral variability
Turkish Abstract	**:**	
Distribution List	**:**	To the Institute of Graduate Studies of Istanbul Gelisim University

Signature

Khalid BOUNAIL

REPUBLIC OF TURKEY
ISTANBUL GELISIM UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

Department of Electrical and Electronics Engineering

INCREASING SPECTRAL RESOLUTION OF HYPERSPECTRAL IMAGES WHILE DECREASING SPECTRAL VARIABILITY USING DEEP GENERATIVE MODELS

Master Thesis

Khalid BOUNAIL

Supervisor

Asst. Prof. Dr. Sevcan KAHRAMAN

Istanbul – 2023

DECLARATION

I hereby declare that in the preparation of this thesis, scientific ethical rules have been followed, the works of other persons have been referenced in accordance with the scientific norms if used, there is no falsification in the used data, any part of the thesis has not been submitted to this university or any other university as another thesis.

Khalid BOUNAIL

.../.../20...

TO ISTANBUL GELISIM UNIVERSITY
THE DIRECTORATE OF SOCIAL SCIENCES INSTITUTE

The thesis study of increasing spectral resolution of hyperspectral images while decreasing spectral variability using Deep. Generative Modeling. has been accepted as MASTER THESIS in the department of Electrical and Electronics Engineering by out jury.

	Signature	
Director	Asst. Prof. Dr. Sevcan KAHRAMAN	
	Signature	
Member	Dr. Öğr: Üyesi Yusuf Gürcan ŞAHİN	
	Signature	
Member	Dr. Öğr. Üyesi Kenan BÜYÜKATAK	

APPROVAL

I approve that the signatures above signatures belong to the aforementioned faculty members.

... / ... / 20..

Signature

Prof. Dr. İZZET GÜMÜS

Director of the Institute

SUMMARY

Hyperspectral imaging has become an indispensable tool for analyzing Earth's surface materials and extracting valuable information in various applications, including environmental monitoring, agriculture, and remote sensing. One fundamental task in hyperspectral analysis is spectral unmixing, which aims to decompose mixed pixel spectra into their constituent endmember spectra and corresponding abundances. Traditional spectral unmixing methods typically rely on linear models, assuming that the mixed pixel spectra are linear combinations of the pure spectral signatures.

in this thesis, an approach to enhance the spectral resolution of hyperspectral images while simultaneously reducing spectral variability is explored by proposing a technique called Deep Generative Endmember Modeling (DGEM) applied to unsupervised spectral unmixing.

The Linear Mixing Model (LMM) is commonly used for spectral unmixing, which aims to decompose the mixed spectral information in an image into individual pure spectral signatures called endmembers. However, traditional LMM approaches struggle with limited spectral resolution and high spectral variability, which can hinder accurate unmixing results.

To address these challenges, a Deep Generative EM is proposed, which leverages deep generative models to learn the underlying structure of the endmembers. By employing a deep neural network architecture which is capable of capturing intricate relationships and generating high-resolution endmembers with reduced spectral variability.

The proposed Model framework enhances the spectral resolution of hyperspectral images by estimating high-resolution spectral bands based on the learned endmember representations. Additionally, the generated endmembers exhibit reduced spectral variability, resulting in improved unmixing performance.

Key Words: hyper spectral images, deep learning, deep neural network, spectral resolution, spectral unmixing, endmember, MATLAB

ÖZET

Hiperspektral görüntüleme, Dünya'nın yüzey materyallerini analiz etmek ve çevresel izleme, tarım ve uzaktan algılama dahil olmak üzere çeşitli uygulamalarda değerli bilgiler elde etmek için vazgeçilmez bir araç haline gelmiştir. Hiperspektral analizdeki temel görevlerden biri, karışık piksel spektrumlarını kendilerini oluşturan son üye spektrumlarına ve karşılık gelen bolluklara ayrıştırmayı amaçlayan spektral karıştırma işlemidir. Geleneksel spektral karıştırma yöntemleri tipik olarak doğrusal modellere dayanır ve karışık piksel spektrumlarının saf spektral imzaların doğrusal kombinasyonları olduğunu varsayar.

Bu tezde, hiperspektral görüntülerin spektral çözünürlüğünü artırırken aynı zamanda spektral değişkenliği azaltmaya yönelik bir yaklaşım, denetimsiz spektral karıştırmaya uygulanan Derin Üretken Uç Üye Modellemesi (DGEM) adı verilen bir teknik önerilerek araştırılmıştır.

Doğrusal Karıştırma Modeli (LMM), bir görüntüdeki karışık spektral bilgiyi uç üye adı verilen bireysel saf spektral imzalara ayrıştırmayı amaçlayan spektral karıştırma için yaygın olarak kullanılmaktadır. Bununla birlikte, geleneksel LMM yaklaşımları sınırlı spektral çözünürlük ve yüksek spektral değişkenlik ile mücadele eder ve bu da doğru karıştırma sonuçlarını engelleyebilir.

Bu zorlukların üstesinden gelmek için, uç öğelerin altında yatan yapıyı öğrenmek için derin üretken modellerden yararlanan bir Derin Üretken EM önerilmiştir. Karmaşık ilişkileri yakalayabilen ve azaltılmış spektral değişkenliğe sahip yüksek çözünürlüklü uç üyeler üretebilen derin bir sinir ağı mimarisi kullanılarak.

Önerilen Model çerçevesi, öğrenilen uç üye temsillerine dayalı olarak yüksek çözünürlüklü spektral bantları tahmin ederek hiperspektral görüntülerin spektral çözünürlüğünü artırmaktadır. Buna ek olarak, üretilen uç üyeler daha az spektral değişkenlik sergileyerek daha iyi karıştırma performansı sağlar.

Anahtar kelimeler: hiper spektral görüntüler, derin öğrenme, derin sinir ağı, spektral çözünürlük, spektral unmixing, Endmember, MATLAB

ACKNOWLEDGMENT

I would like to offer my heartfelt appreciation to everyone who helped me finish my thesis. Throughout this process, their advice, counsel, and encouragement have been priceless.

First and most of all, I am deeply grateful to my supervisor, Asst. Prof. Dr. SEVCAN KAHRAMAN, for her unwavering guidance, expertise, and patience. Her insightful feedback and constructive criticism have shaped this thesis and enriched my understanding of the subject matter. I am truly thankful for her mentorship and dedication.

I am grateful to my sister Hind, my mother Saadia and my father Abdallah (rest in peace), for their unconditional love, unwavering support, and understanding throughout my academic journey. Their encouragement and belief in my abilities have been a constant source of motivation.

In conclusion, I would like to express my heartfelt appreciation to all those who have contributed directly or indirectly to the completion of this thesis. Their support and encouragement have played a significant role in making this work possible.

TABLE OF CONTENT

CHAPTER THREE
METHODOLOGY

ABBREDIVATIONS

CNN	**:**	Convolutional Neural Network
LMM	**:**	Linear Mixing Model
VAE	**:**	Variational Autoencoder
GAN	**:**	Generalized Adversarial Network
FCLSU	**:**	Fully Constrained Least Square Unmixing
VCA	**:**	Vertex Component Analysis
ICA	**:**	Independent Component Analyses
ELMM	**:**	Extended Linear Mixing Model
GLMM	**:**	Generalized Extended Linear Mixing Model
SAM	**:**	Spectral Angle Mapper
NRMSE	**:**	Normalized Root Mean Square Error
HSI	**:**	Hyperspectral Image

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Human vision is known to extend between 400 and 700nm, in other words, it can only see visible light in three bands which are red, green and blue, however in scientific inquiry, it is necessary to look far beyond than that, into the infrared ranges whereas it can divides the spectrum into several further bands, for this reason a recent advances in optical analysis introduced a technology called hyper spectral imaging that allows for the detection and identification of objects from a large scale of images providing enormous number of wavebands, such imaging is based on the idea that it can record underlying processes (e.g., chemical traits, biophysical qualities, etc.) down to the pixel level, In other words it is based on the idea of a continuous spectrum, or of narrow, contiguous spectral bands, which is crucial and allows for the greatest possible use of the information. Thus, by examining their tiny spectrum features, it is able to recognize surface objects (or atmosphere gases/particles) and evaluate their optical qualities.

Hyperspectral imaging has transformed a study of Earth's surface substances, allowing for exact characterization as well as analysis across several domains such as:

- remote sensing in which, it enables gathering information about the planet’s surface and environment that includes tracking the utilization of land, identifying alterations in vegetation’s health and even monitoring coral reef’s health.
- Agriculture in which hyperspectral imaging can be deployed in maintaining crop health, identifying nutritional deficits, insects and diseases.
- Biotechnology and science such as process control and medical diagnosis which can be used to detect cancerous tissue and skin illness.

This technology relates to a group of methods known as spectrum imaging or spectral analyses that has evolved as an effective and adaptable technique which can collect and process detailed information from across the electromagnetic spectrum with the goal of locating objects, identifying materials, or detecting processes from the acquired spectral images. It can be compared to fingerprints but instead of identifying a person, it identifies materials.

However, to attain maximum result’s efficiency hyperspectral imagery must overcome numerous conditions and circumstances that reduce the capability to interpret and analyze the data which can affect its performance leading to weak

spectral Resolution and high spectral variability. These conditions can be as atmospheric changes or in some cases, it can occur a lot of materials in the same spectral signature as it gets more difficult to correctly define and measure the basic substances.

In order to attain the research problem mentioned above, a proposed models and methods are deployed to overcome them, such as: deep generative modeling which is a class of machine learning approaches that learns the distribution of endmembers in a hyperspectral image using deep neural networks leading in. In other words, this approach enables the generation of realistic samples and the learning of what underlies the data distribution.

Moreover, Spectral unmixing which is able to distinguish the endmembers present in a hyperspectral image scene and determine their abundance fractions in each pixel without any endmember's knowledge being provided. In this case a well-known method in spectral unmixing will be employed, called linear mixing model which is able to overcome the unmixing problem by defining how the endmembers are mixed in the image.

CHAPTER ONE

LITERATURE REVIEW

1.1. Hyperspectral imaging

Hyperspectral imaging, also called "spectro-imaging" as opposed to "multispectral" or "superspectral" imaging, is a good way to work with materials, identify them or define their properties, using the process of capturing a large number of spectral bands (generally more than hundred), when employing hundreds (or even thousands) of narrowband (≤ 10nm) and contiguous channels (Fig.1 and 2) that cover the visible and infrared regions of electromagnetic spectrum and this is done by studying how light interacts with them. "Imaging spectrometers" (Goetz et al., 1985) is the acquisition of a continuous spectrum in a given range of wavelengths, generally corresponding to the visible and near-infrared domains (400 to 2500nm), with a sampling step of about 10nm.

Moreover, it is possible to distinguish and identify the elements of a pixel using hyperspectral imaging by analyzing their "spectral signature" that include specific absorption peaks' wavelength positions in which they are influenced by their chemical make-up, also, the quantity (or concentration) of the elements present will be indicated by their amplitude. Finally, the surface's physical characteristics affect the spectrum's (or continuum's) overall form (granulometry, roughness, humidity, etc.). Thus, it is feasible to determine the different materials present, as well as their concentrations and physical features (e.g., minerals, plants, chemical components, etc.).

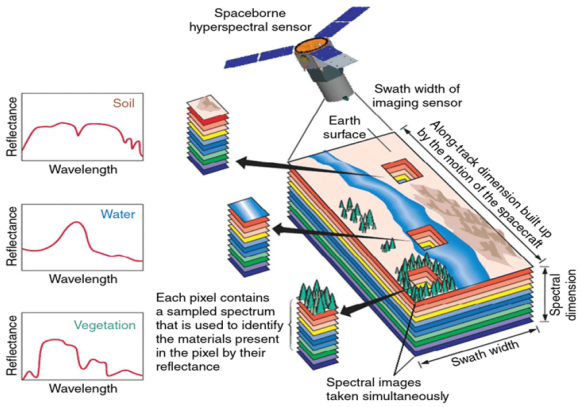


Figure 1: A hyperspectral satellite's underlying concept. The output of a hyperspectral satellite is a data cube and spectrum for each pixel.
Source: Khan M. S., (2018)

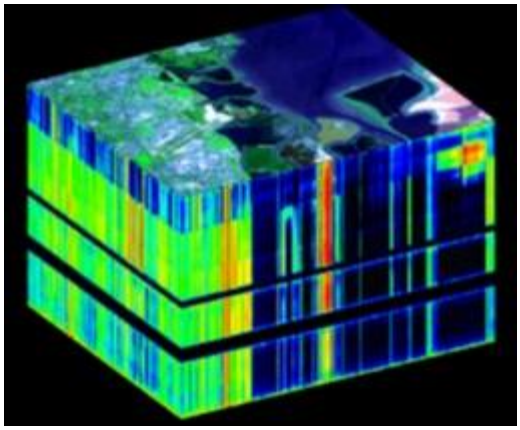


Figure 2 : Example of a "hyperspectral cube" - Image of Moffett Field, CA (San Francisco Bay Area) acquired on August 20, 1992 by the AVIRIS Spectro-imager (NASA/JPL) embarked on an ER-2 stratospheric aircraft.

To understand further, hyperspectral imagery helps in analyzing and evaluating of the reflected (also emitted) radiation detected by a high number of narrow, contiguous and continuous spectral bands, where instead of merely assigning primary colors to each pixel; it can evaluate a broad spectrum of light. This study of light interaction with materials is based on spectroscopy which is done by using a spectrometer is an instruments that splits the incoming light into a spectrum and measures thousands or hundreds of thousands of spectra by examining how light behaves in the target and recognizing materials based on their different spectral signatures, a spectrum information also called a reflectance spectrum is collected, this imaging spectrometer is called the hyperspectral camera and its goal is to collect full spectral and spatial information of the target line by line and as a result a collected spectra presented on a scale of intensity and wavelength which is used later to form an image of the target in a way that each pixel includes a complete spectrum.

Hyper spectral imaging is used frequently in a lot of fields which are remote sensing field that includes Intelligent farming and environmental monitoring also in advanced machine vision that includes food or materials forming and counterfeit detection either illustrating various characteristics of the land (such as surface temperature, soil moisture, agricultural output, defoliation, biomass, or leaf area coverage), the water (such as yellow material, ocean color, suspended particles, or chlorophyll content), or the atmosphere (e.g., temperature, moisture, or trace gases), moreover a well know field which is Biotechnology and science that includes process

control and diagnosis such as using infrared hyperspectral imaging to identify infections without labels, also hyper spectral imaging used in defense and security which includes situational awareness and surveillance.

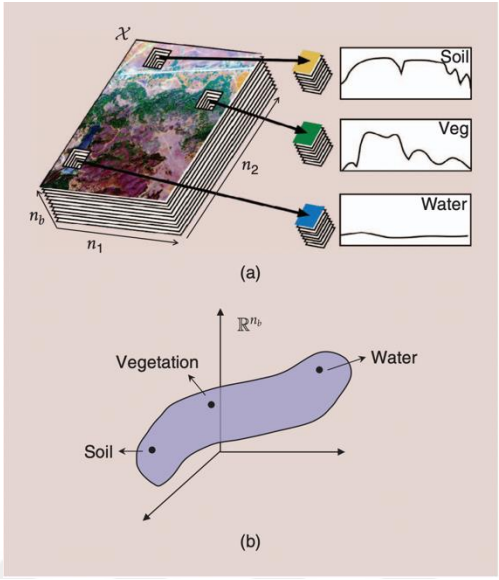


Figure 3: (a) The concept of hyperspectral imaging (b) Hyperspectral vectors expressed within a lower-dimensional manifold.
Source: Bioucas-Dias J. M., (2013)

Moreover, one of the main components in hyperspectral imaging is the hyperspectral camera also called imaging spectrometer which is a line scan camera based on a technology called push-broom technology, this technology gathers the target's whole spectral and spatial data line by line, its key elements are an imaging spectrograph, a grayscale camera and an objective.

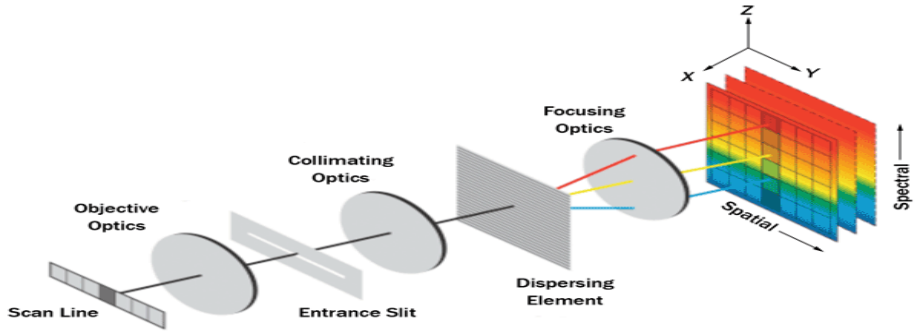


Figure 4: illustration of the hyperspectral camera.
Source: Dell'Endice, F., (2009)

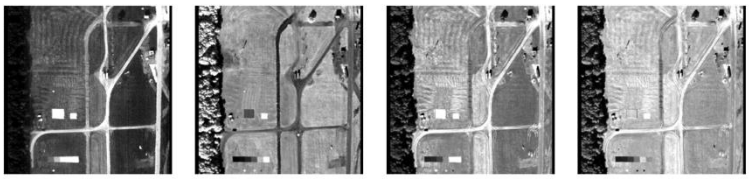
The objective projects the target to the camera sensor where the image is recorded, the second key element in the hyper spectral camera is the imaging spectrograph which contains an input split in which the incoming information is

limited so that it passes only from a single line where the objective form an image into it, a collimating optics which directs the light from the split to the dispersive unit so the incoming light spreads into the spectra, and a dispersive unit as well as a focusing lens which is needed to focus the image to the gray scale camera which used to measure the intensity of the dispersed light, in case of hyperspectral image acquisition, there are various available hyperspectral sensors, each of them has a unique property depending on the application.

The main primary features of these instruments are the type of scanner (pushbroom or wiskbroom), coverage and spectral resolution, viewing angle, and spatial resolution.

For the case of acquisition strategies, there are three types:

a- Spectral scanning techniques that include sequentially capturing every single monochromatic spectral picture, as seen in Fig.4. Up until the whole hyperspectral cube is collected, each acquisition first records the entire 2D scene at a wavelength chosen via a bandpass optical filter. The optical filter is then adjusted to capture the next spectral picture. For this technique, very accurate optical filters that can filter the required wavelength are needed. However, it necessitates that the scene (as well as the sensor) be still for the duration of the capture.



(a) Bande 467nm (b) Bande 746nm (c) Bande 1515nm (d) Bande 1706nm

Figure 5 : Monochromatic spectral pictures.

b- The second category of spectral scanning techniques consist in gathering all of the spectral data for one or more pixels at once and the acquisition is done line by line using two different techniques: push broom or whiskbroom.

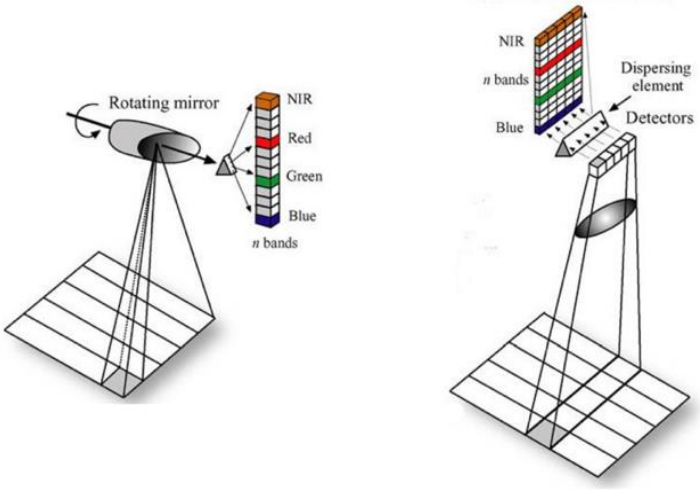


Figure 6: Hyperspectral acquisition with a whiskbroom sensor (left) and a "push broom" sensor (right).

c- The third method is snapshot imaging, which enables the simultaneous capture of spectral and spatial data without the need for scanning by involving "slicing" the image into bands and rearranging these bands into a single line using a sophisticated optical apparatus.

1.2. Spectral resolution

The accuracy with which the planet surface's environment is observed can be primarily depending on the different characteristics and properties that manage how much detail the data about the earth's surface of the satellite's sensors can provide in which, one of these properties is the spectral resolution that has been a fundamental concept that can serve in improving material discrimination, accurate quantitative analysis and also feature's similarities discrimination, it is an information provided by the sensor's ability to distinguish between the ranges of the electromagnetic spectrum by detecting precise details that can help in enhanced decision making and effective analysis.

In hyperspectral imaging, spectral resolution is crucial as it's central key component that determines the spectral data's amount of detail and discriminating skills in which the hyperspectral sensor is able to record, by distinguishing features from the hyperspectral images, it enables to go more deeply within the domain of spectral analysis to acquire vital data that is typically hidden in imaging analysis. Thus, spectral resolution has several advantages in hyperspectral imaging, two mains of them are fine discrimination and detailed analysis, in which, the high spectral resolution can

capture subtle details and fine spectral variations, making it easier to distinguish between different materials and features in the scene as well as is can allow for more precise analysis of the spectral properties of objects, which can be useful in various applications, such as change detection, material classification, vegetation mapping, etc.

In this context of hyperspectral imaging, the spectral resolution can contain two main aspects which are the number of spectral bands and their width.

- Number of spectral bands:

  The term "spectral bands" relates to the total number of distinct wavelength ranges or channels used by the hyperspectral sensor to sample the electromagnetic spectrum in which the higher of spectral bands, the higher ability to analyze and record fine-scale of spectral variations.

- Width of spectral bands:

  The range of wavelengths covered by each particular band is referred to as the width of spectral bands which is defined by the sensor's spectral filter as shown in the table 1 or dispersive element, that distinguishes and narrows certain wavelengths into distinct sensors, the narrower band widths, the more enhanced spectral resolution which is done by reducing the range of wavelengths collected within each band, allowing for a more accurate separation and more precise discrimination of spectral characteristics.

Sensor	**Number of bands**	**Wavelength range**	**Spectral bandwidh**
AVIRIS	224	360-2500nm	9.6nm
ROSIS	115	430-860nm	4nm
Hyperion	220	400-2500nm	10-11nm
HyspIRI	212	380-2500nm	10nm

Table 1: some of the famous hyperspectral sensors.

However, regardless of the hyperspectral sensor's capacity to capture a large number of spectral bands, the spectral resolution is still generally lower than the spatial

resolution, which can result in significant information loss which refers to the limited ability of hyperspectral sensors to capture fine, detailed spectral information leading to several challenges in the analysis of hyperspectral data such as:

- High dimensionality:
  Hyperspectral images with high spectral resolution can have a large number of bands, which increases the dimensionality of the data. This poses challenges in terms of data storage, processing and analysis.
- Spectral variability:
  The presence of spectral variations due to factors such as atmospheric conditions, lighting, shadows, etc., can make the interpretation and analysis of hyperspectral data more complex. Spectral variability can mask subtle differences between materials and affect the accuracy of results.
- Difficulty distinguishing similar materials:
  The reduced spectral resolution can make it difficult to distinguish between materials with similar spectral signatures. This can lead to confusion in classifying materials or identifying specific characteristics of a scene.
- Spectral interference:
  When two or more materials have similar spectral signatures in one or more bands, it can be difficult to distinguish them with limited spectral resolution. This can lead to errors in identifying and quantifying the materials present in a scene.

To overcome these spectral resolution problems, several approaches and methods have been developed in which the main objective them is to exploit the available spectral information as much as possible and to reduce the spectral resolution limitations in order to improve the accuracy and quality of the hyperspectral analysis, the most known of them are as follows:

- Atmospheric Processing:
  Atmospheric corrections are intended to remove or reduce the effects of the atmosphere on hyperspectral data. This can involve removing atmospheric scattering effects, correcting for illumination variations, and restoring reflected radiometric values.

- Spatial and Spectral Filtering:

Filtering techniques can be used to reduce noise and variability in hyperspectral data. Spatial and spectral filters, such as mean filters, median filters, anisotropic filters, can be applied to smooth the data and improve image quality.

- Dimension reduction techniques:

By projecting the data into a reduced dimensional space, dimension reduction algorithms attempt to extract the most relevant characteristics from hyperspectral data which minimizes the data's dimensionality and keeps only the most important information for analysis.

- Deep learning approaches:

Deep neural networks, such as convolutional neural networks (CNNs) and variational autoencoders (VAEs), can be used to automatically extract discriminative features from hyperspectral data and reduce the variability spectral.

1.3. Spectral features and its extraction

Spectral features are the properties or spectral signatures that a material exhibits when assessed across different wavelengths in hyperspectral imaging. Every material has a unique spectral signature that corresponds to its interactions with light at various wavelengths in which a material's spectral properties are often represented as a reflectance spectrum or a vector of reflectance values at distinct spectral bands. Each reflectance value in the spectrum represents the amount of light reflected by the material at a certain wavelength. As a result, spectral properties define how a material's reflectance vary with wavelength.

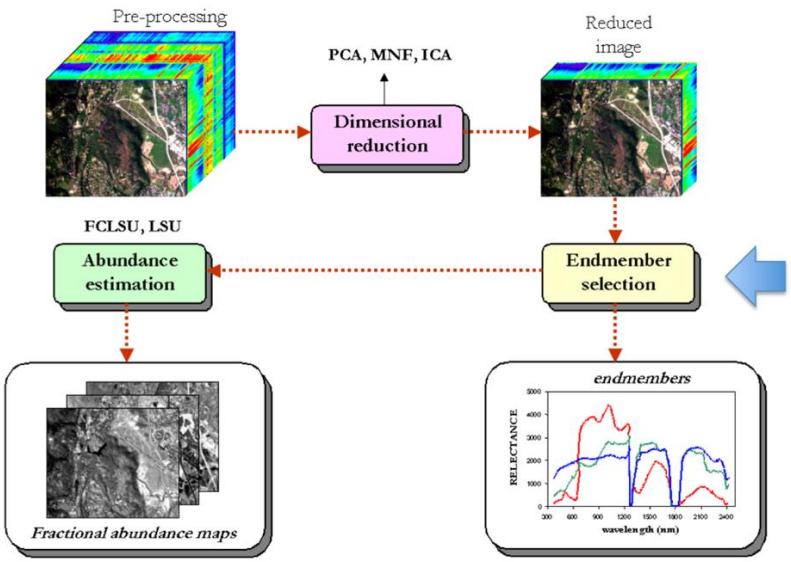


Figure 7: feature extraction process.
Source: Sánchez, S., (2010)

In hyperspectral imaging, spectral features play an essential role as they refer to the properties of objects and materials that manifest themselves through their unique spectral signature.

As mentioned before, hyperspectral imaging records data in many narrow spectral bands, often across the entire electromagnetic spectrum, providing detailed information about the reflection or emission of light from objects. Each spectral band records the amount of light reflected or emitted at a specific wavelength. Thus, spectral characteristics are therefore used to reflect variations in the spectral signatures of objects and provide valuable information about their composition, structure and physical properties.

This can be done by preforming several methods independently such as:

1.3.1. Independent Component Analysis (ICA)

ICA is another dimensionality reduction technique that aims to find statistically independent components in hyperspectral data. It assumes that the observed spectra are linear mixtures of underlying independent sources. By separating these sources, relevant features can be extracted.

1.3.2. Principal Component Analysis (PCA)

PCA is a widely used technique for dimensionality reduction as it transforms hyperspectral data into a new set of orthogonal variables called principal components. The principal components are ranked according to their contribution to the data variability, and a subset of these components can be selected as characteristic.

1.3.3. Non-Negative Matrix Factorization

NMF is a popular approach for obtaining spectral signatures as it deconstructs the hyperspectral data into a set of endmembers and their abundances in which, these endmembers, reflect the spectral fingerprints seen in the data, whereas the abundances show the relative contributions of each endmember to the observed spectra.

1.3.4. Deep learning approaches

To extract spectral signatures, deep learning techniques such as autoencoders, convolutional neural networks, and recurrent neural networks can be used which enables the extraction of significant spectral signatures as well as capturing sophisticated spectral patterns by learning hierarchical representations from hyperspectral data.

1.4. Spectral unmixing

In hyperspectral imaging, unmixing is particularly important when there are little items in the picture where each pixel in a hyperspectral picture may be thought of as the reflection spectrum of the region being scanned, and these images are rich in both spectral and geographical information in which they are constructed of many materials, each of which has a distinct reflectance spectrum and according to these materials present in each pixel in which, each pixel carry spectral data, however, it sometimes occurs that a single pixel comprises many materials, in this case, the observed spectral data is a combination of the various material's separate spectra. For this case, spectral unmixing is an essential approach that comes in an emerging between the spare unmixing and deep learning approaches for remotely sensed hyperspectral image interpretation.

Spectral unmixing involves a series of steps to interpret the measured spectrum of a mixed pixel. These steps include dividing the spectrum into constituent spectra called endmembers and determining the corresponding fractions or abundances that

indicate the proportion of each endmember present in the pixel. In simpler terms, the spectral unmixing process expresses each pixel vector, which may be a mixture, as a combination of pure spectral signatures (endmembers) and their respective fractional abundances.

1.4.1. Endmember selection

Endmembers are pure spectral signatures that indicate the spectral features of specific components in the scene. Endmember selection enables in finding and collecting a group of reference spectra that indicate the components in the hyperspectral image.

1.4.2. Endmember extraction

Once the endmembers have been determined, the endmember spectra must be extracted from the hyperspectral data which is done by through searching to identify the optimal linear combination of endmembers which corresponds to the observed mixed pixel spectrum.

In this thesis, endmember extraction is crucial for spectral unmixing as it identify the endmembers or spectral signatures from the hyperspectral image, the most famous algorithms that have been developed and used are:

- N-FINDER (N-dimensional Visual Feature Extraction)

The N-FINDR (N-dimensional Visual Feature Extraction) algorithm is a commonly used method for estimating abundance fractions of constituent materials present in a mixed hyperspectral scene, it is based on the assumption that each mixed pixel can be represented as a positive linear combination of pure endmembers and the objective is to identify the pure endmembers and estimate the corresponding abundance fractions.

- Pixel Purity Index (PPI)

The PPI algorithm is also a method in spectral unmixing to identify pure or unmixed pixels which operates by evaluating the degree of spectral purity of each pixel in the image while calculating a purity index for each pixel using statistical measures such as standard deviation, spectral distance, spectral coherence and other characteristics leading to facilitating spectral unmixing by providing references for estimating abundance fractions of constituent materials in the mixed pixels.

- Vertex Component Analysis (VCA)

In this thesis report the algorithm used will be based on the VCA algorithm, Vertex component analysis is a technique used in hyperspectral imaging to extract information about spectral mixtures present in a scene which focuses on identifying the pure endmembers that represent the constituent materials in hyperspectral data (Borsoi et. al., 2018), in other words, it is a method used in the field of spectral unmixing to estimate the abundance fractions of pure endmembers in a hyperspectral scene and its main objective is to detect the endmembers present in the mixed pixels by exploiting the convexity of all the spectra.

The VCA algorithm works by taking a hyperspectral data matrix as input in which each column indicates a pixel in the image and p is the needed number of endmembers, then, by using singular value decomposition (SVD) or principal component analysis (PCA) the data is then projected into a lower-dimensional subspace and normalized in which it can also enables in reducing noises and redundancies in the data, The VCA algorithm proceeds in iterations in which, at each iteration, it identifies a potential endmember by searching for the spectrum that maximizes the angle with the other endmembers already identified and continues this iterative process until a predefined stopping criterion is reached, such as a fixed number of desired endmembers or sufficient convergence of abundance fraction estimates. Once the endmember is selected, the algorithm uses orthogonal projection to update the estimated abundance fractions for the mixed pixels.

- Sequential Maximum Angle Convex Cone (SMACC)

The SMACC algorithm operates using an iterative approach for estimating abundance fractions which is done by starting to select a reference pixel and identifying it as a potential endmember, further, it constructs a convex cone from this endmember using the other mixed pixels, and then, identify the pixel that forms the maximum angle with the existing convex cone which is added as a potential new endmember. By repeating this process iteratively, the algorithm searches for the next endmembers that maximize the angle between the convex cone and the remaining pixels which allows the contribution of the constituent materials in the mixed pixels to be quantified and facilitates spectral unmixing.

- Minimum Noise Fraction (MNF)

The MNF algorithm is used in the field of spectral unmixing to reduce noise and correlations between spectral bands in a hyperspectral image which is based on a Vertex Component Analysis adapted to hyperspectral data where it transforms the data into a new space called MNF space where the information is better represented, thus facilitating spectral unmixing by reducing the dimensionality of the data while preserving the relevant information (Rajabi, 2014).

1.4.3. Abundance estimation

Thereafter, it comes the abundance fraction estimation of every single endmember in which, in spectral unmixing, refers to the process of estimating the abundance fractions of different endmembers present in a hyperspectral scene. In hyperspectral imaging, the materials and objects in a scene are often linearly mixed, meaning that each pixel in the image is a linear combination of the spectral signatures of the endmembers, in this case, abundance estimation involves determining the proportions or fractions of each endmember contributing to the spectral signature of a given pixel.

A hyperspectral image may be analyzed as a collection of sampled spectra that correspond to the unique spectral signatures of each of the materials in the observed image, known as endmembers, as well as the locations of each of these endmembers in the image, however, due to insufficient spatial resolution and spatial complexity which occurs when a pixel contains many materials, this map, known as an abundance map, shows the percentage of each endmember in that pixel.

Moreover, sometimes when it comes to break down a picture into a list of source spectra, known as endmembers, that correspond to the reflection spectra of the scene which has many elements on the one hand, and the percentage of each of these source spectra in each individual pixel on the other hand, hyperspectral imaging uses the unmixing technique in which there are several , however they are less successful when there are so-called uncommon source spectra (i.e. spectra present in very few pixels, and often at a sub-pixel level). These uncommon spectra may be thought of as abnormal whose identification is frequently important for certain applications because they correlate to components that are present in the scene in small quantities.

To estimate the fractional abundance of endmembers inside mixed pixels, these approaches use various assumptions and models. Among the most popular are as follows:

1.4.3.1. Non-negative Matrix Factorization (NMF)

The Non-negative Matrix Factorization (NMF) technique which is used in the field of spectral unmixing for the purpose of breaking a matrix of data into two non-negative matrices which assumes that the spectral signatures of the pixels can be expressed as a positive linear combination of the spectral signatures of the endmembers, multiplied by non-negative abundance fractions, leading in representing endmembers and abundance fractions respectively by representing the spectral mixtures of pixels as a non-negative combination of endmembers, thus providing an estimate of the abundance proportions of different materials in a hyperspectral scene.

1.4.3.2. Sparse Unmixing:

Sparse unmixing techniques assume that the abundance fractions are sparse, which means that only a few endmembers contribute significantly to the spectrum of a mixed pixel while the other endmembers have negligible abundance fractions which helps in identifying the most dominant endmember and their abundance fraction, and this process is done by sparse representation technique which are:

a- Variable Splitting (SUNS)

b- Orthogonal Matching Pursuit (OMP)

c- Sparse Bayesian Unmixing (SBU)

1.4.3.3. Bayesian-based approaches:

In the context of spectral unmixing, Bayesian approaches consider the problem probabilistically where they use statistical models to represent the data and unknown variables, such as modeling uncertainty and incorporating historical endmember and abundance fractions. the Bayesian linear unmixing (BLU), Bayesian abundance estimation (BAE), and Markov Chain Monte Carlo (MCMC)-based methods are an example of such statistical models.

1.4.3.4. The Mixing models

- The Linear and Non-Linear Mixing Models

The primary objection in spectral unmixing is to estimate both the endmembers set present in the picture and their abundance map at the same time, for this case

unmixing algorithms are based on certain mixing models, which can be classified as Linear or Non-Linear.

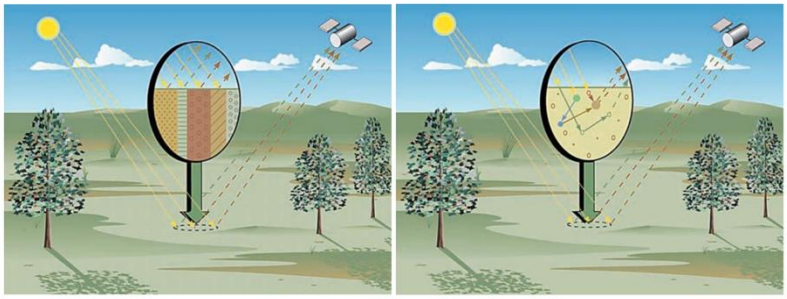
The Non-Linear Mixing model assumes that incoming radiation interacts with numerous components and is impacted by various scattering effects, and it's occurred when deviations from linearity are caused by interactions between various materials, hence, it typically involves previous knowledge of object geometry and physical properties of the observed items.

In the other hand, the combination will be supposed to be linear if the endmembers in a pixel appear in spatially separated patterns, in this case, a single component on the surface dominates the scattering and absorption of incoming electromagnetic radiation in any region of the surface. Linear Mixing Model which is the deployed method in this thesis, has a number of practical advantages in which it can overcome the unmixing problem by defining how the endmembers are mixed in the image, including ease of implementation and adaptability in a variety of applications.

To illustrate more, the observation of multiple materials within a single pixel can have two origins:

The first is due to the spatial resolution of the sensors, and that means that the materials are spatially separated but occupy an area of the scene smaller than that covered by a pixel which in this case, just one substance of the seen image interacts with each incoming light from the image and the measured spectrum is thus a linear mixture of the individual surface material spectra, and the mixing coefficients correspond to the surface proportions of each component (Fig.6.a).

In the other case, the observed surface can be composed of a single entity which is an intimate mixture of several materials, such as a sand whose grains have various chemical compositions. The incoming light can then be reflected several times by different materials before being sent back to the measuring instrument. These multiple reflections lead to the observation of a spectrum which is a non-linear mixture of the pure spectra of the components present (Fig.6.b).



(a) (b)

Figure 8: Different origins of spectral mixtures (a) Linear Mixing. (b) Non-linear Mixing.

The hyperspectral image is described by the linear equation (Borsoi, 2019):

$$x_p = \sum_{k=1}^{K} a_{p,k} s_k + e_p \qquad \text{Equation 1}$$

Thus, the matrix compact version of the LMM may be defined as follows for all pixels:

$$X = AS + E \qquad \text{Equation 2}$$

With:

- X is the matrix of the image.
- A is the abundance matrix obtained by correlating the abundance vectors a_p.
- S is the matrix of endmembers.
- E is the noise matrix which corresponds to the correlation of the noise vectors e_p.

The mixing model must follow specific restrictions in order to be physically realistic, as well as the endmembers matrix's coefficients cannot be negative since it comprises the precise reflection spectra of each substance in the image being observed, moreover, because the abundance matrix A includes the endmember proportions for each pixel, its coefficients are thus positive and for each pixel, the sum of the abundances constituting it must be equal to 1.

However, although this assumption results in a well-posed and computationally simplified framework, it restricts the LMM's application since it can risk the accuracy of predicted abundances in many situations due to EM spectral variability in which, for this case, in order to account for nonideally factors like nonlinearity and endmember variability, which are frequently observed in actual scenarios, enhanced

unmixing extension of the LMM have been developed which is The Extended Linear Mixing Model.

- Extended Linear Mixing Model (ELMM)

Various limitations in the LMM are considered which reflect on the accuracy of the abundance estimation such as endmember variability which in this case endmembers may exhibit variability do to illumination or sensor noise conditions, and additional constraints such us nonlinearity behavior or simplicity which is the opposite of what the LMM's assumption (Drumetz, 2020).

The Extended Linear Mixing Model (ELMM) was developed as an extension of the Linear Mixing Model (LMM) to overcome these limitations by considering additional factors and constraints that were not accounted for in the LMM, the ELMM introduces various improvements to enhance the accuracy of unmixing results. These enhancements include addressing nonlinearity by using nonlinearity correction approaches, accounting for variations in endmembers, and incorporating additional constraints.

The extended linear mixing method (ELMM) is a technique that is frequently used in hyperspectral data analysis to model and analyze the mixtures of materials found in a scene. It works under the assumption that the spectrum of an observed pixel can be thought of as a linear combination of the pure spectra of the constituent materials present in that pixel.

The two global equations for the Extended Linear Mixing Model are:

$$x_n = \sum_{p=1}^{P} a_{pn}\ \psi_{pn}\, s_{Op} + e_n = S_0\, \psi_n a_n + e_n \qquad \text{Equation 3}$$

Where: x_n in the input image, a_n is the abundance coefficient for pixel n, ψ_n is a diagonal matrix which contains the scaling factor for each material and $\psi_n \in \mathbb{R}^{P\times P}$, S_0 is the reference endmember, and e_n is the additive noise.

After recovering the model, the Equation (3) will be written as:

$$X = S_0(\Psi \odot A) + E \qquad \text{Equation 4}$$

Where $\odot$ is the Hadamard product, Ψ is the scaling factor matrix and A is the abundance matrix.

The fundamental idea of ELMM is that each pixel in a hyperspectral image is composed of a mixture of several pure materials, such as soil, vegetation, buildings, water, etc. Each pure material has a known characteristic spectrum associated with it. Each pure material has a known characteristic spectrum associated with it. ELMM aims at estimating the fractions or proportions of each pure material in each pixel of the image that enables to find the optimal values for endmembers and abundances that can effectively represent the mixed spectra observed in the data. However, this task is approached as an optimization problem where the aim is to minimize the difference between the observed spectra and the spectra reconstructed using the estimated linear combinations of endmembers.

In this case, ADMM, which stands for Alternating Direction Method of Multipliers, is a powerful optimization method used in the ELMM framework. It is an iterative algorithm that breaks down a complex optimization problem into smaller, more manageable subproblems. By doing so, ADMM simplifies the overall problem by applying the following steps:

To begin with, the ELMM algorithm starts by initializing the values of endmembers and abundances and then proceeds with iterative steps until it reaches convergence, indicating a satisfactory solution, further, the algorithm modifies the endmembers on each iteration by reducing the goal function while holding the abundances constant, in which, this particular step aims to improve the accuracy of the estimated endmembers by iteratively refining their values. Next, the objective function is minimized while maintaining the fixed endmembers which is done by adjusting the abundances, and thus, the abundance estimations will be enhanced by this procedure.

Finally, to ensure the consistency between the updated endmembers and abundances throughout the optimization process, a dual variable is updated by incorporating a scaled difference between them. This step plays a crucial role in aligning the estimates of endmembers and abundances, promoting convergence towards a coherent solution.

Through an iterative process that takes into account the interdependencies between the endmembers and abundances, ADMM aims to discover a solution that simultaneously fulfills the specified constraints and optimizes the objective function. By systematically updating the endmembers and abundances in a coordinated manner,

the algorithm strives to converge towards an optimal solution that balances the constraints and objectives of the problem.

- Generalized Linear Mixing Model (GLMM)

Another unmixing method is proposed which is the Generalized Linear Mixing Model which is an approach for estimating the amounts of different materials present in mixed pixels and is frequently employed in remote sensing and hyperspectral images as it offers a solid theoretical basis for modeling the spectrum mixing process and is widely utilized in a variety of applications, including land mapping, environmental change detection, and natural resource monitoring (Borsoi et. al.,2018).

GLMM is founded on the core assumption that the spectral behavior of a mixed pixel can be represented by combining the spectral characteristics of individual pure materials, known as endmembers. This combination is achieved through a linear relationship, where the spectra of the mixed pixels are constructed by multiplying the spectra of the endmembers by abundance coefficients that represent their respective proportions within the mixture, each pixel x_n is written as:

$$x_n = (S \odot \Psi_n)\alpha_n + e_n \qquad \text{Equation 5}$$

Where: S is the $L \times$ R endmember matrix, Ψ_n is scaling matrix with $[\Psi_n]_{l,k} = \psi_{n_{l,k}} \geq 0$.

GLMM offers great flexibility, allowing the use of different statistical distributions to model residuals and to take into account various sources of variability as well as the capability to be adapted to account for atmospheric effects, illumination, shadows, instrumental noise, and other factors that can influence the spectral response of mixed pixels.

Different methods, including the least squares method, maximum likelihood estimation, and Bayesian methods, can be used to estimate the proportions of endmembers in the GLMM. The main goal of these methods is to solve an optimization problem that aims to minimize the difference between the observed spectra and the spectra estimated from the proportions of endmembers.

However, GLMM does have several difficulties and restrictions, it should be acknowledged. One of the most difficult obstacles is identifying the number of endmembers in the data. Overestimating or underestimating the number of

endmembers can result in considerable inaccuracies in percentage estimation. Furthermore, GLMM implies a linear mixing process, which may not be the case in some real-world scenarios. Sensor saturation, non-linear illumination effects, and material interactions can all contradict this assumption and cause estimate mistakes.

- Generalized full Extended Linear Mixing Model (GELMM)

The Generalized Extended Linear Mixing Model, an extension version of GLMM designed specifically for this situation, differs from the original model primarily in the way it addresses the linearity of the mixing process where the primary benefit of the GELMM is its capacity to consider non-linearities and supplementary factors that impact spectral response.

Comparing to the Generalized Linear Mixing Model, the GELMM enables the exploration of intricate relationships between endmembers and various variables, including illumination effects, shadows, and non-linear transformations, in which, this enhanced capability allows for a more precise representation of the connection between endmember proportions and the spectra of mixed pixels by introducing non-linear features into the model which enables the GELMM to more effectively manage scenes where the spectrum properties of materials varies non-linearly with their proportions and this is particularly important when lighting circumstances vary, as well as in images captured at different times of day or with changing illumination angles.

- The Perturbed Linear Mixing Model (PLMM)

The perturbed linear mixing model is an approach used in hyperspectral imaging to model mixed pixels more realistically. Instead of the classical linear mixing model, which assumes a strictly linear relationship between the spectra of the mixed pixels and the proportions of the endmembers, the perturbed linear mixing model considers the perturbations and non-linear effects that can occur during the mixing process.

Assuming that the number of endmembers K is known, the proposed PLMM varies from the standard LMM in that each pixel x_n is represented by a combination of the K endmembers (indicated as m_k) that has been modified by a perturbation vector $dm_{n,k}$ that accounts for endmember variability. The final PLMM can be written.

The PLMM equation can be written as:

$$x_n = \sum_{k=0}^{K} a_{kn} (m_k + dm_{n,k}) + e_n \quad \text{for n=1,...,N} \qquad \text{Equation 6}$$

Where:

- x_n is the nth pixel of hyperspectral image.
- a_{kn} is the proportion of the kth endmember in the nth pixel of the image.
- $m_1, \dots, m_k$ is the linear combination of the K endmember.
- $dm_{n,k}$ denotes the perturbation of the kth endmember in the nth.
- e_n represents the noise caused by data collection and modeling errors.

The PLMM equation can be expressed as a matrix as follows:

$$X_n = MA + [dM_1 a_1 | \dots | dM_N a_N] + E_t \qquad \text{Equation 7}$$

Where:

- X_n is an $N \times L$ matrix that contains the image pixels.
- M is an $L \times K$ endmember matrix.
- A is an $K \times N$ the abundance matrix.
- dM_N is an $L \times K$ matrix where its columns contain the perturbation vectors.
- E_t is the noise matrix.

In the context of hyperspectral imaging, where every individual pixel is identified by a continuous spectrum of spectral values, the perturbed linear mixing model takes into account various sources of disturbance such as atmospheric effects, surface interactions, instrumental errors, illumination variations, shadows and other factors that can affect the spectral response of mixed pixels.

Utilizing the perturbed linear mixing model in hyperspectral imaging offers several benefits. Primarily, it enhances the representation of intricate material-environment interactions, resulting in more precise estimations of endmember proportions. Additionally, by incorporating disturbances, it becomes possible to mitigate estimation errors and enhance the overall quality of spectral unmixing outcomes.

- Least square Unmixing methods

The least squares unmixing technique is a popular method for estimating the quantities of distinct materials (endmembers) in a mixed pixel spectrum in which the linear combination of endmember spectra weighted by their abundances and the observed mixed pixel spectrum are assumed to have a linear relationship. The least squares unmixing approach comes in two frequent iterations:

a- Constrained Least Square Unmixing (CLSU)

Constrained least squares unmixing is a spectral unmixing method widely used to estimate the proportions of different materials (endmembers) in the spectral signature of a mixed pixel where, it assumes a linear relationship between the observed spectrum of the mixed pixel and the linear combination of endmember spectra weighted by endmember abundance.

In this method, extra limitations are placed on the abundance values while performing the unmixing procedure which can encompass non-negativity (abundance values cannot be negative), sum-to-one (the sum of abundances for all endmembers must be equal to one), or other constraints derived from prior knowledge about the materials being unmixed. By integrating these constraints, the unmixing algorithm guarantees that the estimated abundances conform to the anticipated characteristics of the endmember proportions.

Constrained least squares unmixing includes the formulation of an optimization problem where the goal is minimizing the sum of the squared deviations between the observed mixed pixel spectrum and the linear combination of the endmember spectra weighted by their abundances. This optimization issue can be handled using a variety of strategies, including direct solvers, iterative methods, and convex optimization algorithms.

b- Fully Constrained Least Square Unmixing (FCLSU)

Fully constrained least squares unmixing is an advanced method for estimating the number of endmembers present in a mixed pixel spectrum in which, it seeks to address the unmixing problem by taking into consideration both endmember abundance limitations and spectral constraints (Heinz et. al., 2001).

The aim of this method is to handle the Abundance Sum-to-One Constraint (ASC) in linear unmixing problems, so in order to take care of the ASC, the

FCLSU method tackles the challenge of decomposing hyperspectral data into its constituent materials while simultaneously imposing constraints on the abundance fractions. These constraints are part of the Fully Constrained Least Squares (FCLS) model, which comprises three specific requirements:

- Non-negativity Constraint: This constraint ensures that the abundance fractions, representing the proportions of materials in a pixel, cannot be negative in which, it aligns with the fundamental principle that materials cannot have negative amounts within a given pixel.
- Abundance Sum-to-One Constraint (ASC): The ASC constraint guarantees that the total sum of abundance fractions for each pixel equals one in which it signifies that the entire pixel's composition is accounted for by the identified materials, leaving no fraction unexplained.
- Endmember Convex Constraint: Under this constraint, it is assumed that the endmembers, which represent the pure spectral signatures of materials, lie within the convex hull formed by the observed spectra and this can imply that the endmembers are linear combinations of the observed spectra and are confined within the range defined by the observed data.

  In order to take the ASC into account, the ASC in the matrix signature M is embedded by defining a new matrix signature which is expressed by N and defined by (Heinz et. al., 2001):

$$N = \begin{bmatrix} \delta M \\ 1^T \end{bmatrix} \qquad \text{Equation 8}$$

Where: $1 = (1, 1, \ldots, 1)^T$ and a vector s by $s = \begin{bmatrix} \delta r \\ 1 \end{bmatrix}$ Equation 9

Where, the impact of the ASC is controlled by the use of δ in these two equations, allowing the FCLS model to incorporate the sum-of-abundance (SCA) constraint when analyzing linear spectral mixtures. This is achieved by introducing an extra endmember into the model that represents the ASC. This additional endmember is a spectral vector consisting of constant values set to one, indicating the constraint of the abundance fractions summing to one.

To solve the modified FCLS problem, the FCLSU method applies the Least Squares method to estimate the abundance fractions. By integrating the ASC into the analysis, the FCLSU method offers a more precise depiction of the material proportions within each pixel of the hyperspectral image.

1.5. Spectral variability

Spectral variability is a frequent phenomenon seen in many settings in which the spectral fingerprints of the pure component elements changes over the hyperspectral image, it can also be associated with the concept of a pure substance, as an example of signatures that contains a single plant species that differ widely depending to growth and climatic circumstances.

Spectral variability in hyperspectral imaging, refers to the variation in the spectral signatures of different materials or objects present in a scene captured by a hyperspectral sensor. Each substance contains its unique spectral signature that is determined by how it interacts with light at various wavelengths, in other words, spectral variability refers to variances in the spectral profiles of the components present in an image and the presence of this phenomenon can lead to substantial errors in estimation that are carried forward during the unmixing process.

Spectral variability can be a result of various factors such as the chemical composition of the materials, their physical structure, their surface condition, their density, the mainly three conditions are:

1.5.1. atmospheric conditions

When measuring ground reflectance, atmosphere interference is one of the primary causes of spectral variability where, significant quantities of electromagnetic radiation including visible and infrared light are absorbed by three main occurrences which are:

1.5.1.1. Atmospheric absorption

Certain atmospheric gases (such as oxygen, ozone, methane, carbon and others) that can be significantly wavelength based, also aerosols which can change smoothly throughout the spectrum, and, most particularly, water vapor, whereas other molecules and vapors distort the light that comes in which can affect the measured radiance at the sensor, thus, it leads to spectral variability by causing a great deviation on the required ground reflectance.

1.5.1.2. Atmospheric scattering

Light interacts with gas molecules, aerosols, and other particles as it passes through the Earth's atmosphere where, the spectral characteristics of the entering light changes as an outcome of this scattering process in which, it can reduce the intensity of specific wavelengths and affect the general structure of the spectral signature, factors such as air composition, humidity, and aerosol concentration all impact atmospheric scattering, resulting in spectral variability.

1.5.1.3. Path length variation

In response to atmospheric conditions, the light path length between the target object and the sensor could possibly change that is caused by factors such as air density, temperature gradients, and altitude variations, resulting in variations in the quantity of atmosphere by which the light passes. The quantity of air scattering and absorption experienced by the light is affected by this change, resulting in spectral variability.

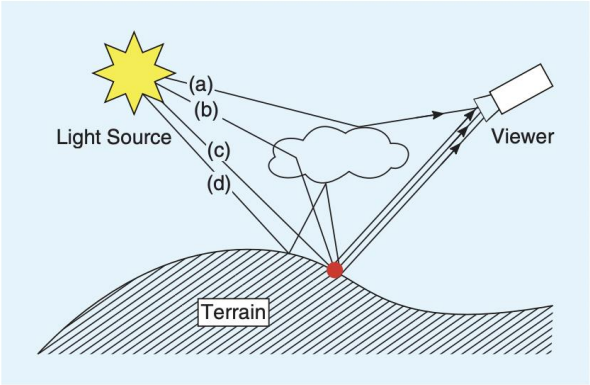


Figure 9 : The atmospheric impacts on the obtained hyperspectral image.
Source: Borsoi R. A., (2021)

The radiation sources in the figure above are: (a) light directly reflected by the atmosphere towards the sensor; (b) light scattered by the atmosphere and reflected by the ground; (c) light directly reflected by the ground; and (d) light reflected by surrounding areas on the ground and then scattered towards the sensor.

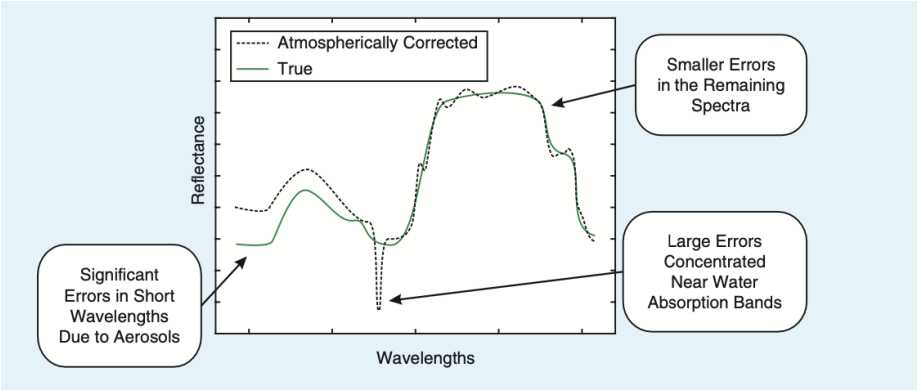


Figure 10: Illustration of the effect of atmospheric circumstances on spectral variability.
Source: Borsoi R. A., (2021)

1.5.2. Illumination and Topographic effects

Changeable illumination and topographic occurs when spectral signatures of the materials constituent natural variation such as physicochemical differences, in the one hand, Illumination refers to the way the scene is illuminated, typically by sunlight and it can occur when the variation in the angle of incidence of light, shadows cast by surrounding objects, and atmospheric conditions cause variations in the spectral reflectance of objects leading to the ability of the same surface to appear differently depending on the angle of illumination, which contributes to spectral variability.

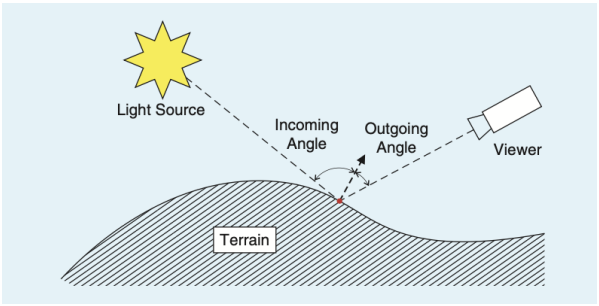


Figure 11: an example of the angle of Illumination.
Source: Borsoi R. A., (2021)

In the other hand, Topographic effects refer to changes in spectral reflectance caused by the topography of the terrain. Shaded areas, slopes, surface orientations, and elevation changes can influence the amount of light reflected from objects, resulting in spectral variations. For example, a surface facing the sun may have a higher reflectance than a surface facing another direction.

1.5.3. Intrinsic spectral variability

Intrinsic spectral variability refers to the natural variability in the spectral signatures of materials present in a hyperspectral scene which is caused by intrinsic factors such as material composition, molecular structure, surface roughness, density, particle size and shape, object orientation and so on, where, each material has unique spectral properties that can vary depending on these intrinsic factors.

1.5.4. Methods for decreasing HI's spectral variability

These mentioned above effects can introduce additional variability into hyperspectral data, which can make the interpretation and analysis of spectral information more complex, however, they are several methods and techniques that mitigate these effects and improve the accuracy of hyperspectral analyses and decreasing the spectral variability which are:

1.5.4.1. Spectral Library Purification

Spectral library purification is a method used to reduce spectral variability in hyperspectral imaging. The spectral library refers to a collection of spectral signatures of pure materials or endmembers. However, these libraries may contain impurities or outliers that can introduce unwanted spectral variability into the spectral unmixing process.

The purpose of spectral library purification is to improve the quality of the spectral library by identifying and removing impurities or outliers which can be accomplished by statistical techniques or filtering methods that identify spectral signatures that do not match the characteristics of the pure materials in the scene.

The goal of spectral library purification is to reduce spectral variability by ensuring that the spectral signatures in the library accurately represent the pure materials present in the hyperspectral scene leading in improving the accuracy of the estimation of abundance fractions and endmembers during the spectral unmixing process as well as, reducing errors introduced by unwanted data or impurities in the spectral library.

1.5.4.2. Spectral Angle Mapper (SAM)

The Spectral Angle Mapper is a method used to quantify the spectral similarity between a spectral signature of interest and the spectral signatures present in a hyperspectral scene. This method is used in the context of spectral variability to compare spectral signatures and assess how similar they are in terms of spectral angle which is calculated by measuring the angle between the vectors of two spectral signatures in a vector space. The smaller the angle, the more similar the spectral signatures are. By calculating the spectral angle between the spectral signatures, it is possible to identify which endmembers are most similar and therefore have reduced spectral variability.

1.5.5. Deep Generative Models

Deep Generative Models are machine learning methods for modeling and generating complex data in which, they are used in the context of spectral variability to capture the distribution of spectral signatures present in hyperspectral data and based on deep neural networks, such as generative adversarial neural networks (GAN) or variational autoencoders (VAE), these models are able to learn latent representations of hyperspectral data, allowing them to generate new spectral signatures that are consistent with the original data distribution.

In the context of spectral variability, deep generative models can be used to reduce variability by learning latent representations that capture the subtle variations and underlying structures of spectral signatures and also, they can be used to generate synthetic spectral signatures that are more stable and less prone to unwanted variation.

CHAPTER TWO

DEEP LEARNING

Nowadays, deep learning market size has been estimated to be worth over $18.16 billion, growing at a rate of 41.7 percent from 2018 to 2023 since it has a lot of benefits from optimum data utilization, cost reduction, high quality results and reduced need for feature engineering, it has seen a rise in a variety of different application domains and begun to understood how it has the potential to revolutionize so many different research areas and parts of society ranging from advances in autonomous vehicles and robotics to medicine, biology and health care, reinforcement learning, generative modeling, security etc....

To illustrate more, deep learning is the use of artificial neural networks to model non-linear relationships between input data and predicted outputs, or labels which has a long-standing history cognitive science and neuroscience-inspired artificial intelligence. These neural networks develop the ability to extract pertinent, abstract information from the data used to train them. Advances in voice recognition, genomics, and computer vision have all been made possible by deep learning.

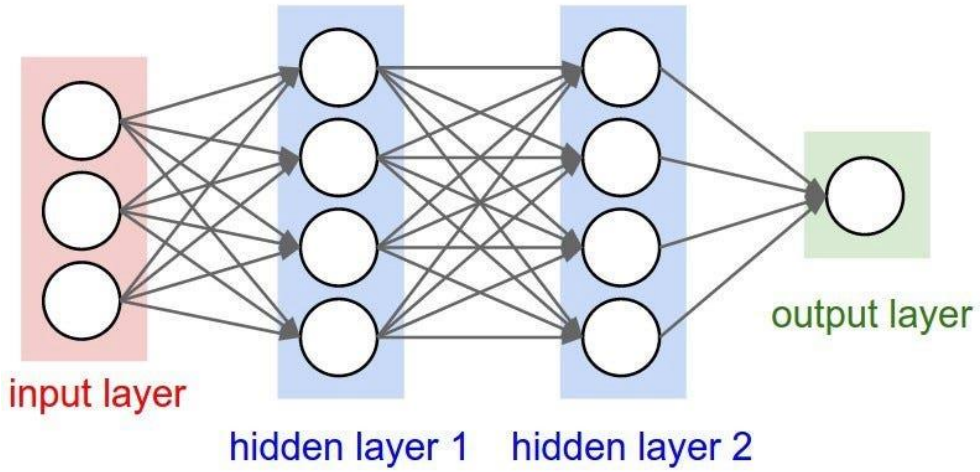


Figure 12: neural network architecture.

An equally vast spectrum of artificial neural network topologies has been designed to handle such a wide variety of jobs.

So, what is the neural network actually is? Neural network is a strategy for constructing a computer software that learns from data. It is based on the understanding of how the human brain operates. Initially, a group of software "neurons" is built and linked together, allowing them to communicate with one another. The network is then given a challenge to solve, which it repeatedly tries to accomplish while attempting to strengthen the connections that lead to success and weaken those that lead to failure.

Deep learning is a high-level abstraction algorithm that model the data from large sets of learned data. Additionally, it serves as an element of machine learning, which is a particular type of artificial intelligence. Artificial intelligence is a technology that allows a computer to emulate human behavior; machine learning, on the other hand, is a technique that accomplishes artificial intelligence using algorithms trained on data. In the context of deep learning, this structure is known as an artificial neural network.

Neural network topologies for deep learning have several hidden layers that can be anywhere between 2-3 layers up to 150 layers. This is the origin of the term "deep" learning. Large datasets of (labeled) data are typically utilized to train neural network designs and deep learning models. As a result, instead of depending on manual feature extraction, the models may learn about and adjust to the characteristics directly from the dataset.

There are three of the most historically significant architectures for Computer vision which are Feedforward Neural Networks (FNNs), Fully Connected Networks (FCNs), Convolutional Neural Networks (CNNs) and Transformers.

2.1. Feedforward Neural Networks

A Feedforward Neural Network is a fundamental architecture in the field of deep learning which is widely used for various tasks such as classification, regression and feature extraction.

In a forward propagating neural network, information flows in a single direction from the input layer through one or more hidden layers to the output layer where each layer is composed of several interconnected neurons, called neurons or units and the neurons in neighboring layers are completely linked, which means that each neuron's output is coupled to every neuron in the next layer, in the one hand, the input layer of

a forward propagating neural network receives the input data, which can be a vector or a matrix depending on the nature of the problem, then the hidden layers process the input data by applying nonlinear transformations to create increasingly abstract representations, in which, these transformations are performed using activation functions such as the sigmoid function or the rectified linear unit function (ReLU), in the other hand, the output layer produces the final prediction or output based on the representations learned from the hidden layers. The output activation function's layer is chosen according to the task at hand.

To train a forward propagating neural network, a process called gradient backpropagation is typically used which involves propagating the error signal from the output layer to the input layer and adjusting the weights of the network to minimize the difference between the predicted output and the actual output and this is accomplished using an optimization algorithm such as gradient descent, which iteratively updates the network weights.

2.2. Fully Connected Networks (FCNs)

FCNs are the simplest type of neural network architecture. Each input element in a fully connected layer is combined linearly to generate each output element, which is then subjected to a non-linear activation function.

One of the main type in FCN for pixel classification is the Autoencoder (AE), in which the idea of it, is to build some encoding of the input and try to reconstruct an input directly, the process is to take in as input raw data then pass it through some series of deep neural network layers and as a result the output is directly a low dimensional latent space (a feature space) , so in general "encoder" learn mapping from the data x to a low-dimensional latent space z.

2.3. Convolution Neural Networks (CNNs)

In this report, Convolutional Neural networks (CNN) will be focused on, which is an artificial neural network, also called a class of Deep Neural Networks, that is so far been most popularly used for analyzing images due to its remarkable performance in various visual recognition problems and also its ability to identify and categorize certain features from images, one of the founders to do so is (Makantasis et. al., 2015) who employed two convolution layers followed by two fully linked layers, and nowadays, it has applications in image and video recognition, image classification,

medical image analysis, computer vision, and natural language processing. Convolutional neural network is made up of convolutional layers that are alternately stacked and spatial pooling layers. The convolutional layer enables the feature maps extraction using linear convolutional filters, which will be followed by nonlinear activation functions such as rectifier, sigmoid and so on, it is also used to change the number of spectral channels and improving classification and also allowing gradient propagation over the network by employing residual connections. Spatial pooling is the process of grouping local characteristics from spatially neighboring pixels in order to increase object durability to minor deformations.

For the case of hyperspectral imaging, CNN is used mainly in HIS analyses, where a pixel and its neighbors in a hyperspectral image are fed into the CNN as inputs, and the final CNN output is set as the anticipated class labels.

2.4. Transformers

The Transformer architecture is a neural network architecture used in deep learning, particularly in the field of natural language processing. It was introduced by (Vaswani et al., 2017) and has seen great success in tasks such as machine translation and text generation.

The Transformer architecture differs from typical recurrent neural network architectures by using only attention mechanisms to capture the relationships between elements in an input sequence, however, it does not rely on recurrence or convolution to process sequences, which gives it increased computational efficiency and facilitates parallelism.

The Transformer architecture is composed of several blocks of attention layers and fully connected layers, also called encoder and decoder blocks in which, each encoder and decoder block consists of several sublayers.

The sublayer of attention is the central part of the Transformer in which it allows the model to distinguish long-range dependencies between the elements of the sequence by calculating attention weights between them which is used then to weight the importance of each element when generating output representations.

In each of these sublayers, there are several attention heads in which they enable calculating them independently of the other, allowing the model to capture different types of information and to be used to generate the output representations. hips in the

sequence. The attention weights are then combined to obtain an overall representation of the sequence.

In the context of hyperspectral imaging, the Transformer architecture can be used to improve the spectral resolution of images while reducing spectral variability and it can be achieved by exploiting the complex and non-linear relationships between different wavelengths in hyperspectral data.

2.5. Convolutional Neural Network's structure:

Because the collected hyperspectral data set contains tens of thousands of reasonably big data sets, deep learning facilitated the analysis of large amounts of data. For this case, CNN was used which is capable of extracting features from input data automatically by detecting and retrieving as well as learning specific spectral structures such as light reflection intensity, slope, and peak. Although noise has an influence on CNNs, the effect becomes minor as the amount of data grows.

To illustrate more, CNNs are neural networks that learn by repeating operations and highlighting the features of the input data where, on the input hyperspectral data, a convolution operation was performed, as illustrated in Figure 13. Equation (10) represents the convolved output data here, where x is the image, m*n is the kernel size, w is weight, and b is for bias parameter.

$$a_{ij} = \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} w_{st} x_{(i+s)(j+t)} + b \qquad \text{Equation 10}$$

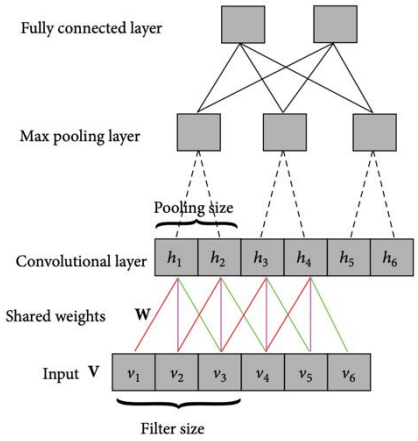


Figure 13 : a typical CNN design made up of a fully connected layer, a convolutional layer, and a max pooling layer
Source: Prasad S., (2020)

Convolutional Neural Networks are feed-forward neural networks made up of different combinations of convolutional layers, max pooling layers, and fully connected layers.

A CNN is composed of several layers that work sequentially to extract meaningful features from the data by applying convolution, pooling and classification operations.

- A convolution tool that separates and classifies numerous properties of an image for further analysis using a process known as feature extraction. This CNN feature extraction approach aims to reduce the number of features in a dataset. It creates new features by combining an initial collection of existing features into a single new feature. Not to forget that there are several CNN levels, as seen in the CNN architectural diagram below.
- In the feature extraction network, there are several pairs of convolutional or pooling layers.
- A fully connected layer that uses the output of the convolutional process to identify the class of the picture using previously extracted features.

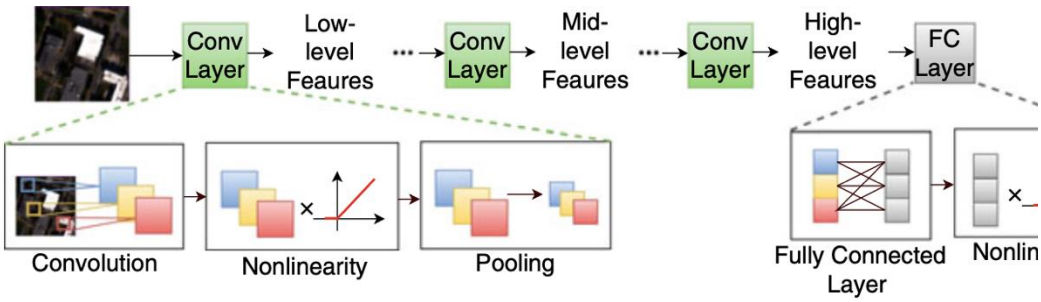


Figure 14: CNN architectural diagram.
Source: Prasad S., (2020)

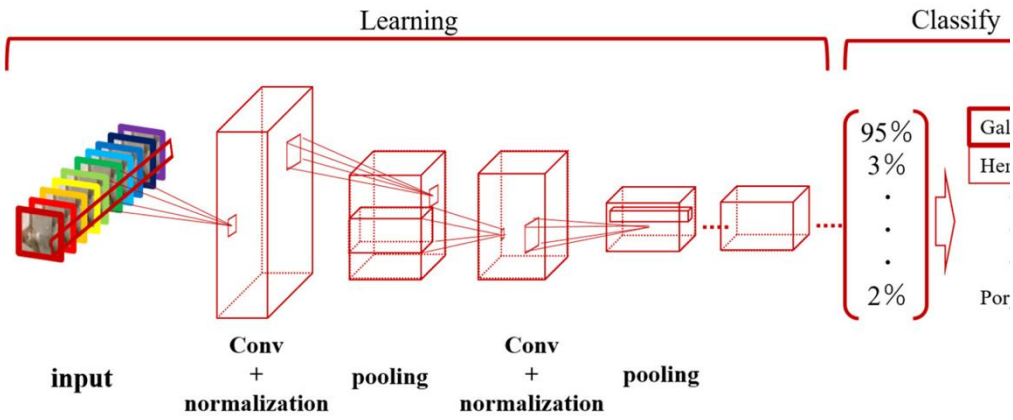


Figure 15: CNN structure for Hyperspectral data.

2.6. Building blocks of CNN

CNNs are multi-layer architectures that can be trained and consist of numerous feature extraction phases. Each level is made up of multiple layers such as dense layers, convolutional layers, pooling layers, recurrent layers, normalization layers, etc. Each layer may perform different kinds of transformations on their inputs and some layers are better suited for certain tasks than others, for example, a convolutional layer would likely be used in a model that has related work with image data, a recurrent in the other hand, will be used that's doing work with time series data and the dense layer or fully connected layer is just a layer that connects each input to each output within its layer, in the case of hyperspectral image four main layers will be discussed which are:

2.6.1. Convolutional layer

A CNN layer uses several convolutions in parallel to generate a collection of linear activation functions. Convolutional layers are in charge of extracting local features at various places employing trainable Kernels $W_{ij}^{(l)}$ that serve as link weights between the i,j feature maps of layer L-1 and layer L respectively, which is effectively a sliding dot product, in this scenario, the kernel moves along the input matrix, and the dot product between the two is taken as if it were a vector.

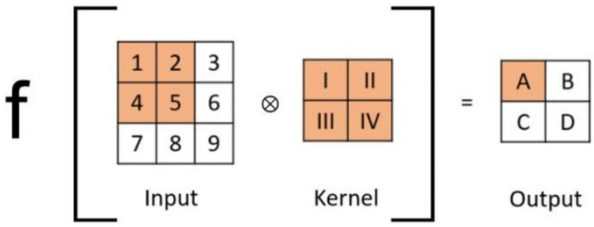


Figure 16: An illustration of how a convolutional layer works.
Source: Prasad S., (2020)

The convolution may be thought of as the multiplication of the pixels of the image I by components of a sliding window represented by the kernel, which is nothing more than a filter or a feature detector used to extract features from images. In other

words, the kernel is a matrix that traverses across the input data, performs the dot product on the sub-region of the input data, and returns the matrix of dot products as the output.

2.6.2. ReLU layer:

The Rectified Linear Unit (ReLU) is an artificial neural network's activation function which can be defined as follows:

$$f(x) = max(0, x) \quad \text{Equation 11}$$

Where, x is the input value, in other words, if the input value is positive or zero, the output is equal to the input value. If the input value is negative, the output is 0.

The ReLU is widely used in neural networks because of its computational simplicity and efficiency. It allows to introduce nonlinearity in the model, which is essential to learn complex and discriminative representations of data. In addition, ReLU promotes sparsity by making activations zero for negative values, which can lead to better computational efficiency, reduced information redundancy as well as it can be a key role in discriminative feature learning and contributes to improved model performance in tasks such as classification, object detection and image segmentation.

2.6.3. Pooling layer

A pooling function decreases a feature map's dimensionality and is repeated to each data channel (or band) to minimize susceptibility to rotation, translation, and scaling as well as aggregating data both inside and across the feature maps.

To illustrate more, pooling layers enables down sampling feature maps by summarizing the existence of features in feature map patches where the most active presence of a feature is summarized by max pooling, whereas average pooling focuses on the average presence of a feature.

For max pooling function, its purpose is to retain texture information by applying a window function to the input patch and computes the maximum within the neighborhood. In the other hand, average pooling function purpose is to retain background information by computing the mean of the input components inside a patch.

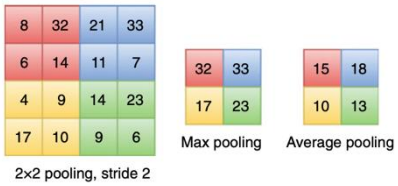


Figure 17: Pooling operations of several types.
Source: Prasad S., (2020)

The maximum is located via the max pooling technique within the neighborhood proximity window patches of size 2×2 with stride of 2. The average pooling method computes the average of the input components in a window patch of size 2×2 with a stride of 2, and by stride, it means the amount of movement between applications of the filter to the input image and it is almost always symmetrical in height and width dimensions.

2.6.4. Fully connected layer

Fully connected ANN layers are often utilized in the latter stages of a CNN for classification or regression based on feature maps acquired by convolutional filters where each node on neural network is fully connected to every previous and next node (Basha et. al., 2020).

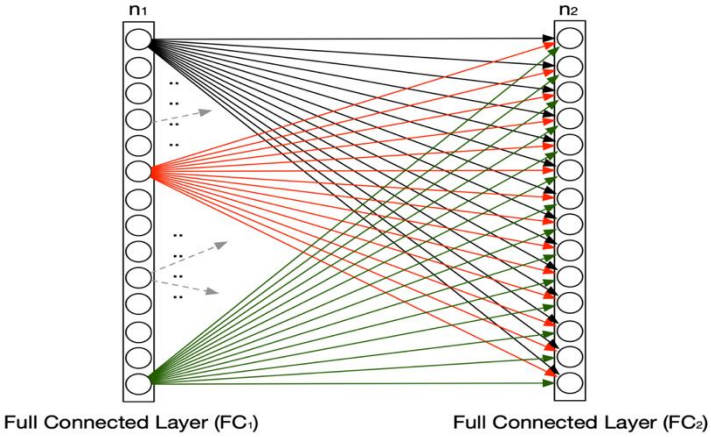


Figure 18: Fully connected layers with two layers.
Source: Ma W., (2017)

In other words, after several layers of convolution, pooling and ReLU, the extracted features are flattened and passed to one or more fully connected layers. These layers are similar to those of a classical neural network and are used to combine the extracted features and perform the final classification.

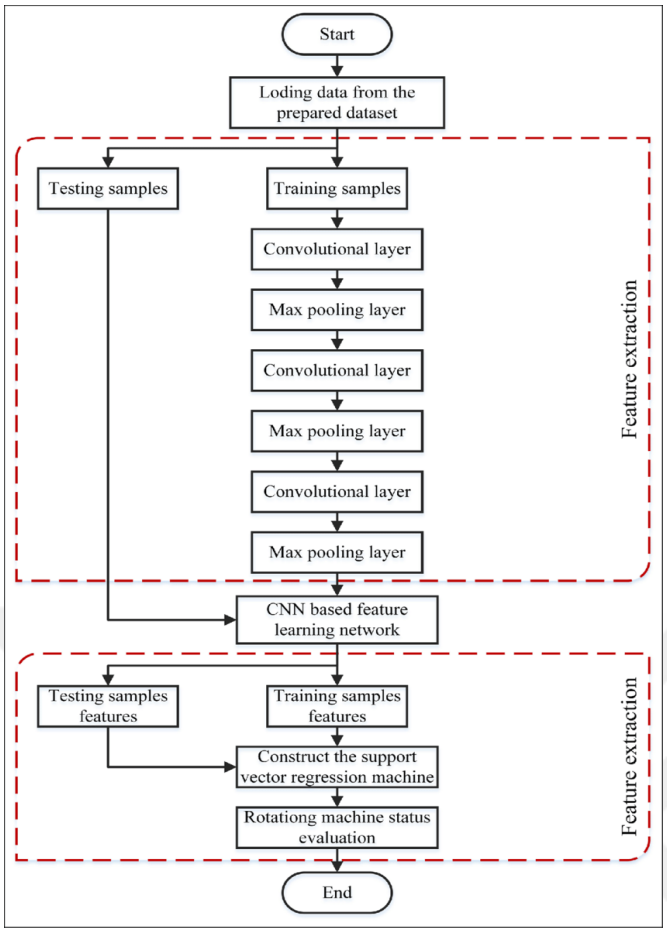


Figure 19 : CNNs algorithm flow chart.
Source: You W., (2017)

2.7. CNN and HSI

Convolutional neural networks (CNNs) are neural network architectures specifically designed for image processing and analysis. In the field of hyperspectral imaging, CNNs play a crucial role in the exploitation of spectral and spatial information contained in hyperspectral images, in which they are deployed to accomplish a variety of specific objectives such as:

2.7.1. Spectral classification:

CNNs are used in hyperspectral imaging to accomplish automated categorization of pixels or areas by learn to extract discriminative spectral characteristics and apply them to pixels or areas to assign class labels.

2.7.2. Abnormality Detection

CNNs are used to detect abnormal or unusual features in a hyperspectral image by learning to distinguish pixels or areas which are significantly different from the normal distribution of the data.

2.7.3. Spectral reconstruction

CNNs are deployed to reconstruct complete hyperspectral spectra from sample observations or compressed data which is done by learning to estimate missing spectral values or to reconstruct high resolution spectra from low resolution spectra.

2.7.4. Object segmentation

CNNs are employed in hyperspectral image segmentation to automatically detect and delineate objects or regions of interest within the image which is done by leveraging their ability to learn and recognize relevant features, CNNs can effectively identify and classify different objects and areas in the hyperspectral image.

A convolutional neural network is composed of several layers, including convolution layers, pooling layers and fully connected layers where, the convolution layer is the key component of CNNs in hyperspectral imaging which applies a set of convolution filters to the hyperspectral image to extract relevant features at different scales and abstraction levels that leads in capturing patterns and structures present in hyperspectral data.

Three groupings may be made about CNN:

(i) 1-dimension CNNs spectral features extraction (Huang et. al., 2015).

(ii) 2-dimension CNNs spectral features extraction (Zhao et. al., 2016).

(iii) 3-dimension CNNs spectral features extraction (Li et. al., 2017).

Hyperspectral image can be depicted as a three-dimensional data cube when spectral sampling is done along the z axis, 3D CNN used to learn nearby input signal in the HSI data's spatial and spectral dimensions, as well as to extract critical classification information.

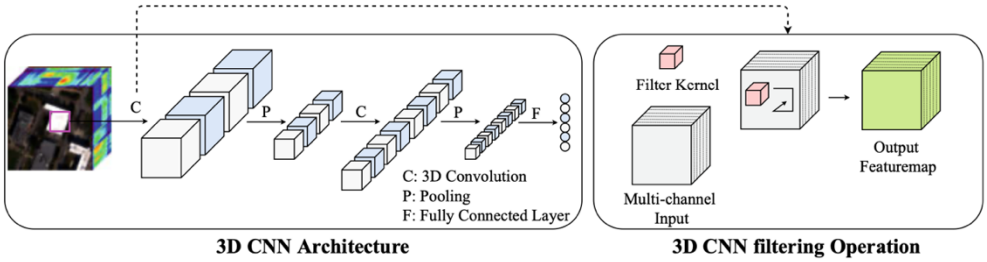


Figure 20 : A 3D CNN architecture for hyperspectral image.
Source: Prasad S., (2020)

2.8. Deep Generative Models

Generative models are a powerful concept where the goal is taking the trained samples from some distributions as an input and learn the represented model, in other words, all of them are seeking to learn the real data distribution of the training set in order to produce new data points with slight deviations by simulating the data's probability distribution and produce samples in accordance with it.

To Illustrate more, theoretically, based on a collection of data (xi, i)= 1,.....,Nx , generative models seek to estimate the probability distribution p(X) of a random variable X∈ RL in such a way that it is possible to generate new samples that resemble new realizations of X where, through the utilization of generative models, it becomes possible to comprehend the fundamental connections and structures within hyperspectral data. Consequently, this enables the generation of fresh observations that possess comparable attributes to the initial dataset. This capability finds relevance in diverse scenarios, including the simulation of hyperspectral data to facilitate model training, the completion of missing data, the amplification of data for enhanced learning, and even the creation of authentic hyperspectral scenes.

Generative modeling is based on two main ways and the purpose of them is fundamentally similar by trying to learn a probability distribution using the given model, furthermore, by trying to match that probability distribution as same as the true distribution of the data, first is the density estimation where data samples are given in which they going to fall to some probability distribution and the goal is to learn an approximation of what the function of that probability distribution could be an the second way is sample generation where, given some input samples (data distribution), the goal here is to learn a model of that data distribution and then use that process to actually generate new instances, or in other words new samples, in which it will fall in line with what the true data distribution is.

Generative data can be very complex and interesting due to frequently working with some data types such as images where the distribution is very high dimensional, for this reason a task called neural networks is used that can be very important in learning these extraordinarily complex functional mappings and estimating of these high dimensional data.

Generative models can be very useful of uncovering underlying features in a dataset in a completely unsupervised manner and this can be very important in applications that requires understanding more about what data distribution look like in a setting where the given model is being applied for some downstream task, for example, in facial detection, many different faces could be given as a data set so when starting, it could be very hard to know the exact distribution of these faces with respect to features such as skin tone, hair or illumination, for this case, generative modeling can be used to not only uncover what the distribution of these underlying features may be but also use this information to build more fair and representative data sets that can be used to train machine learning models that are unbiased and equitable with respect to these different underlying features.

Another example in generative model application is the outlier detection (autonomous driving case) and the goal is to detect rare events that may not be very well presented in the data but very important for the given model to be able to handle and effectively deal with when deployed, in this case generative model can be used to estimate this probability distribution and identify those instances such as when pedestrian walks in or there is a strange events like a deer crossing the road and be able to effectively handle and deal with this outliers in the data.

Various types of generative models exist, including the latent variable models which are generative neural networks (GANs), variational autoencoders (VAEs), and inverse adversarial generative networks (IRGANs). Each of these models employs distinct techniques for modeling and generating data.

Before exploring these neural networks, a common underlying variable deployed in these models called latent variable which is a lower-dimensional space where information is condensed in a compressed form, it acts as a hidden variable that captures the fundamental features of the input data and this latent variable can be seen as an abstract and condensed representation of information, and it is essential in producing new data samples that are realistic and coherent leading, by manipulating the values of the latent variable, it enables to exert control over the properties and attributes of the generated samples, leading to customization and exploration of the model's output.

2.8.1. Autoencoders

The idea of Autoencoders is to build some encoding of the input and try to reconstruct an input directly, the process is to take in as input raw data then pass it through some series of deep neural network layers and as a result the output is directly a low dimensional latent space (a feature space) , so in general "encoder" learn mapping from the data x to a low-dimensional latent space z (Alexsander et. al., 2023).

this low-dimensional z enables to effectively build a compression of the data by moving from the high dimensional input space to this lower dimensional latent space, thus, it enables to get a very compact and hopefully meaningful representation of the input data and the way to do this is to use the input data maximally by complementing this encoding with a decoder network as shown in the figure (20) , after that, the latent representation that lower dimensional set of variables will be taken and goes up from to try to learn a reconstruction of the original input image, thus a reconstructed output which is an imperfect reconstruction of the original data called (x-hat) will be observed, therefor, this network can be trained end-to-end by looking at the reconstructed output and the input and simply trying to minimize the distance between them, to illustrate more, the output and Input will be subtracted and then squared, and this is called a mean squared error between the input and the reconstructed output.

$$\mathcal{L}(x, \hat{x}) = \| x - \hat{x} \|^2 \qquad \text{Equation 12}$$

In the case of images, this is just a pixel by pixel difference between the reconstructed output and the original input.

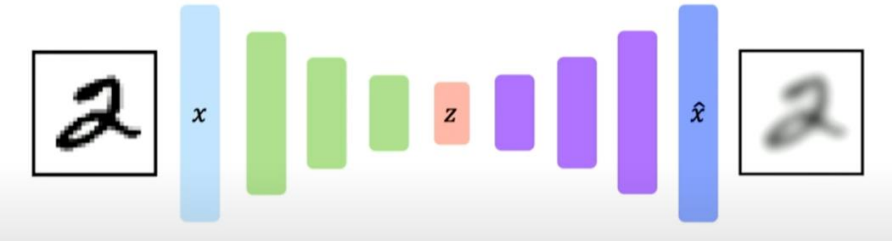


Figure 21 : encoder and decoder of the autoencoder model.
Source: Alexander A., (MIT 6. S191)

so, in general a "decoder" learns mapping back from latent space z to a reconstructed observation $\hat{x}$.

The idea of using this reconstruction is very powerful in taking a step towards the unsupervised learning idea by effectively trying to capture these variables which could be very interesting without requiring any labels to the given data and this is because of the fact of lowering dimensionality of the given data into this compressed latent space, the degree in which the compression is performed has a really big effect on the reconstruction quality, so this idea of autoencoders is a powerful first step in trying to learn a compressed representation of the given input data without any sort of label (unsupervised learning) from the start, in this way an automatic encoding of the data will be built, as well as self-encoding the input data which is why the term of autoencoders comes in the first place.

Autoencoding = **Auto**matically **encoding** data, "Auto" = **self**-encoding.

2.8.2. Variational Autoencoders (VAEs):

The concept of variational autoencoder is more commonly used in actual generative modeling of today, the difference between the traditional autoencoder and VAE is in the latent variable which means that for the traditional autoencoder, given some input x, when it is passed through after training, this will always result in the same output no matter how many times the input gets passed.

However, in the case of variational autoencoder and more generally in requiring to learn better and smoother representation of the input data and actually generating new images that weren't be able to generate before with autoencoder structure due to its purely deterministic, VAEs introduce an element of stochasticity of randomness to try to be able to generate new images and also learn smoother and more complete representations of the latent space and to do so, the latent space z will be breaking down into a mean and a standard deviation and the purpose of the encoder portion of the network is to output a mean vector and a standard deviation vector in which they correspond to distributions of these latent variables z, thus it could be said that variational autoencoders introduce some element of probability or randomness that will enable to generate new data also to build up a more meaningful and more informative latent space.

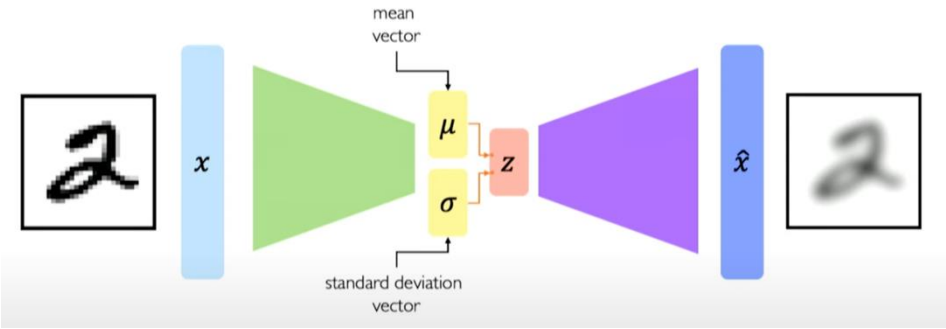


Figure 22: the additional elements in the VAEs compared to the AEs models.
Source: Alexander A., (MIT 6. S191)

The key here is that by introducing this notion of probability distribution for each of these latent variables, each latent variable will be defined by a mean and a standard deviation, and thus, new data will be generated by sampling from that latent distribution in which, because of it, both of the encoder and decoder architectures are going to be fundamentally probabilistic in their nature and that means is that over the course of training, the encoder is trying to infer a probability distribution of the latent space with respect to its input data while the decoder is trying to infer a new probability distribution over the input space given that same latent distribution and so when these networks are trained, two separate sets of weights (theta and phi) will be learned.

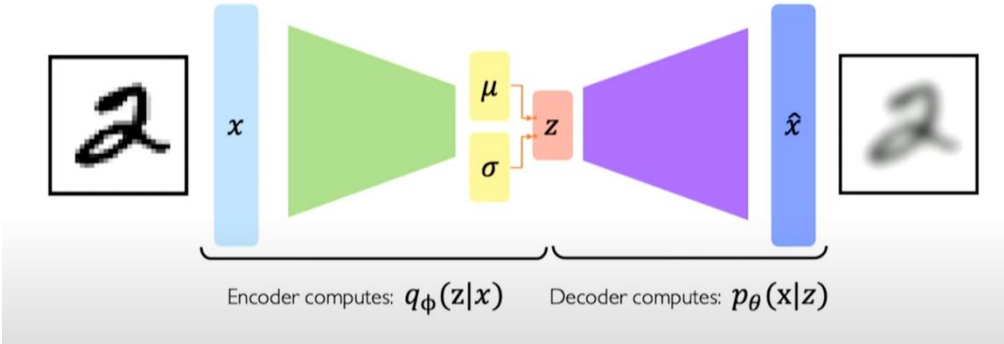


Figure 23 : An overview of variational autoencoder's key elements.
Source: Alexander A., (MIT 6. S191)

The loss function is now going to be a function of those weights and a new term will be introduced which is called the regularization term (the loss is no longer constituted by the reconstruction term) (Borsoi et.al., 2019).

$$\mathcal{L}(\phi, \theta, x) = (reconstructed\ loss) + (Regularization\ term) \quad \text{Equation 13}$$

$$\mathcal{L}(\phi, \theta, x) = E_{q_\phi(z|x)}[log\ p_\theta(x|z)] - D(q_\phi(z|x) \parallel p(z)) \quad \text{Equation 14}$$

Where:

- Reconstructed loss = $\parallel x - \hat{x} \parallel^2$ or the difference between the reconstructed output and the original input.
- The theory of this regularization term ‘‘D ($q_\phi(z|x) \parallel p(z)$)” is Kublac Libler divergence or KL divergence between the inferred latent distribution $q_\phi(z|x)$and the fixed prior on latent distribution p(z) that it’s going to impose some notion of structure in this probabilistic space in which:

$$D(q_\phi(z|x) \parallel p(z)) = -\frac{1}{2}\sum_{j=0}^{k-1}(\sigma_j + \mu_j^2 - 1 - log\sigma_j) \quad \text{Equation 15}$$

Properties to achieve from regularization:

a- Continuity: points that are close in latent space / similar content after decoding.

b- Completeness: sampling from latent space / “meaningful” content after decoding.

After defining the loss function, the objective is to optimize this loss by updating the weights of the network during the training process where, the weights are iteratively adjusted to minimize the loss. Breaking down the loss term, the reconstruction loss plays a crucial role and is similar to the loss used in autoencoder structures. In the case of image data, the reconstruction loss measures the pixel-wise difference between the original input and the reconstructed output. The goal is to minimize this difference, ensuring that the network can accurately reconstruct the input data by iteratively updating the weights based on this loss which enables the network to learns and improve its reconstruction capabilities over time.

It is important to note that when applying regularization techniques in a network, there is a trade-off between the degree of regularization and the quality of the reconstruction. Increasing the level of regularization can have a negative impact on the accuracy of the reconstructed data. Therefore, finding the right balance is crucial in practice. The goal is to achieve both a high-quality reconstruction and effective regularization that enforces a smooth and comprehensive latent space. This is typically

accomplished by applying regularization methods that encourage a more regular distribution of latent variables, such as imposing a normal-based regularization.

However, the only missing step back is that how could practice advance in training this network end-to-end, and notice that by introducing the mean and variance term and by imposing this probabilistic structure of the latent space, stochasticity was introduced which a sampling operation, operating over a probability distribution defined by this mean and variance terms (μ and σ) and what that means is that during back propagation, it is impossible to affectively back propagate gradients through this layer because it's stochastic, so, in order to solve this step back is to actually re-parametrize the sampling layer a little bit then diverting the stochasticity away from these μ and σ terms and then, ultimately be able train this network end-to-end.

To summarize VAE, the key problem of variational autoencoders is a concern of density estimation trying to estimate this probability distributions of these variables z.

2.8.3. Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN) is an artificial neural network architecture where Instead of explicitly modeling the probability density or distribution of the given data, the goal here is to care about this implicitly while using this information to sample and generate new instances and this artificial neural network involves two main components of two parts which are the generator and the discriminator in which, the generator is in charge of producing new synthetic data that looks similar real-world instances, such as images, text, or sounds in which the method employs a training data set to create samples that are similar to the distribution of real data. The discriminator, on the other hand, is trained to discriminate between actual and created data by learning from a training data set and classifying real data as "real" and generated data as "fake."

The step back here is that the given input data is very complex and it's very difficult to go with it to generate new realistic samples, thus, GANs are generative models that are created by competing two neural networks, in other words, by creating two individual neural networks that are affectively adversaries.

To illustrate more, to mislead the discriminator, the generator converts noise into an imitation of the data, while the discriminator attempts to distinguish actual data from fakes made by the generator, in other words, these components are set up to

compete with each other in such a way that the discriminator is forced to classify fake and real data and the generator is forced to produce better fake data to deceive the discriminator.

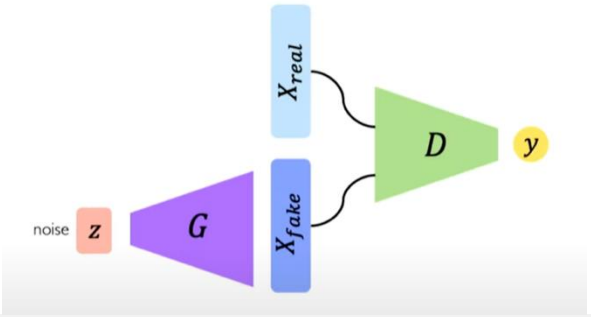


Figure 24 : An overview of GAN's elements.
Source: Alexander A., (MIT 6. S191)

This iterative process of interaction between the generator and the discriminator establishes a dynamic learning cycle, wherein both sides continuously refine their performance and the ultimate objective here is to achieve a generator that can produce synthetic data of exceptional quality, making it virtually indistinguishable from real data in the eyes of the discriminator.

The way GANs is trained is by formulating an objective or a loss function which known as adversarial objective and the goal is that the generator exactly reproduces the true data distribution.

2.8.3.1. Loss function:

The loss function is essential for directing training and maintaining that the generator and discriminator both learn effectively which is separated into two parts:

- the discriminator loss:

The discriminator plays a crucial role in GANs by accurately distinguishing between real and generated data. The discriminator loss quantifies the discriminator's performance in this task. It is determined by evaluating how well the discriminator correctly classifies real and generated samples. The discriminator strives to minimize this loss by improving its accuracy in distinguishing between the two types of data in which the goal is maximizing the objective D overall

$$Argmax_D\ E_{z,x}(logD\big(G(z)\big) + log(1 - D(x))) \qquad \text{Equation 16}$$

- the generator loss:

The primary goal of the generator in a GAN is to create synthetic data that exhibits a realistic appearance. In order to accomplish this, the generator seeks to decrease the probability of the discriminator accurately identifying the generated samples as artificial. The generator loss is computed by evaluating the discriminator's output when presented with the generated data. The objective is to motivate the generator to generate samples that have a higher probability of being classified as genuine by the discriminator.

For the generator, the goal is minimizing the objective G from the perspective of the generator

$$Argmin_G\ E_{z,x}(logD\big(G(z)\big) + log(1 - D(x))) \qquad \text{Equation 17}$$

So, in general the overall loss function is

$$Arg\ max_D\ min_G E_{z,x}(logD\big(G(z)\big) + log(1 - D(x))) \qquad \text{Equation 18}$$

CHAPTER THREE

METHODOLOGY

A wide range of applications, including agriculture, environmental monitoring, and remote sensing, benefit from hyperspectral imaging, however, the low spectral resolution and intrinsic spectrum fluctuation of hyperspectral data make correct analysis and interpretation difficult.

In this thesis, a novel methodology for improving the spectral resolution of hyperspectral data will be explored by deploying the models and methods explained in the previous chapters which mainly based on the combination between the concept of Linear Mixing Model and deep generative models where, in one hand, the linear mixing model is frequently employed to portray the spectral characteristics of pixels with mixed contents in hyperspectral images, resulting from the presence of multiple materials within a single pixel and in the other hand, deep generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) that have demonstrated impressive abilities in producing realistic synthetic data.

3.1. Experiment tools and set up

3.1.1. Matlab_R2022b

MATLAB R2022b, developed by MathWorks, is a software version that belongs to the ongoing release cycle of MATLAB in which, it aims to bring forth enhancements, bug fixes, and novel features as well as tools and libraries to improve the overall functionality and performance of the software.

3.1.2. Google Colab

Google Colab is a cloud-based collaborative notebook platform. It is developed by Google and allows users to write, run and share Python code. Colab provides an online integrated development environment (IDE), as well as free access to computing resources, including processors and GPUs. It also makes it easy to collaborate with other users in real time. Colab is popular among researchers, students, and developers because it provides a convenient solution for running and experimenting with code without having to set up your own development environment.

In this thesis, Google Colab has provided a cloud-based environment with pre-installed libraries and resources necessary for running Python code.

3.1.3. Python

Python is a versatile and widely used interpreted programming language. It was developed by Guido van Rossum and its design focuses on code readability and syntactic simplicity, making it a very accessible language, even for programming starters.

3.2. Simulation set up structure

In this thesis, a strategy is presented for increasing spectral resolution while decreasing spectral variability in hyperspectral images using a deep generative modeling approach. In hyperspectral data processing, the LMM is a well-established approach for spectral unmixing, thus, in this methodology, the LMM framework is deployed and expanded further to address the problems regarding spectral resolution enhancement and spectral variability reduction.

The objective of spectral unmixing in hyperspectral image analysis is to estimate the abundance fractions of pure spectral components known as endmembers, for this thesis, an approach known as the Generative Extended Linear Mixing Model (GELMM) is presented, which is a combination of the Extended Linear Mixing Model (ELMM) with deep generative modeling where the goal is to improve spectral resolution while decreasing spectral variability in hyperspectral pictures.

So, to begin with, the implemented code follows several steps to achieve the thesis methodology which are:

3.2.1. Data loading:

The code starts by importing the data necessary for the spectral unmixing implementation which are four synthetic Each of these synthetic data contains:

- The variable "M" in the MATLAB file represents the endmembers used in the data cube. In spectral unmixing, endmembers refer to the pure spectral signatures of materials found in the scene. In this context, "M" contains the spectral signatures of the endmembers that have been extracted from the USGS Spectral Library.

- Mvs: This variable corresponds to a modified version of the endmembers (M) in which they have been normalized. Normalization is the process of scaling the endmembers to have a unit-norm, meaning that their magnitudes are adjusted so that their Euclidean lengths become equal to 1. This normalization step is often beneficial in specific spectral unmixing algorithms as it helps to maintain consistent and reliable results.
- The variable 'alphas' in the MATLAB file represents the abundance maps or coefficients associated with the endmembers present in each pixel of the data cube. These maps indicate the relative proportions or quantities of each endmember within a pixel. The variable 'alphas' contains these abundance coefficients.
- The variable "alphas_cube" is a data structure in the shape of a cube, which contains the abundance maps for each pixel in the data cube in which, each entry in the "alphas_cube" corresponds to the abundance coefficients of the respective endmembers in that particular pixel.
- The variable "r" in the MATLAB file represents the correlation structure of the abundance maps. It signifies the spatial correlation or interdependence between adjacent pixels concerning their abundance values in which this correlation structure information will be stored in the variable "r".
- The variable "r_cube" in the MATLAB file is a structure resembling a cube which stores the correlation structure (r) for each pixel in the data cube where, each entry in the "r_cube" variable represents the correlation values of the abundance maps between a specific pixel and its neighboring pixels. Noting that the matrix “r_cube” represents the hyperspectral cube.

The hyperspectral data consists of a hyperspectral image displayed as a three-dimensional cube structure, with each pixel containing a spectral vector where its dimensions are the variables m, n and L that specify the width, height, and number of spectral bands, respectively and The variable P, which specifies the number of endmembers, denotes the size of the spectral vectors within each pixel.

3.2.2. Endmember extraction:

The next step in the implemented MATLAB code is the endmember extraction which is has an important role in hyperspectral analysis. In this thesis, in order to obtain the endmembers, a specialized algorithm called Vertex Component Analysis (VCA) has been employed. This algorithm is implemented through the VCA function located in the 'utils' directory, and it is specifically designed to cater to the task of endmember extraction. The main objective of this technique is to identify and separate pure spectral characteristics that are embedded in the hyperspectral data enabling these signatures to indicate several components or substances present in the hyperspectral sensor-captured image. By utilizing the VCA algorithm and its implementation, it is possible to identify and extract these unique spectral signatures, providing valuable insights into the composition of the hyperspectral data.

The endmember extraction algorithm follows a series of steps to identify the endmembers from the hyperspectral data.

After preprocessing the data by normalizing each spectral vector, the algorithm calculates the Spectral Angle Mapper (SAM) to determine the pixel with the minimum spectral angle as the initial endmember which is done by employing the equation:

$$SAM = acos((M(:,k).') * M0(:,l) /(norm(M(:,k)) * norm(M0(:,l))))$$

Equation 19

Where:

- acos () computes the inverse cosine (or arccosine) of the dot product divided by the magnitude product.
- M(:,k) is the k-th column of matrix M, which represents an identified endmember spectrum in the observed hyperspectral data.
- $M0(:,l)$ is the l-th column of matrix M0, and it represents a reference endmember $norm(M(:,k)$spectrum.
- computes the k-th endmember spectrum's Euclidean norm or magnitude.
- $norm(M0(:,l))$ computes the l-th reference endmember spectrum's Euclidean norm or magnitude.
- And $(M(:,k).') * M0(:,l)$ computes the dot product of the k-th endmember spectrum with the l-th reference endmember spectrum noting that the resultant value is converted from radiance to degree using the multiplication by $180/\pi$.

Then the endmember extraction algorithm employs orthogonal projection to determine the pixel with the maximum projection length as the next endmember, which then, these endmembers are updated through iterative selection based on maximizing spectral angles until the target number is reached. And finally, the resultant endmembers, which indicate the spectral signatures of different materials or substances in the hyperspectral data, are stored in the matrix M0 and it's done by preforming the command

$$\mathrm{M0} = \mathrm{vca}(\mathrm{r}, 'Endmembers', \mathrm{P})$$

By applying the VCA method, the objective is to identify the most unique and representative spectral signatures that characterize various materials or chemicals in the observed scene, where, for a variety of applications, including object identification, material categorization, and environmental monitoring, these extracted endmembers offer useful information.

Ultimately, by employing the VCA algorithm and its implementation, it becomes possible to effectively uncover and extract crucial spectral details as it is a vital tool to ensure the accuracy and dependability of the endmember extraction process, as it directly influences the reliability of subsequent analyses and results.

3.2.3. Spectral unmixing algorithms

Several spectral unmixing techniques are used in hyperspectral data analysis to determine the abundance percentages of various materials or substances present in the observed scene, where, these methods are used to extract relevant information from hyperspectral data that are used as follows:

3.2.3.1. Fully Constrained Least Square Unmixing algorithm

In this thesis the first deployed unmixing algorithm is the FCLSU which is a method for determining endmember abundance fractions in a hyperspectral image which is done by solving a linear least squares problem repeatedly while applying non-negativity requirements on abundance fractions.

The FCLSU algorithm is implemented within the FCLSU function with the two arguments which are the hyperspectral data 'x' and the reference endmember matrix

‘M0’ with the goal of estimation the abundance fractions of endmembers of hyperspectral image.

The next step, a small value Delta which is a regularization parameter is defined that serves in addressing numerical stability during abundance fraction’s estimation which helps in avoiding numerical errors that could occur in zero division or extremely small values and also, a matrix N of size $(l + 1) \times p$ is created which is deployed to generate a linear system of equations that is solved for the purpose to estimate the abundance fractions in hyperspectral data where

Delta is multiplied by the reference endmember matrix M0 to assign the first l rows of N, and the last row of N is set to a row of ones, moreover, to store the abundance fractions, an empty matrix, out, is created.

The next step in the FCLSU algorithm is abundance estimation, where the abundance fractions for each sample in the hyperspectral image are calculated which is done by using the MATLAB built-in function lsqnonneg that stands for “Least Square Non-Negative” to solve the constrained least squares problem, where the goal is to estimate the abundance fractions while upholding the non-negativity constraint and the resulting is stored in the empty matrix ‘out’ and after calling the FCLSU function, the estimated abundances are store in the variable A_FCLSU and transposed to match the desired output.

3.2.3.2. The Extended Linear Mixing Model & The Generalized Extended Linear Mixing Model (GELMM)

The next spectral unmixing algorithm is the Extended Linear Mixing Model and the Generalized Extended Linear Mixing Model which as explained in the previous chapter, their main role is to estimate the endmembers and abundance fractions in hyperspectral images, this ELMM algorithm is based on the principle of linear mixing, where each pixel's spectral signature in the hyperspectral image is modeled as a linear combination of endmembers, with associated abundance fractions.

The ELMM technique enables the characterization and interpretation of mixed pixels in hyperspectral data by accurately estimating endmembers and abundance fractions and in this thesis, the ELMM is deployed during the unsupervised spectral unmixing process, which comprises several steps. Initially, the hyperspectral image cube undergoes a preprocessing stage to eliminate noise and artifacts, although the

specific techniques employed in this code snippet are not explicitly mentioned. Following this, the initial endmembers are estimated either by selecting pixels or utilizing a spectral library. In this experimental code structure, the abundance cube (A_FCLSU_cube) is transformed into a matrix (A_init) using the hCubeToMatrix function to obtain the initial endmembers. Subsequently, the ELMM algorithm is executed, incorporating the ADMM optimization scheme, which refines the estimated endmembers by considering residuals which aims to improve the accuracy and convergence of the unmixing results.

Before executing the ELMM algorithm, it iteratively updates the abundance fractions (A_ELMM) and endmembers (psis_ELMM) while considering several parameters are set which are:

- the regularization parameters **lambda_s**, **lambda_a,** and **lambda_psi** which play a crucial role in achieving a trade-off between accurately representing the observed data and promoting sparsity in the abundance matrix and endmembers and by adjusting these parameters, the balance between smoothness and sparsity in the estimated endmembers and abundances can be controlled, thereby influencing the final unmixing results.
- The **nnorm** parameter specifies the regularization method used for the abundances, in this case, it is Total Variation (TV) norm with **1,1** as the input parameters.
- The **verbose** parameter is set to **true** to display the output while running the algorithm.
- the **maxiter_anls** parameter which specifies the maximum number of iterations for the ANLS (Alternating Non-negative Least Squares) phase in the ELMM algorithm for this case, the abundance fractions are repeatedly updated using ANLS.
- the **maxiter_admm** parameter which specifies the maximum number of iterations for the ADMM (Alternating Direction Method of Multipliers) step in the ELMM algorithm. ADMM is an optimization method for adjusting estimated endmembers and abundance fractions.
- the **epsilon_s, epsilon_a,** and **epsilon_psi** parameters, which determine the convergence requirements for the abundance fractions (s), abundances (a), and endmembers (psi). If the change in these variables between consecutive

iterations falls below the specified epsilon values, the algorithm is considered to have converged.

- The parameters **epsilon_admm_abs** and **epsilon_admm_rel** specify the convergence conditions for the ADMM step where the absolute convergence criteria is represented by epsilon_admm_abs, whereas the relative convergence criterion is represented by epsilon_admm_rel. The ADMM step is deemed convergent if the absolute and relative change in the ADMM variables are less than these criteria.

Overall, these parameters are essential for the ELMM method to achieve a balance between properly capturing observed data and promoting low density in the abundance matrix and endmembers.

Furthermore, by executing ELMM_ADMM function in which, within it, the algorithm iteratively updates the abundance fractions and endmembers, considering the regularization terms and constraints which helps to solve the optimization problem that involves alternating updates between the abundance fractions and the endmembers and finally, The ELMM algorithm is executed using the parameters set above and returns the abundance matrix **A_ELMM**, the endmember matrix **M0**, the unmixed image **S_ELMM**, and an empty **optim_struct** variable, in the other hand, The Generalized Extended Linear Mixing Model extends the capabilities of the ELMM algorithm by introducing more versatile constraints or priors on the abundance fractions in which, this expansion enables the inclusion of intricate relationships between the abundances and additional parameters, thereby improving the accuracy and adaptability of the unmixing procedure. With the GELMM, the unmixing process becomes more robust and flexible, accommodating a wider range of scenarios and offering enhanced capabilities for analyzing hyperspectral data.

the code sets the parameters for the GELMM algorithm by modifying the values of maxiter_anls, maxiter_admm, and the stopping criterion values as well as the regularization parameters are set differently. psis_init is initialized to ones with dimensions $L \times P \times N$, where L is the number of endmembers, P is the number of pixels, and N is the number of spectral bands. The GELMM algorithm is then executed using the modified parameters, and it returns the abundance matrix A_GELMM, the endmember matrix M0, the unmixed image S_GELMM, and an optim_struct variable

which represent the estimated abundance fractions, the endmembers, and the unmixed image obtained by applying the GELMM algorithm.

3.2.3.3. The Perturbed Linear Mixing Model

In this section of experimental code structure, the Perturbed Linear Mixing Model is deployed where the primary purpose is to generate synthetic hyperspectral images that closely resemble real-world scenarios by incorporating multiple factors such as endmember spectra, abundance maps, noise, perturbations, spectral response, and sensor-specific noise to accurately represent the complexities and characteristics of real-world scenarios.

The PLMM algorithm's code structure begins by initializing threshold values and the initial abundances and endmembers, after then, the algorithm calls the interface_PLMM function with the hyperspectral data cube, initial abundances, initial endmembers, and threshold values which serves as a bridge for the Perturbed Linear Mixing Model (PLMM) algorithm takes such as hyperspectral data, initial abundances, initial endmembers, and regularization parameters as input, the main role of this interface_PLMM function is estimate the abundance maps A_PLMM, endmember variation matrix dM_PLMM, and the estimated endmembers M_PLMM by preparing the data, configuring the required parameters, and proceeding to execute the PLMM algorithm which involves utilizing both Block Coordinate Descent (BCD) and Alternating Direction Method of Multipliers (ADMM) iterations which enables the estimation of abundance maps, endmembers, and perturbation matrices by iteratively optimizing the unmixing problem.

3.2.4. Deep generative modeling

In this thesis methodology, in order to increase the spectral resolution while decreasing spectral variability, a method is proposed which integrates the deep generative model with the linear mixing model methods which involves several steps. Initially, a deep generative model is trained using a large dataset of pure endmember spectra, allowing it to learn the statistical properties of the endmember spectra where, this trained model is then utilized to generate synthetic endmember spectra, that represent additional variations beyond what was present in the training dataset, for this case the variational autoencoder is employed for training the generated endmembers in which, they are subsequently integrated into the linear mixing model framework,

and expanding the set of endmembers used in the unmixing process which means that the set of endmembers used in the linear mixing model now includes both the original endmembers and the generated endmembers.

The goal of this integration is to introduce new spectral variations to the endmember set that may be present in real-world scenarios but aren't well represented by the original endmembers, thus, in this thesis, a spectral unmixing (SU) formulation is proposed that combines the benefits of deep learning methods with the physical mixing process, while using limited training data by utilizing a deep generative neural network to represent the manifold of endmember spectra and incorporate it within the linear mixing model (LMM).

The unmixing procedure is carried out in this section of the experimental code structure by calling the DeepGUn function, which employs a deep generative model for hyperspectral unmixing where various input parameters are required, including the hyperspectral data cube (r_cube), an initial abundance matrix (A_init), and an initial estimate of the endmember matrix (M0). Next, it trains a deep generative model using the hyperspectral data cube (r_cube) to capture the statistical properties and variability of the endmember spectra. The unmixing process utilizes the Linear Mixing Model (LMM) and an optimization algorithm (such as alternating least-squares) to estimate the abundances and determine the optimal combination of endmembers for explaining the observed spectra. To improve the estimation of abundances and account for endmember variability beyond the training dataset, the extended LMM incorporates the parameters obtained from the deep generative model training, including the generative endmember models. This integration exploits the deep generative model within the LMM framework. Finally, the algorithm provides the estimated abundance matrix (A_deepGen) and the estimated endmember matrix (M_DeepGen) as outputs, which represent the proportions of endmembers and the spectral signatures of materials present in the scene.

3.3. Experimental results

In this section, the performance of the proposed DeepGUn method is evaluated using synthetic data which are generated with different levels of noise and variability to evaluate the robustness and accuracy of the methods and this is done through simulation results. A comparison is made

with other methods including the fully constrained least squares (FCLS), the PLMM, the ELMM, and the GLMM which is made by evaluating the results using several metrics, such as the Normalized Root Mean Squared Error (NRMSE) between the estimated abundance maps ($NRMSE_A$), and also between endmember matrices ($NRMSE_M$), and between the reconstructed images ($NRMSE_Y$), as well as the Spectral Angle Mapper (SAM) which is employed to assess the estimated endmembers where:

- The Normalized Root Mean Square Error (NRMSE) quantifies the difference between a given tensor χ and its estimated equivalent $\hat{\chi}$ is defined by *(Drumetz et.al., 2016) as* as:

$$NRMSE_{\chi} = \sqrt{\frac{\|\chi - \hat{\chi}\|_F^2}{\|\chi\|_F^2}} \qquad \text{Equation 20}$$

And the Spectral Angle Mapper is defined by (Borsoi et.al., 2018) as:

$$SAM_{\chi} = \frac{1}{N} \sum_{n-1}^{N} \sum_{p-1}^{P} arccos\left(\frac{m_{p,n}^{\intercal} \hat{m}_{p,n}}{\|m_{p,n}^{\intercal}\| \|\hat{m}_{p,n}\|}\right) \qquad \text{Equation 21}$$

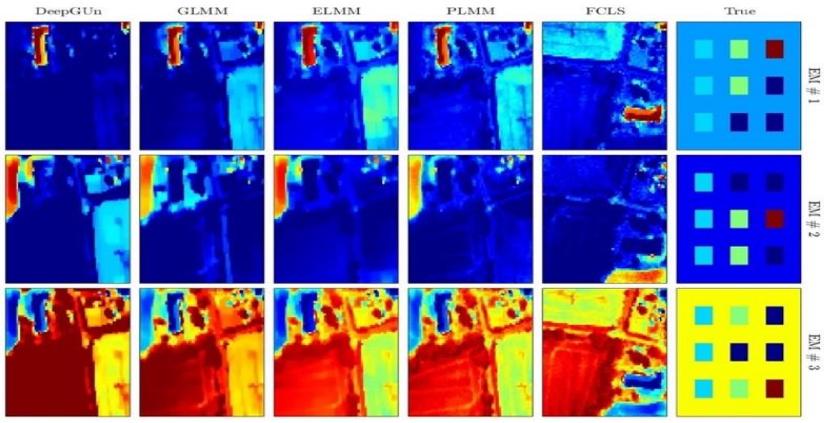


Figure 25: The tested algorithms displayed abundance maps for data cube 1, with colors representing abundance values in which the color spectrum ranged from blue (indicating an abundance value of 0) to red (indicating an abundance value of 1).

Data Cube 1					
	$NRMSE_A$	**$NRMSE_M$**	**SAM_M**	**$NRMSE_Y$**	**TIME (s)**
FCLSU	0.2854	--	--	0.0350	0.42
PLMM	0.2604	0.1075	0.0440	**0.0007**	96.55
ELMM	0.2554	**0.1032**	0.0398	0.0321	31.18
GLMM	0.2480	0.1036	0.0355	0.0235	25.23
DeepGUn	**0.1464**	0.1075	**0.0231**	0.0459	147.23

Table 2: simulation results for data cube 1.

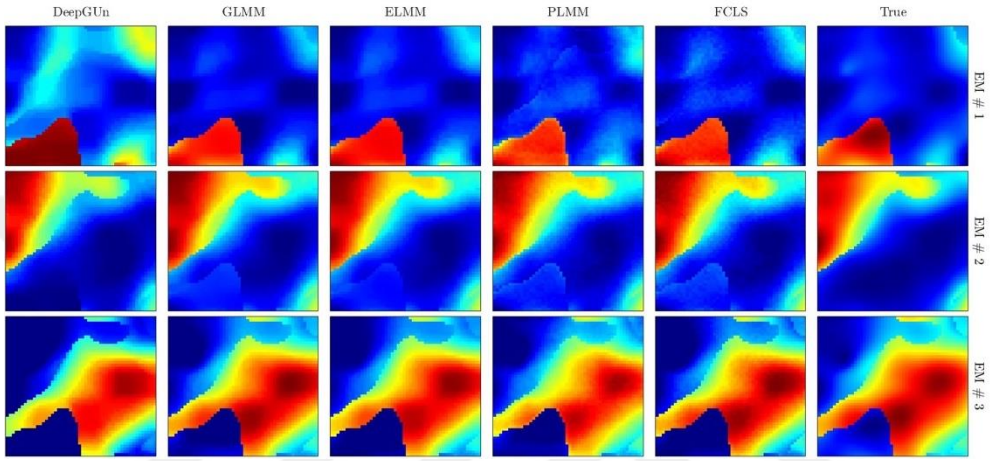


Figure 26: The tested algorithm as displayed abundance maps for data cube 2.

Data Cube 2					
	$NRMSE_A$	**$NRMSE_M$**	**SAM_M**	**$NRMSE_Y$**	**TIME (s)**
FCLSU	0.1294	--	--	0.0393	0.21
PLMM	0.1287	0.0600	0.0441	**0.0007**	32.74
ELMM	0.1189	0.0620	0.0380	0.0325	16.48
GLMM	**0.1125**	**0.0531**	0.0369	0.0228	21.10
DeepGUn	0.2605	0.0928	**0.0273**	0.0551	75.41

Table 3: simulation results for data cube 2.

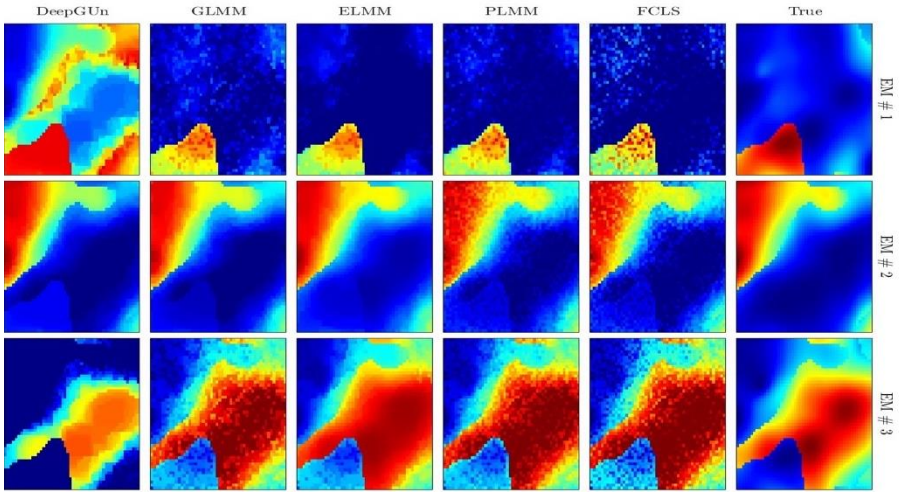


Figure 27: The tested algorithms displayed abundance maps for data cube 3.

Data Cube 3					
	NRMSE$_A$	**NRMSE$_M$**	**SAM$_M$**	**NRMSE$_Y$**	**TIME (s)**
FCLSU	0.2606	--	--	0.0542	0.18
PLMM	0.2379	0.0784	0.0329	**0.0006**	36.39
ELMM	0.2307	0.0716	**0.0170**	0.0315	20.95
GLMM	**0.1841**	**0.0638**	0.0185	0.0226	24.94
DeepGUn	0.5213	0.1796	0.0202	0.1086	97.71

Table 4: simulation results for data cube 3.

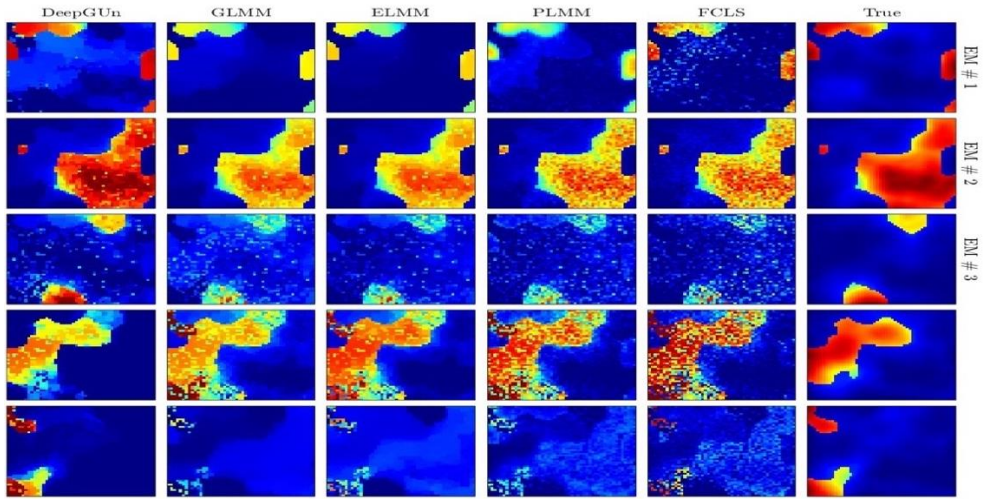


Figure 28 : The tested algorithms displayed abundance maps for data cube 4.

Data Cube 4					
	$NRMSE_A$	**$NRMSE_M$**	**SAM_M**	**$NRMSE_Y$**	**TIME (s)**
FCLSU	0.5109	--	--	0.1712	0.25
PLMM	0.5187	0.5531	0.3304	**0.0020**	228.72
ELMM	0.4551	0.4721	0.1311	0.1007	25.17
GLMM	0.5029	0.4890	0.1676	0.0827	38.33
DeepGUn	**0.2107**	**0.3331**	**0.1092**	0.2114	99.49

Table 5: simulation results for data cube 4.

Noting that the $NRMSE_M$ and SAM_M evaluate the precision of endmember reconstruction and material identification, respectively.

The experimental results show that the DeepGUn method outperforms the other methods in terms of accuracy and robustness for the data cube 4 whereas observed in the table, it consistently achieves lower NRMSE values for abundance maps, endmember matrices as well as it outperforms the competing algorithms in terms of $NRMSE_A$ for data cube 1 which indicates that the DeepGUn achieves higher accuracy in estimating the proportions of materials in the scenes compared to other methods.

DeepGUn also demonstrates the best performance in terms of SAM_M for data cube 4 and 2, indicating its ability to accurately reconstruct endmembers and identify materials from the observed hyperspectral scenes, moreover, it is noticed that the DeepGUn algorithm's NRMSEY, which measures reconstruction error, was similar to the FCLS technique but much higher than the GLMM method which is perfectly expected since the GLMM has more degrees of freedom which indicates that the GLMM model has greater complexity compared to other algorithms.

For the execution periods of the algorithms, it is noticed that the computational complexity of DeepGUn is longer then the GLMM but shorter from the PLMM which means that DeepGUn requires more computational resources and time compared to the GLMM but less computational resources and time compared to the PLMM, in other words, in terms of computing complexity, a compromise between the two methods is achieved, delivering better efficiency compared to the PLMM while still being more demanding than the GLMM and that implies that DeepGUn finds a compromise between computational efficiency and performance.

In summary, The DeepGen algorithm consistently outperforms FCLS, PLMM, ELMM, and GLMM in terms of RMSE for abundances across all four data cubes in which, it achieves comparable or slightly worse results in terms of RMSE for

endmembers, but it demonstrates superior accuracy in material identification, as indicated by lower SAM values for endmembers compared to the other algorithms. Although DeepGen has higher RMSE for R compared to PLMM and GLMM, it remains comparable to FCLS and ELMM. Additionally, DeepGen strikes a balance between computational efficiency and performance, with execution times falling between those of GLMM and PLMM for all data cubes, and overall, these results highlight the effectiveness of the DeepGen algorithm in spectral unmixing tasks, showcasing its superior performance in estimating abundances and accurate material identification while maintaining a reasonable computational complexity.

CONCLUSION AND FUTURE WORKS

In this thesis, the application of the DeepGen algorithm for increasing the spectral resolution of hyperspectral images while decreasing spectral variability using the Linear Mixing Model (LMM) is focused on.

Beyond the training dataset, the system used a deep generative model to capture the statistical features and variability of endmember spectra in which it proceeded with initial abundance and endmember matrices, then moved on to deep generative model training with the hyperspectral data cube, further, the LMM framework and an optimization algorithm were used in the unmixing process to estimate abundances and select the optimal combination of endmembers, where in the meantime, the DeepGen technique combined the parameters gained from deep generative model training, improving abundance estimation and taking endmember variability into account.

For the experimental results, the DeepGen algorithm, as observed in the study, outperformed existing algorithms including FCLS, PLMM, ELMM, and GLMM in terms of abundance estimation and material identification. It consistently achieved lower RMSE values for abundances and demonstrated more accurate endmember modeling, as evidenced by lower SAM values. Furthermore, the results indicated that DeepGen strikes a balance between computational efficiency and performance. Although it had longer execution times compared to GLMM and FCLS, it still remained within reasonable bounds, positioning itself between the complexities of GLMM and PLMM.

Based on these findings, the future works for the thesis could include:

- DeepGen may be improved further by experimenting with other deep generative models or architectures that can improve abundance estimate and minimize reconstruction mistakes.
- Investigating techniques to improve DeepGen's computational efficiency without compromising performance which might require experimenting with parallel computing approaches or creating more efficient algorithms.
- Investigating techniques to improve DeepGen's computational efficiency without sacrificing performance which might involve experimenting with parallel computing approaches or creating more efficient algorithms.Extending the assessment to larger and more varied datasets

provides evidence of the DeepGen algorithm's performance and generalization under various situations and spectral resolutions.

- Investigating the integration of additional data or previous knowledge into the DeepGen method, such as geographical information or material-specific constraints, to increase the accuracy of abundance estimation and endmember modeling.
- Investigating the DeepGen algorithm's applicability to various relevant hyperspectral image processing tasks, such as target detection, classification, or anomaly detection, and comparing its performance to existing approaches.

REFERENCE

Alkhatib, M. Q., & Velez-Reyes, M. (2018). Superpixel-Based Hyperspectral Unmixing with Regional Segmentation. https://doi.org/10.1109/igarss.2018.8518222

Alexander Amini and Ava Amini, MIT 6. S191: Introduction to Deep Learning. IntroToDeepLearning.com

Borsoi, R. A., Imbiriba, T., Bermudez, J. C. M., & Richard, C. (2021). Fast Unmixing and Change Detection in Multitemporal Hyperspectral Data. IEEE Transactions on Computational Imaging, 7,975988. https://doi.org/10.1109/tci.2021.3112118

Borsoi, R. A., Imbiriba, T., & Bermudez, J. C. M. (2020). A Data Dependent Multiscale Model for Hyperspectral Unmixing With Spectral Variability. IEEE Transactions on Image Processing, 29, 3638–3651. https://doi.org/10.1109/tip.2020.2963959

Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing, 378, 112–119. https://doi.org/10.1016/j.neucom.2019.10.008

Borsoi, R. A., Imbiriba, T., & Bermudez, J. C. M. (2019). Improved Hyperspectral Unmixing with Endmember Variability Parametrized Using an Interpolated Scaling Tensor. https://doi.org/10.1109/icassp.2019.8683155

Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N. M., & Chanussot, J. (2013). Hyperspectral Remote Sensing Data Analysis and Future Challenges. IEEE Geoscience and Remote Sensing Magazine, 1(2), 636. https://doi.org/10.1109/mgrs.2013.2244672

Borsoi, R. A., Imbiriba, T., Bermudez, J. C. M., Richard, C., Chanussot, J., Drumetz, L., Tourneret, J., Zare, A., & Jutten, C. (2021). Spectral Variability in Hyperspectral Data Unmixing: A comprehensive review. IEEE Geoscience and Remote Sensing Magazine, 9(4), 223–270. https://doi.org/10.1109/mgrs.2021.3071158

Drumetz, L., Chanussot, J., & Jutten, C. (2020). Spectral Unmixing: A Derivation of the Extended Linear Mixing Model from the Hapke Model. IEEE Geoscience and Remote Sensing Letters, 17(11), 1866-1870. https://doi.org/10.1109/lgrs.2019.2958203

Drumetz, L., Chanussot, J., & Jutten, C. (2016). Variability of the endmembers in spectral unmixing: Recent advances. https://doi.org/10.1109/whispers.2016.8071675

Dell'Endice, F., Nieke, J., Koetz, B., Schaepman, M. E., & Itten, K. I. (2009). Improving radiometry of imaging spectrometers by using programmable spectral regions of interest. ISPRS J Photogramm Remote Sen, 64(6), 632–639. https://doi.org/10.1016/j.isprsjprs.2009.05.007

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

Heinz, D., & Chein-I-Chang. (2001). Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. IEEETransactions on Geoscience and Remote Sensing, 39(3), 529–545. https://doi.org/10.1109/36.911111

Imbiriba, T., Borsoi, R. A., & Bermudez, J. C. M. (2018). Generalized Linear Mixing Model Accounting for Endmember Variability. https://doi.org/10.1109/icassp.2018.8462214

Khan, M. S., Khan, H., Yousaf, A., Khurshid, K., & Abbas, A. (2018). Modern Trends in Hyperspectral Image Analysis: A Review. IEEE Access, 6, 14118–14129. https://doi.org/10.1109/access.2018.2812999

Li, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep Convolutional Neural Networks for Hyperspectral Image Classification. Journal of Sensors, 2015, 1–12. https://doi.org/10.1155/2015/258619

Li, Y., Zhang, H., & Shen, Q. (2017). Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. Remote Sensing, 9(1), 67. https://doi.org/10.3390/rs9010067

Makantasis, K., Karantzalos, K., & Doulamis, A. (2015).Deep supervised learning for hyperspectral data classification through convolutional neural networks. https://doi.org/10.1109/igarss.2015.7326945

Ma, W., & Lu, J. (2017). An Equivalence of Fully Connected Layer and Convolutional Layer. ArXiv, abs/1712.01252.

Nascimento, J. M. P., & Dias, J. (2005). Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing, 43(4), 898–910. https://doi.org/10.1109/tgrs.2005.844293

Palsson, B., Sigurdsson, J., Sveinsson, J. R., & Ulfarsson, M. O. (2018). Hyperspectral Unmixing Using a Neural Network Autoencoder. IEEE Access, 6, 25646–25656. https://doi.org/10.1109/access.2018.2818280

Prasad, S., & Chanussot, J. (2020). Hyperspectral Image Analysis: Advances in Machine Learning and Signal Processing. Springer Nature.

Prasad, S., & Chanussot, J. (2020). Hyperspectral Image Analysis: Advances in Machine Learning and Signal Processing. Springer Nature.

R.A. Borsoi, T. Imbiriba, J.C.M. Bermudez. (2019) Deep Generative Endmember Modeling: An Application to Unsupervised Spectral Unmixing. IEEE Transactions on Computational Imaging.

Rajabi, R., & Ghassemian, H. (2014). Spectral unmixing of hyperspectral imagery using multilayer NMF. IEEE Geoscience and Remote Sensing Letters, 12(1), 38-42.

Sánchez, S., Martin, G., Plaza, A., & Chang, C. (2010). GPU implementation of fully constrained linear spectral unmixing for remotely sensed hyperspectral data exploitation. In Proceedings of SPIE. SPIE. https://doi.org/10.1117/12.860775

Thouvenin, P., Dobigeon, N., & Tourneret, J. (2016). Unmixing multitemporal hyperspectral images with variability: An online algorithm. https://doi.org/10.1109/icassp.2016.7472298

T. Kouyama, Y. Yokota, Y. Ishihara, R. Nakamura, S. Yamamoto, and T. Matsunaga. (2016). "Development of an application scheme for the SELENE/SP lunar reflectance model for radiometric calibration of hyperspectral and multispectral sensors," Planetary Space Sci., vol. 124, pp. 76–83.

Uezato, T., Murphy, R. F., Melkumyan, A., & Chlingaryan, A. (2016). A Novel Endmember Bundle Extraction and Clustering Approach for Capturing Spectral Variability Within Endmember Classes. IEEE Transactions on Geoscience and Remote Sensing, 54(11), 6712–6731. https://doi.org/10.1109/tgrs.2016.2589266

Wang, L., & Zhao, C. (2015). Hyperspectral Image Processing. Springer.

Xu, X., & Shi, Z. (2017). Multi-objective based spectral unmixing for hyperspectral images. ISPRS Journal of Photogrammetry and Remote Sensing, 124, 54-69.

You W., Shen C., Guo X., Jiang X., Shi J., & Zhu Z. (2017). A hybrid technique based on convolutional neural network and support vector regression for intelligent diagnosis of rotating machinery. Advances in Mechanical Engineering, 9(6), 168781401770414. https://doi.org/10.1177/1687814017704146

Zhao, W., & Du, S. (2016). Spectral–Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. IEEE Transactions on Geoscience and Remote Sensing, 54(8), 4544–4554. https://doi.org/10.1109/tgrs.2016.2543748