OPTIMIZATION AND DEEP LEARNING BASED MULTI MODEL ABUNDANCE ESTIMATION AND UNMIXING ALGORITHMS FOR HYPERSPECTRAL IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

OKAN BİLGE ÖZDEMİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

DECEMBER 2020

OPTIMIZATION AND DEEP LEARNING BASED MULTI-MODEL ABUNDANCE ESTIMATION AND UNMIXING ALGORITHMS FOR HYPERSPECTRAL IMAGES

Submitted by **OKAN BİLGE ÖZDEMİR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics**

Prof. Dr. Sevgi Özkan Yıldırım
Head of Department, **Information Systems**

Prof. Dr. Yasemin Yardımcı Çetin
Supervisor, **Information Systems, METU**

Assoc. Prof. Dr. Alper Koz
Co-Supervisor, **Center for Image Analysis, METU**

Examining Committee Members:

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering, METU

Prof. Dr. Yasemin Yardımcı Çetin
Information Systems, METU

Assoc. Prof. Dr. Banu Günel Kılıç
Information Systems, METU

Assoc. Prof. Dr. Behçet Uğur Töreyin
Applied Informatics, Istanbul Technical University

Assoc. Prof. Dr. Seniha Esen Yüksel
Electrical and Electronics Engineering, Hacettepe University

Date: *18.12.2020*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Okan Bilge Özdemir

Signature : ________________

ABSTRACT

OPTIMIZATION AND DEEP LEARNING BASED MULTI MODEL ABUNDANCE ESTIMATION AND UNMIXING ALGORITHMS FOR HYPERSPECTRAL IMAGES

Özdemir, Okan Bilge
Ph.D., Department of Information Systems
Supervisor: Prof. Dr. Yasemin Yardımcı Çetin
Co-Supervisor: Assoc. Prof. Dr. Alper Koz

December 2020, 96 pages

Hyperspectral unmixing aims to identify the materials within the pixels of an image and estimate the corresponding abundance values of these materials. This thesis proposes an optimization based abundance estimation method for the case where the spectral signatures of the materials are available, and a deep learning based hyperspectral unmixing method for the case where the spectral signatures of the materials are unavailable. The proposed abundance estimation algorithm assumes that real data can contain complex interactions that cannot be modeled with a single model, and therefore, use multiple mixing models for determining the abundance of real data. The proposed optimization-based coarse-to-fine estimation algorithm first adopts a linear mixing model for the tested pixel until the error between the reconstructed and original pixel is smaller than a threshold. The algorithm then proceeds by integrating the other nonlinear mixing models to the cost function. Among various utilized optimization algorithms and metrics, the proposed solution with the sequential quadratic programming and spectral angle mapper combination is found more successful than other search methods and baseline algorithms. As the second contribution of this thesis, a new 3D convolutional encoder based deep learning method is proposed for hyperspectral unmixing by observing that the local neighborhood information is not sufficiently used for the unmixing problem in hyperspectral images. Given that nonlinear mixing has not been adequately covered in deep learning based hyperspectral unmixing literature, the proposed method is especially designed to solve the nonlinear mixture models with the 3D convolutional encoder structure. The proposed method gives better performance than the well-known pure material extraction and abundance detection algorithms on synthetic and real data.

Keywords: Hyperspectral Unmixing, Deep Learning, Abundance Estimation, 3D Convolutional Encoder, Nonlinear Mixtures

ÖZ

HİPERSPEKTRAL GÖRÜNTÜLERDE OPTİMİZASYON VE DERİN ÖĞRENME TABANLI ÇOK MODELLİ BOLLUK TAHMİNİ VE AYRIŞTIRMA ALGORİTMALARI

Özdemir, Okan Bilge
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Yasemin Yardımcı Çetin
Ortak Tez Yöneticisi : Doç. Dr. Alper Koz

Aralık 2020, 96 sayfa

Hiperspektral ayrıştırma, görüntünün içindeki malzemeleri tanımlamayı ve bu malzemelere karşılık gelen bolluk değerlerini tahmin etmeyi amaçlamaktadır. Bu tez, malzemelerin spektral imzalarının mevcut olduğu durum için optimizasyona dayalı bir bolluk tahmin yöntemi ve malzemelerin spektral imzalarının olmadığı durumlar için derin öğrenme tabanlı bir hiperspektral ayrıştırma yöntemi önermektedir. İlk çalışmada sunulan bolluk tespit algoritması gerçek verilerin tek bir modelle ifade edilemeyecek kadar karmaşık etkileşimler içerebilmesi varsayımına dayanmaktadır. Bu nedenle, gerçek verilerde bolluk tespiti yapılırken çoklu model kullanılması hedeflenmiştir. Önerilen optimizasyon tabanlı bolluk tespit algoritması, hedef piksele yakın bir hata oranına ulaşılana kadar doğrusal karışım modelini varsayan bir yaklaşımı benimser. Optimizasyon algoritması daha sonra maliyet fonksiyonunu, olası karışım modelleri için yeniden tanımlayarak işleme devam eder. Kullanılan çeşitli optimizasyon algoritmaları ve uzaklık metrikleri arasında, sıralı ikinci dereceden programlama ve spektral açı haritalama kombinasyonu ile önerilen çözüm, diğer arama yöntemleri ve temel algoritmalardan daha başarılı bulunmuştur. Bu tezin ikinci katkısı olarak, hiperspektral görüntülerde komşuluk bilgisinin ayrıştırma problemi için yeterince kullanılmadığı gözlemlenerek hiperspektral ayrıştırma için yeni bir 3 boyutlu evrişimli kodlayıcı tabanlı derin öğrenme yöntemi önerilmiştir. Doğrusal olmayan karıştırmanın daha önce sunulmuş derin öğrenme tabanlı hiperspektral ayrıştırma çalışmalarında yeterince ele alınmadığı göz önüne alındığında, önerilen yöntem doğrusal olmayan karışım modellerini 3D evrişimli kodlayıcı yapısıyla çözmek için tasarlanmıştır. Önerilen yöntem, sentetik ve gerçek veriler üzerinde iyi bilinen saf malzeme çıkarma ve bolluk tahmini algoritmalarından daha iyi performans göstermiştir.

Anahtar Sözcükler: Hiperspektral Ayrıştırma, Derin Öğrenme, Bolluk Tahmini, 3D Evrişimli Kodlayıcı, Lineer Olmayan Karışımlar

To My Family…

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Dr. Yasemin Yardımcı Çetin and my co-advisor Assoc. Prof. Dr. Alper Koz for their encouragement, invaluable guidance, constant support and friendly attitude throughout my research.

I would also like to thank my thesis monitoring committee members, Prof. Dr. Uğur Halıcı and Assoc. Prof. Dr. Banu Günel Kılıç, for their feedback and guidance during my thesis. I would also like to thank my thesis jury members, Assoc. Prof. Dr. Seniha Esen Yüksel and Assoc. Prof. Dr. Behçet Uğur Töreyin.

I am grateful to my friend Selvi Elif Gök for her valuable help during my study.

Special thanks to my friends Görkem Polat and Cihan Öngün for their support and suggestions. I would also like to thank Sibel Gülnar, Yücelen Bahadır Yandık, Ece Işık Polat, İlker Coşan and Umut Çınar for their endless support, encouragement since the beginning of this research. I thank my family for their love, trust and patience during this work.

I would like to express my deepest gratitude to my wife Havva Özdemir for her love, support and the patience provided during my study.

TABLE OF CONTENTS

ABSTRACT v
ÖZ vi
DEDICATION vii
ACKNOWLEDGMENTS viii
TABLE OF CONTENTS ix
LIST OF TABLES xii
LIST OF FIGURES xiii
LIST OF ABBREVIATIONS xvi

CHAPTERS

1. INTRODUCTION 1
  1.1. Motivation 3
  1.2. The Purpose of the Study 4
  1.3. Contribution of the Thesis 5
  1.4. Thesis Outline 6
2. HYPERSPECTRAL UNMIXING 9
  2.1. Number of Endmember Estimation Methodologies 10
  2.2. Endmember Estimation Algorithms 11
    2.2.1. Methods with Pure Pixel Assumption 11
    2.2.2. Methods Without Pure Pixel Assumption 12
  2.3. Abundance Estimation Methodologies 13
    2.3.1. Linear Mixing Model 14
    2.3.2. Non-linear Mixing Model (NMM) 15
    2.3.3. Intimate Mixing Model (IMM) 16
  2.4. Optimization Methodologies 16
    2.4.1. Convex Optimization Methods 17
    2.4.2. Stochastic Optimization Methods (SQP) 18
3. DEEP LEARNING AND ITS APPLICATIONS ON HYPERSPECTRAL UNMIXING 21
  3.1. Convolutional Neural Networks (CNN) 22
    3.1.1. Convolution Layer 23
    3.1.2. Fully Connected Layer 23
    3.1.3. Pooling Layer 24
    3.1.4. Batch Normalization 24

3.1.5. Dropout Layer 25
3.1.6. Activation Functions 26
3.1. Autoencoders 28
3.2. Descent Based Optimization Methods 29
3.2.1. Stochastic Gradient Descent (SGD) 29
3.2.2. Adaptive Gradients (Adagrad) 30
3.2.3. Adaptive Moment Estimation (Adam) 30
3.3. Hyperspectral Unmixing with Deep Learning 30
4. EXPERIMENTAL SETUP AND DATASET 33
4.1. Datasets Used in Experiment 33
4.1.1. Synthetic data generation for abundance estimation 33
4.1.2. Synthetic data generation for deep learning 35
4.1.3. Utilized real data for experiments 36
4.2. Distance Metrics 37
5. PROPOSED COARSE TO FINE ABUNDANCE ESTIMATION FOR HIGHLY MIXED HYPERSPECTRAL DATA 41
5.1. Proposed Method 41
5.2. Experimental Results and Discussion 44
5.2.1. Selection of Design Parameters 45
5.2.2. Selection of Distance Metric 47
5.2.3. Comparison of the Coarse to Fine and Direct Searches 48
5.2.4. Coarse to Fine Approaches with Different Number of Endmembers 49
5.2.5. Comparison of the Proposed Method with the Baseline Methods in the Literature50
5.2.6. Comparisons with the Baseline Literature on Real Data 53
6. PROPOSED NONLINEAR UNMIXING WITH 3D CONVOLUTIONAL ENCODER (3DCE) 57
6.1. Proposed 3D Convolutional Encoder Based Hyperspectral Unmixing 57
6.2. Experimental Results and Discussion 63
6.2.1. Experiments for the Utilized Optimization Method and Cost Function 63
6.2.2. Batch Size Experiments 65
6.2.3. Endmember Extraction Performance 66
6.3. Comparison of abundances with baseline methods in the literature 68
6.4. Real Data Experiments 71
6.4.1. Comparisons with Literature 71
6.4.2. Comparison with Respect to Mixing Models and Nonlinear Layer 78

6.4.3. Cross-validation of the Proposed Unmixing Model 79

7. CONCLUSIONS 81

REFERENCES 83

CURRICULUM VITAE 94

LIST OF TABLES

Table 1 Durations of distance metrics for GA, SA, PS, and SQP algorithms 47
Table 2 Duration of the optimization algorithms per pixel (in seconds) for coarse to fine and direct searches 48
Table 3 Performance of the algorithms for the data with different mixing models (LMM, FM and PPNM) 52
Table 4 Main structure and parameters of 3DCE 62
Table 5 3DCE with Adam optimizer abundance estimation performance for synthetic data with batch size 1 in terms of RMSE 63
Table 6 3DCE with SGD optimizer abundance estimation performance for synthetic data with batch size 1 in terms of RMSE 64
Table 7 3DCE with Adagrad optimizer abundance estimation performance for synthetic data with batch size 1 in terms of RMSE 65
Table 8 3DCE with Adams algorithm abundance estimation performance change on the batch size in terms of RMSE 65
Table 9 3DCE with SGD algorithm abundance estimation performance change on the batch size in terms of RMSE 66
Table 10 3DCE with Adagrad algorithm abundance estimation performance change on the batch size in terms of RMSE 66
Table 11 Abundance Estimation Performance with the Extracted Endmembers in terms of RMSE 70
Table 12 The performance of the algorithms for Jasper Ridge dataset as RMSE 74
Table 13 The performance of the algorithms for Samson Ridge dataset as RMSE ... 75

LIST OF FIGURES

Figure 1 The electromagnetic spectrum and the transmittance of the earth's atmosphere [2] 1
Figure 2 Number of publication of hyperspectral imaging per year (Source: Google Scholar) 3
Figure 3 (a) Sample scene and (b) spectral signatures for grass, soil and mixture (50% grass and 50% soil) 9
Figure 4 The Flowchart of the Hyperspectral Unmixing Process 10
Figure 5 SISAL Convergence Example 13
Figure 6 (a) Linear Mixing Model, (b) Non-Linear Mixing Model (c) Intimate Mixing Model 13
Figure 7 Model of a neuron 21
Figure 8 General architecture of CNNs 22
Figure 9 Convolution layer example 23
Figure 10 Sample fully connected network 24
Figure 11 Sample max-pooling process 24
Figure 12 Sample dropout effect on feed-forward network 25
Figure 13 Response for (a) Sigmoid Activation Function, (b) Tanh Activation Function 27
Figure 14 ReLU response 27
Figure 15 Basic autoencoder architecture 28
Figure 16 Learning rate effect (a) low learning rate, (b) high learning rate 29
Figure 17 Simple autoencoder based hyperspectral unmixing scheme 31
Figure 18 Examples for the selected spectral signatures from the hyperspectral library 34
Figure 19 A sample Ground Truth for synthetic data for the abundances of three different endmembers utilized for synthetic data generation (a) abundance map for the first endmember, (b) abundance map for the second endmember, and (c) abundance map for the third endmember 34
Figure 20 Noise levels on spectral signature (a) 40db Noise (b) 10db Noise 35
Figure 21 Spectral signature of the selected endmembers for deep learning method 35
Figure 22 Abundance maps for synthetic data. (a) Abundance map for endmember 1, (b) Abundance map for endmember 2, (c) Abundance map for endmember 3 and (d) Abundance map for endmember 4 36
Figure 23 RGB image of Samson Ridge (a), Ground truth classes (b) Soil, (c) Tree and (d) Water and RGB image of Jasper Ridge (e) and Ground Truth Classes (f) Tree, (g) Water, (h) Soil and (i) Road 37
Figure 24 An example for the change of SAM distance as a function of abundance value for two different endmembers 42

Figure 25 Flowchart of the proposed coarse to fine methodology for abundance estimation 43
Figure 26 (a) Change of error rates of SA, PS and SQP algorithms for different iteration numbers (b) Average abundance estimation time per pixel changes of SA, PS and SQP algorithms for three different endmembers with respect to iteration number 45
Figure 27 (a) RMSE of GA with respect to population and generation size (b) Detection time per pixel change with respect to population and generation size for GA 46
Figure 28 Comparison of GA, SA, PS, and SQP algorithms with SAM, L1 and L2-Norm as RMSE 47
Figure 29 Comparison of coarse to fine and direct searches 48
Figure 30 Comparison of optimization algorithms for different numbers of endmembers 49
Figure 31 Sample ground truth and results for SQP for synthetic data (a-c-e the ground truths for endmember1, endmember2 and endmember3 respectively, b-d-f the results for corresponding endmembers) 50
Figure 32 The comparison of the proposed method with the state of the art algorithms for different endmembers 50
Figure 34 Sample signature with and without 10db noise and estimated signature .. 52
Figure 35 Abundance estimation error for Samson Ridge and Jasper Ridge data sets 53
Figure 36 The results for Jasper Ridge and Samson Ridge. (a), (b) and (c) are the ground truths for the abundances for soil, tree and water, respectively and (d), (e) and (f) are the estimated abundances with the proposed algorithm for Samson Ridge data. (g),(h),(i) and (j) are the ground truths for the abundances for tree, water, soil and road, respectively, and (k), (l),(m) and (n) are the estimated abundances with the proposed algorithm for Jasper Ridge data 54
Figure 37 Displaying the models of the results obtained with the proposed algorithm on the basis of pixels LMM(Black), FM(Grey), PPNM(White) (a-Model estimation results for Samson Ridge, b- RGB image of Samson Ridge, c- Model estimation results of Jasper Ridge and d-RGB image of Jasper Ridge data 55
Figure 38 The convolution part of the proposed 3DCE model 58
Figure 39 An example for the application of filtering for 3D convolutional part...... 59
Figure 40 The autoencoder part of the proposed 3DCE model 60
Figure 41 Nonlinear part of the Proposed 3DCE Model 60
Figure 42 Proposed 3D convolutional autoencoder based deep learning model (3DCE) structure 61
Figure 43 The endmembers extracted with 3DCE using Adams algorithm 67
Figure 44 The endmembers extracted with 3DCE using SGD algorithm 67
Figure 45 The endmembers extracted with 3DCE using Adagrad algorithm........ 68
Figure 46 Extracted endmember with VCA (a) and SISAL(b) with synthetic data .. 69
Figure 47 (a) Ground Truth and the results for (b) VCA+PPNM, (c) SISAL+PPNM and (d) Proposed 3DCE Method 70

Figure 48 Extracted Endmembers with (a) VCA, (b) SISAL and (c) 3DCE for Jasper Ridge dataset 72
Figure 49 Extracted Endmembers with (a) VCA, (b) SISAL, and (c) proposed 3DCE for Samson Ridge dataset...73
Figure 50 The abundance estimation results for Jasper Ridge dataset (a) ground truth, (b) the results for VCA + MLM (c) the results for SISAL+MLM and (d) the results with proposed 3DCE..76
Figure 51 Error map showing the difference between the abundance map derived by the proposed 3DCE method and the ground truth for Jasper Ridge dataset 76
Figure 52 The abundance estimation results for Samson Ridge dataset (a) ground truth, (b) the results for VCA + MLM (c) the results for SISAL+MLM, and (d) the results with proposed 3DCE method 77
Figure 53 Error map showing the difference between the abundance map derived by the proposed 3DCE method and the ground truth for Samson Ridge dataset............78
Figure 54 Displaying the models of the Samson Ridge and Jasper Ridge Data, LMM(Black), FM(Grey), PPNM(White) (a-Results for Samson Ridge, b- RGB image of Samson Ridge, c- Results of Jasper Ridge and d-RGB image of Jasper Ridge 79

LIST OF ABBREVIATIONS

Adagrad	Adaptive Gradients
Adam	Adaptive Moment Estimation
BN	Batch Normalization
CNN	Convolutional Neural Network
DAEN	Deep Autoencoder Network
DCAE	Deep Convolutional Autoencoder Network
FM	Fan Model
GA	Genetic Algorithms
GBM	Generalized Bilinear Model
GPU	Graphical Processing Unit
HFC	Harsanyi–Farrand–Chang
HySime	Hyperspectral Signal Identification By Minimum Error
IEA	Iterative Error Analysis
IMM	Intimate Mixing Model
LMM	Linear Mixing Model
MAE	Mean Absolute Error
MCMC	Markov Chain Monte Carlo
MLM	Multi Linear Mixing Model
MNF	Minimum Noise Fractions
MVES	Minimum Volume Enclosing Simplex
MVSA	Minimum Volume Simplex Analysis
NM	Nascimento Model
NMF	Nonnegative Matrix Factorization
NMM	Non-Linear Mixing Model
NNSAEs	Stacked Nonnegative Sparse Autoencoders
PPI	Pixel Purity Index
PPNM	Polynomial-Post Nonlinear Multivariate
PS	Pattern Search
ReLU	Rectified Linear Unit
rmsAAD	Root-Mean-Square Of The Abundance Angle Distance
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Networks
SA	Simulated Annealing
SAD	Spectral Angle Distance
SAE	Stacked Autoencoders
SAM	Spectral Angle Mapper
SGA	Simplex Growing Algorithm

SGD	Stochastic Gradient Descent
SID	Spectral Information Divergence
SISAL	Simplex Identification Via Split Augmented Lagrangian
SQP	Sequential Quadratic Programming
SVD	Singular Value Decomposition
SVR	Support Vector Regression
USGS	U.S. Geological Survey
VAE	Variational Auto Encoders
VCA	Vertex Component Analysis
VD	Virtual Dimensionality

CHAPTER 1

INTRODUCTION

Remote sensing is defined as any method of image and spatial data acquisition, including aerial measurement and photogrammetry, independent of being satellite-based, airborne-based based, or ground-based environments [1]. Yet, in a more general sense, remote sensing refers to the evaluation of information obtained by various sensors from a distant object without direct intervention. Today's remote sensing technologies have a wide variety of applications in many military and civil areas, such as defense, environment, agriculture, atmosphere, urbanism, and health. For example, remote sensing is used in forest fire control, land use and land cover classification in environmental applications, and drought monitoring, crop production forecasting, and crop recognition in agriculture.

In a remote sensing scenario, the sensor collects the signals originating from a light source after reflected from an object. If this light source is a natural source such as the sun, it is called passive sources. In the case of an energy-dependent source such as laser or radar, they are named as active sources. Provided that an energy source is available, almost any wavelength can be used to display the desired scene's properties. However, this situation may have some limitations. For example, when imaging from satellite sensors, there are wavelengths absorbed by the molecular components of the atmosphere. Figure 1 shows the transmittance of the earth's atmosphere on a path between space and earth over a wide range of electromagnetic spectrums. As can be seen in the figure, the transmittance decreases at certain wavelengths absorbed by the atmosphere, water vapor, and carbon dioxide, and even there are bands with no transmittance.

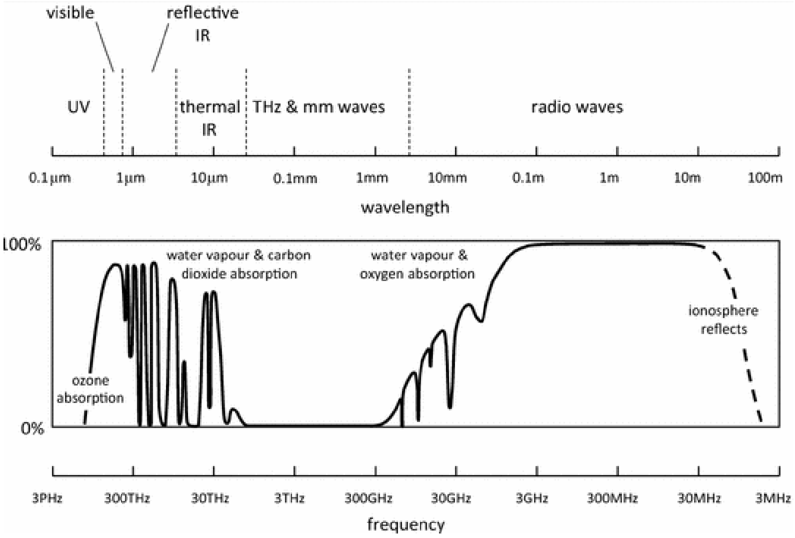


Figure 1 The electromagnetic spectrum and the transmittance of the earth's atmosphere [2]

Depending on the platform used, remote sensing techniques can be examined under two main headings, as ground and airborne platforms [3]. Remote sensing is usually applied when high detail is needed or when the working area is small, with sensors mounted on platforms close to the ground. Camera and radar are examples of sensors used on the ground platform. The second platform type, i.e., the airborne platform, can be classified as aircraft platforms and spacecraft platforms. Image resolution may vary depending on the platform and sensor used. Therefore, the platform should be selected according to the desired application.

Another way to classify remote sensing systems is with respect to the number of spectral bands they use. According to this classification, the first class is the panchromatic imaging system that has only one band image sensor. There are many panchromatic imaging systems, such as QuickBird-PAN and IKONOS-PAN. The second remote sensing systems, namely multispectral systems, there are several spectral bands. These systems, which have been used since the 1970s, have a very important place in remote sensing [4]. Advanced Land Imager, ASTER, MODIS, SPOT, and SENTINEL imaging systems can be given as examples for multispectral imaging systems. Satellite sensors such as Quickbird and Commercial Remote Sensing Satellite can capture both multispectral and panchromatic images.

Hyperspectral imaging has been enhanced with the use of high spectral resolution in multispectral imaging. Hyperspectral sensors can acquire very narrow, contiguous spectral information in many consecutive bands (nominally> 50), from visible to thermal infrared wavelengths. These sensors enable continuous reflection or emissivity information to be acquired. While multispectral images have low spectral resolution and high spatial resolution, hyperspectral images have high spectral resolution with low spatial resolution. Material characterization and recognition are possible with this high spectral resolution. Thus, it has been possible to use hyperspectral images in many areas, such as urban and regional planning, agriculture, mining, and military decision support.

Hyperspectral imaging has been utilized until now for various applications with different requirements. The first task is hyperspectral classification, defined as assigning a unique tag to each pixel. The classification is divided into two main classes as unsupervised or supervised. Unsupervised classification is based on the automatic clustering of pixels by algorithms without user intervention. The properties of the classes resulting from the unsupervised classification are initially unknown. The analyst must compare the classified image with other reference information to obtain more detailed information. In the supervised classification, it is already known which classes the image will be divided into or which classes are desired to be obtained from the image. For this process, learning data belonging to determined classes from the image is given as input to the algorithm. The algorithm then determines the classes of input data using this learning data. The second application is dimensionality reduction, mainly to reduce the data load while avoiding data analysis results. Since hyperspectral images have more than a hundred bands, dimensionality reduction is required to remove redundant information and speed up the process. Target detection is another major application in hyperspectral images, which often employs spectral signatures of materials. In addition to hyperspectral applications, there is change detection task, which is the process of detecting changes in two different hyperspectral images. For example, it can be used to view the change in time of a region taken at two different times. Besides, change detection is common in areas such as natural disasters, agricultural product management, or tracking water resources.

Finally, hyperspectral unmixing is one of the fundamental operations for hyperspectral image processing. Since hyperspectral cameras contain too many bands, they generally have high spectral resolutions and low spatial resolutions. For this reason, there is usually more than one material in the area covered by a pixel. Applications such as classification, target detection, or segmentation generally require knowing what the materials in this pixel are and how much they are. Hyperspectral unmixing includes applications related to both the materials contained in the data and the determination of these materials' abundance values in each pixel.

1.1. Motivation

The importance of hyperspectral image processing techniques has increased with the widespread use of hyperspectral images. Figure 2 emphasizes this fact by giving the number of publications in recent years. The main reason for such a popularity of hyperspectral imaging is both related to the decrease in the costs and the increase in the quality of the sensors with the progress of the technology.

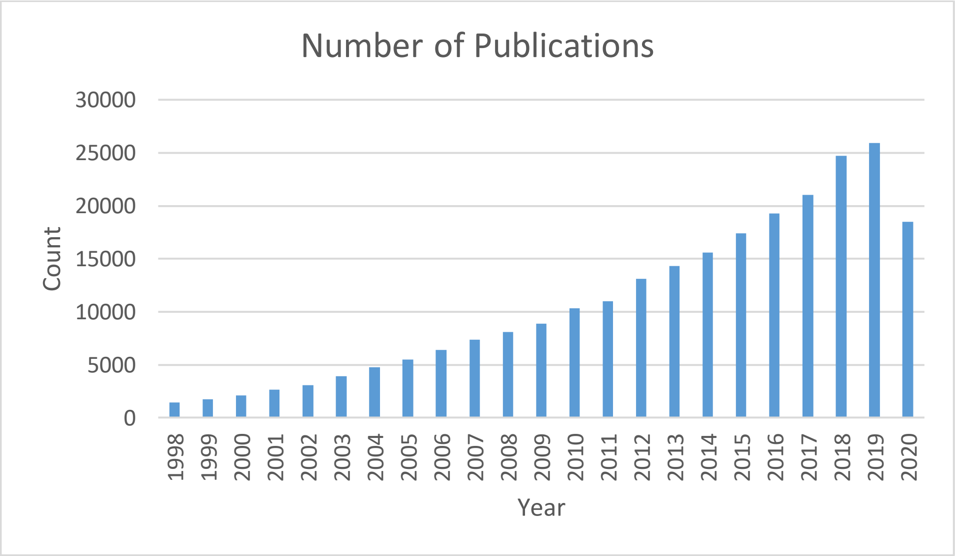


Figure 2 Number of publication of hyperspectral imaging per year (Source: Google Scholar)

One of the biggest problems of hyperspectral sensors stands out as low resolution. As a result, the subpixel detection methods have formed one of the main branches in hyperspectral image processing to detect low-resolution images' targets. The hyperspectral unmixing mainly developed for subpixel detection consists of the determination of the number of endmembers on the data, the extraction of the members, and the estimation of corresponding abundances for the endmembers, in which the amount of products in the pixel is determined. It has been observed that the solutions in the literature have reached a certain maturity for the estimation of the number of endmembers and for the estimation of the endmembers. The most obvious problem in detecting abundance values for endmembers is to model the interactions of spectral signatures caused by the reflection of light rays during the capturing of hyperspectral data from real scenes. Due to their structure, hyperspectral images can not be modeled with a single model. Therefore, more than one model is required to solve this problem. The models in existing solutions try to include as many scenarios as possible, but their performance on real data is often unsatisfactory. The main reason for this is that the presented mixture models are difficult to model the interactions of the materials and the distance units used in the algorithms indicate low performances in real hyperspectral data. In order to solve these problems, a multi-model optimization-based coarse to fine approach to work on real data is utilized in this thesis. The performances of different distance metrics are also investigated in such a framework.

On the other hand, deep learning methods have recently been used for hyperspectral unmixing. However, the studies on hyperspectral unmixing are usually performed independently of pixel neighborhood information, which is referred as blind unmixing. In blind unmixing problems, each pixel is given as separate input, and the result of the unmixing is obtained. This causes the reflection of materials to be modeled without using spatial knowledge. Therefore, the integration of the local neighborhood knowledge to the unmixing process is considered as an underlying idea to increase the unmixing performances. To this end, convolutional networks, which significantly increase the performances in image processing applications, are proposed for hyperspectral unmixing in order to include the spatial information in the unmixing process. Furthermore, the design is specially tailored by adding specific layers to represent nonlinear mixtures. The performance of the proposed deep learning algorithm based on 3D convolutional autoencoder has been tested on real data with different optimization methods and distance metrics.

1.2. The Purpose of the Study

The algorithms proposed for hyperspectral unmixing are known to perform successfully with synthetic data generated by using the presumed mixing model. However, this may not always be the case for real data as the real data can be too complex to be modeled using a single mixing model. For example, an image taken from a flat surface can be modeled with a single mixing model, while there may also be data from mountainous, high-rise, or wooded areas that can be modeled better with multiple mixing models. Given this observation, this study has two objectives regarding hyperspectral unmixing.

The first objective of this study is to determine the abundance rates for the cases where there is endmember information on real data that cannot be modeled with a single mixture model. Therefore, in the first part of the study, an optimization-based coarse to fine abundance estimation method is proposed. The proposed method performs hyperspectral abundance estimation by using more than one mixing model in a single optimization process. Experiments are carried out to examine the performance variation of the method with different optimization algorithms and different distance metrics. Optimization algorithms used in this study are determined as genetic algorithms, simulated annealing, sequential quadratic programming, and pattern search, while distance metrics are determined as spectral angle mapper, L1-norm, and L2-norm. The performance of these optimization algorithms and distance metrics for both direct search and coarse to fine approach has been examined. The proposed coarse-to-fine approach continues the optimization process with the linear mixing model (LMM) up to a certain threshold, then the cost of the minimization process is determined as the minimum of the multiple models. The effects of the parameters of optimization algorithms on performance are investigated in experiments with synthetic data. In order to speed up the optimization process, the difference between the coarse-to-fine approach and direct search approaches is analyzed, and their performance is compared. A comparison of the proposed method with other algorithms in the literature has been made in the experiments performed with the highly mixed synthetic data and two different real data.

The second objective is to use deep learning algorithms, which have been widely used in hyperspectral unmixing in recent years and stand out with their high performance. The purpose of this study is to perform hyperspectral unmixing in cases where the endmember information is not available. In the study, three-dimensional convolution networks are used to benefit from the use of spatial information. The output of the three-dimensional convolution networks used as a predecessor is then given as input to the automatic encoder based neural networks. In this method, the performances of different distance units and optimization algorithms are tested for both synthetic and real data. Stochastic gradient descent, adaptive gradients, and adaptive moment estimation algorithms, which are widely used in the literature, are used for the experimental comparisons. Spectral angle mapper, mean square error, L1-norm, and spectral information divergence methods are utilized to choose the best distance metric in terms of abundance and endmember estimation performances. More specifically, the performances are evaluated by using the root mean squared error between the original and reconstructed abundances.

1.3. Contribution of the Thesis

As the first contribution of this study, an optimization based coarse to fine abundance estimation method is proposed by using multi mixing models to cover the complex interactions in real hyperspectral data. The proposed approach, which combines different mixture models in a single framework, which has various experimental aspects, which involve the selection of design parameters, determination of best optimization method and distance metric, and the comparison of the coarse to fine approach with the direct search. Given these aspects, the main contributions for this part of the thesis can be summarized as follows.

- Different optimization algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA), Sequential Quadratic Programming (SQP) and Pattern Search (PS) are adapted to the proposed coarse to fine abundance estimation framework for hyperspectral unmixing and compare with each other by properly selecting design parameters for each optimization method. The best optimization algorithm with the proposed model is revealed with respect to the abundance estimation performance for highly mixed hyperspectral data.

- The effect of distance metrics such as L1-Norm, L2-Norm and Spectral Angle Mapper (SAM) on optimization performance is examined and compared with the proposed method.

- The performances of the proposed coarse to fine method and the direct search method are compared. Experiments indicate that the coarse-to-fine based multi-model approach provides similar performances as the direct search approach but the convergence time for optimization algorithm is lower.

- The experiments are further detailed by the comparisons with the algorithms in the literature using noiseless and noisy synthetic and real data. The better performance of the proposed abundance estimation method with respect to the state of the art algorithms is validated.

As a second contribution of this study, an unsupervised hyperspectral unmixing algorithm is presented by integrating 3-dimensional convolutional networks to frequently used autoencoder structure. In addition, nonlinear part is included to model for unmixing nonlinear mixing models. The main contribution in this part of the thesis are given as follows:

- The experimental comparisons with respect to the traditional autoencoder based unmixing methods have revealed the superiority of the proposed method for abundance estimation. It has been observed that the spatial information to unmixing process with 3D convolutional encoders significantly improve the hyperspectral unmixing performance.

- The performance of the endmember and abundance estimation with the proposed method are found better than the conventional endmember estimation methods in the literature such as Vertex Component Analysis, Simplex Augmented Langrangian and abundance estimation methods such as Linear Mixing Model, Multi Linear Mixing model and Polynomial Post Nonlinear Mixing model.

- Finally, the addition of nonlinear part for nonlinear mixing models indicate promising performances for nonlinear mixtures compared to the proposed structure without nonlinear part. The suitability of the proposed structure with nonlinear layer to real data which can involve nonlinear interactions is verified with the experiments.

1.4. Thesis Outline

The outline of the thesis is as follows:

Chapter 2 provides an introduction to the hyperspectral unmixing problem. A comprehensive literature review on the main stages of the hyperspectral unmixing problem, hyperspectral mixing models, and optimization algorithms are presented.

Chapter 3 includes the main layers and components used in deep learning algorithms and an examination of previous hyperspectral unmixing studies for hyperspectral unmixing.

In Chapter 4, the procedure for the generation of synthetic data is given along with the real data set utilized in the experiments. This section also describes the distance metrics used for both abundance estimation and deep learning methods.

Chapter 5 includes a description of the proposed optimization based coarse to fine abundance estimation method. This chapter includes the experiments performed for the parameter selection of the algorithms used for the presented abundance estimation model. Different distance metrics and experiments for coarse to fine search and direct search are also included in this section. Additionally, the comparisons of the presented algorithm with the abundance estimation methods commonly used in the literature are included.

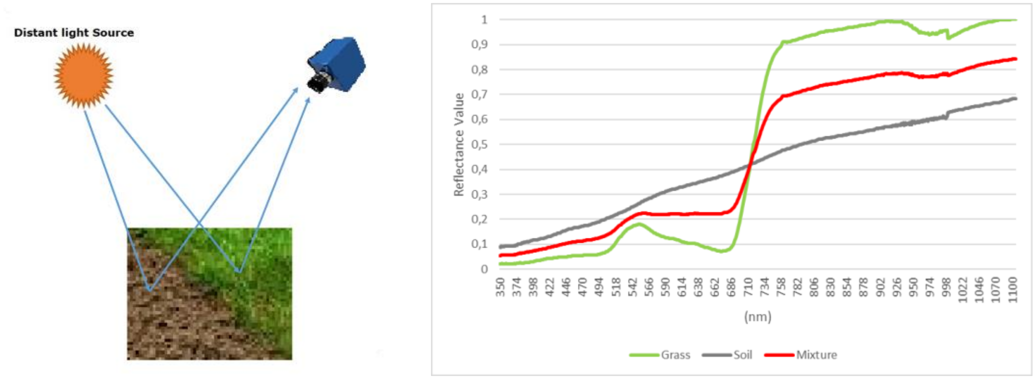
In Chapter 6, the proposed 3D convolution and autoencoder based method for hyperspectral mixing is given. The model elements and parameters are explained in this section. In addition, experimental comparisons between the proposed method and baseline methods in the literature for endmember estimation and abundance estimation are performed and discussed.

In Chapter 7, the main conclusions of the thesis are given.

CHAPTER 2

HYPERSPECTRAL UNMIXING

In hyperspectral imaging, the spatial resolution of the utilized sensors is generally high due to the speed and hardware requirements brought by the excessive wavelengths. This situation causes many different materials to enter into the area covered by a pixel on the earth. In each pixel of the obtained image, the mixture of the spectrum of the materials physically present in that pixel is observed. In Figure 3-a, the mixture spectra obtained for a pixel with 50% grass and 50% soil in a pixel are given as an example in addition to the grass and soil signatures. Detecting the substances in the spectral mixture and determining the proportion of this substance in each pixel is a problem that has been frequently mentioned in the literature and still maintains its importance today. Especially in applications such as target detection, a process is required to detect targets smaller than pixel size in the low spatial resolution image. Hyperspectral unmixing is presented for the solution of such problems, which aims to determine how many materials are in the given hyperspectral data cube, what these materials are and where these materials are located in the data.



(a) (b)

Figure 3 (a) Sample scene and (b) spectral signatures for grass, soil and mixture (50% grass and 50% soil)

The process of hyperspectral unmixing [5-8], which is illustrated in Figure 4, mainly consists of three stages, which are the estimation of the number of endmembers, extraction of endmembers with respect to this estimated number, and the estimation of the abundances for each endmember after the endmember extraction. In this chapter, the existing solutions in the literature for these three problems are presented. To extract the endmember from the data, the endmember number must first be known. For example, in Figure 3, the number of endmembers is determined by running the endmember number extraction algorithm. With this result, which is output as 2 in this example, the endmember extraction algorithm is then run, and again for this example, the grass and soil signatures are detected in Figure 3 (b). In the abundance estimation stage, which is the final stage of hyperspectral mixing, the amount of each pixel from each end member is determined using the output of the endmember extraction algorithm. For example, as a result of abundance estimation in the example above, it will be determined that some pixels are 100% grass or soil, and some pixels contain both grass and soil.

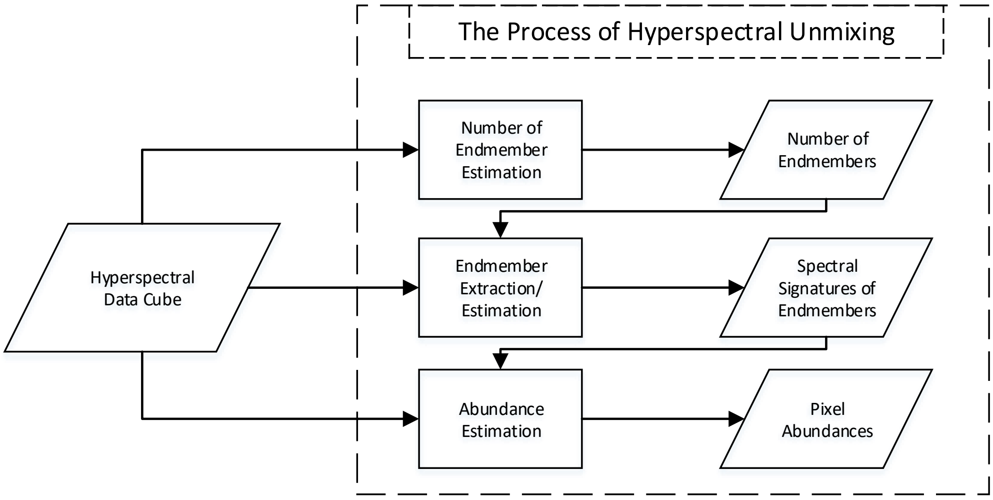

The Process of Hyperspectral Unmixing
Hyperspectral Data Cube
Number of Endmember Estimation
Number of Endmembers
Endmember Extraction/ Estimation
Spectral Signatures of Endmembers
Abundance Estimation
Pixel Abundances

Figure 4 The Flowchart of the Hyperspectral Unmixing Process

In this chapter, detailed information about the main steps of the hyperspectral unmixing process is given. The main steps include the number of endmember estimation methodologies, endmember extraction methodologies, abundance estimation methodologies. This chapter also includes detailed information about mixing models.

2.1. Number of Endmember Estimation Methodologies

The process of determining the number of endmembers is very important for obtaining the preliminary information necessary for the realization of the second and third stages of hyperspectral unmixing, endmember determination, and abundance estimation. The main purpose of this stage is to determine how many different materials are in the given hyperspectral data cube.

One of the well-known methods proposed for estimation of the number of endmembers is the Virtual Dimensionality (VD) proposed by Chang and Du [9]. VD method assumes that the eigenvalues of correlation and covariance matrices for a particular component are close to each other. The equality of correlation and covariance eigenvalues in each specific component was tested by the Neyman-Pearson method, and the number of components containing the signal was determined as VD. The Harsanyi–Farrand–Chang (HFC) method, which is an improved version of [10], estimates the number of endmembers with the VD term using the Neyman-Pearson method. The HFC method uses eigenvalues of sample correlation and covariance matrices with automatic thresholding for the number of endmember estimation.

As another example, hyperspectral signal identification by minimum error (HySime) applies the singular value decomposition technique to hyperspectral data [11]. The eigenvectors which best represent data in the root-mean-square error sense are determined along with the number of endmembers. Markov Chain Monte Carlo (MCMC)-based method for estimation of the number of endmembers is proposed by Tourneret [12]. The method is however, only applicable for a small number of endmembers due to its high computational cost.

2.2. Endmember Estimation Algorithms

2.2.1. Methods with Pure Pixel Assumption

N-FINDR, one of the most widely used methods for endmember extraction presented by Winter [13], selects the purest pixel in the hyperspectral image. The N-FINDR algorithm works on the principle that the volume of pure pixels in the data will be larger than the volume created by different combinations of other pixels. This volume calculation is done by the Minimum Noise Fractions (MNF) algorithm [14]. MNF is used to reduce the data to p-1 size as a preprocessing stage, where p is the number of endmembers. The algorithm calculates volume by starting with the random pixel selection and checks whether the new pixels selected in each iteration are larger than this volume. If the volume is larger than the volume found in the previous iterations, new endmember candidates are selected. The algorithm is terminated after testing all pixels. The simplex growing algorithm (SGA) [15] is very similar to N-FINDR. The algorithm grows simplexes at each iteration. Each new corner that defines the maximum simplex volume is considered a new endmember. As the initial conditions are assigned randomly, the algorithm can indicate inconsistency in finding the correct endmembers.

Another well-known study on this subject is the Pixel Purity Index (PPI) algorithm [14]. PPI algorithm uses MNF for dimensionality reduction. After this step, the algorithm creates a large set of random vectors called skewers. Extreme values are calculated for each projection. A pixel purity image is formed where each pixel is scored with the number of times the pixel is recorded as an extreme point. Pixels with the highest score are determined as the purest signals and returned as the endmembers. Many variations of the PPI algorithm are presented, such as random PPI, parallel PPI, iterative PPI, or graphical processing unit (GPU) implemented PPI [16-20].

The vertex component analysis (VCA) algorithm is presented by Nascimento and Dias [21]. Its performance is reported as better than N-FINDR and PPI algorithms. The algorithm firstly reduces the data to the p-1 dimensional Euclidean space using Singular Value Decomposition (SVD), where p is the number of endmembers, allowing each vector on the hyperspectral data to appear in this space. Then, extreme points in each band of the projected space are selected as endmembers. VCA works with very low computational cost and high accuracy compared to other algorithms. Because of this performance, it is also generally used as the starting point in the methods without a pure pixel assumption [22–24].

Iterative error analysis (IEA), as the last example in this group, has a high computation cost [25]. The algorithm extracts the endmember and calculates the error between the generated and real data for each iteration. The average spectrum of the data is selected to start the process. The constrained linear unmixing process is performed with this mean vector, and the error caused by the errors remaining in each pixel after the unmixing is calculated. The spectral signal corresponding to the pixel with the largest single error is selected as the second endmember, and the hyperspectral unmixing process is performed again. This process continues until the error falls below a certain threshold value or until the desired endmember number is reached.

With the development of hyperspectral sensors, the resolution in hyperspectral images is increasing. Therefore, algorithms working with these pure pixel-based assumptions that select pixels from within the image are considered to be important. Among these algorithms, VCA is thought to work more successfully than other algorithms.

2.2.2. Methods Without Pure Pixel Assumption

Minimum volume-based endmember extraction methods assume that there are no pure pixels in the data. In these methods, the pure pixels from which the data is formed are determined by different minimum volume detection methods. Among the algorithms presented with the principle of minimum volume, the Minimum Volume Simplex Analysis (MVSA) algorithm [23] uses the endmembers from the above-mentioned VCA algorithm as the seed point. Then, in each iteration, the volume of these points was expanded, and the endmember extraction was carried out with the SQP method.

Figure 5 shows a convergence example for three endmembers with Simplex Identification via Split Augmented Lagrangian (SISAL) algorithm [24]-another algorithm that uses the VCA algorithm as its seed point. SISAL regards the minimum volume definition as a non-convex optimization problem with convex constraints. It performs endmember extraction by determining the minimum volume from the starting points obtained using the VCA algorithm with quadratic approaches. The use of the augmented Lagrange multipliers has made the algorithm computationally efficient. Unlike MVSA, SISAL is more resistant to noise and strong against errors in initial values. In the figure, M(0) is the seed point assigned using VCA. M(k) is the new point assigned by the algorithm at each iteration. M (final) is given as the last point reached by the algorithm.

Another algorithm in this group, Minimum Volume Enclosing Simplex (MVES) algorithm integrates the concepts of convex analysis and volume minimization using linear programming and cyclic minimization [26]. This method is also based on the assumption that the simplex surrounding the minimum volume must coincide with the true endmember simplex using rigid positivity constraints. Later, a robust MVES (RMVES) algorithm is presented to reduce the sensitivity of the MVES algorithm to noise [27]. In this algorithm, positivity constraints of abundance rates are used as soft constraints. Additionally, quadratic solvers are used in the RMVES algorithm.

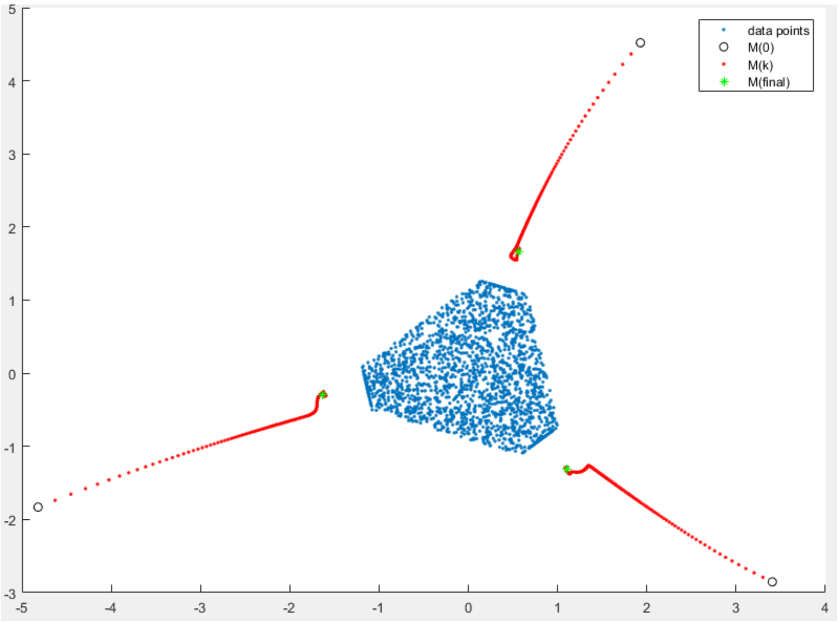


Figure 5 SISAL Convergence Example

Finally, the Iterative constrained endmembers (ICE) algorithm replaces the volume simplex with the squared distances between all the simplex vertices [28]. Sparsity Promoted Iterative Constrained Endmember (SPICE) is an extension of the ICE algorithm that incorporates sparsity-promoting priors to estimate also the number of endmembers [29]. These two algorithms, which make the endmember estimation with pseudoinverse, do not impose any restrictions when predicting endmembers as in the SISAL algorithm. As a result, output endmembers can take values less than 0 and greater than 1. In order to solve this problem, the extension of the SPICE (SPICEE) algorithm is recommended [30].

2.3. Abundance Estimation Methodologies

In the abundance estimation stage, which is the last step of the hyperspectral unmixing process, the percentage of endmembers in each pixel is determined. One of the most important problems of this stage is determining the mixing model of the data. These mixing models can be categorized as Linear Mixing Model (LMM), Non-Linear Mixing Model (NMM), and Intimate Mixing Model (IMM), which are created by taking into account the different interactions of the light in or between the materials in the scene [31-33].

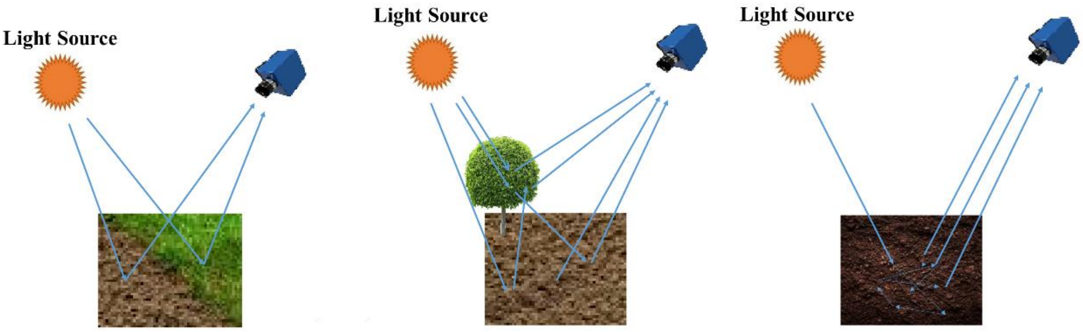


Figure 6 (a) Linear Mixing Model, (b) Non-Linear Mixing Model (c) Intimate Mixing Model

Figure 6 shows the light interactions between objects before they reach the sensor for the main categories of mixing models. The most common mixing model is LMM, which assumes that the incident light reflected from the surface comes directly to the sensor without any further interference [34-38]. Another mixing model NMM assumes that the light interacts with another material before reaching the sensor. Although there are different approaches in this model, the most common approach is the bilinear mixing model, which assumes that the light consecutively hits two materials before reaching the sensor [39-42]. The radiance value obtained in the sensor as a result of this interaction is considered to be a non-linear mixture of light reflected directly and light interacting with the material.

This nonlinearity is modeled in [40] and [41], by adding an extra interaction term to the LMM, described as the Hadamard products of the endmembers and multiplied by a certain coefficient. While this coefficient is introduced as a product of abundances by Fan et al. [40], Nascimento introduced the coefficient as an independent parameter in [41]. Unlike the bilinear mixing models presented as NML, multi linear mixing model (MLM) offered by Heylen and Scheunders was created with the assumption that there might be multiple and complex interactions between materials. Finally, the IMM assumes that the signal interacts with multiple materials at microscopic levels. Therefore, the radiance in the sensor consists of lights scattered from more than one material. Such interactions are originally modeled by Hapke in his seminal paper [43], and various researchers such as Nascimento and Bioucas-Dias [44] and Rand et al. utilize that model to unmix the hyperspectral data with IMM [45].

2.3.1. Linear Mixing Model

Figure 6 (a) illustrates the interactions in LMM, which is one of the most commonly used mixing models in hyperspectral unmixing. Due to its simplicity, LMM is one of the highly preferred mixing models in literature [46]. The formulation of LMM is given as,

$$\boldsymbol{s} = \sum_{i=1}^{p} a_i \boldsymbol{e}_i + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{s}$ and $\boldsymbol{e}_i$ are the L-dimensional spectral signatures of pixel and the ith endmember vector respectively, a_i is the fractional abundance corresponding to endmember $\boldsymbol{e}_i$, p corresponds to the number of endmembers, and $\boldsymbol{n}$ denotes the noise.

The given LMM formulation is subject to two constraints, which are described as

$$\begin{cases} a_i \geq 0, & \forall i \\ \sum_{i=1}^{p} a_i = 1 & \end{cases} . \tag{2}$$

The non-negativity constraint, the first of these two constraints, states that the abundance values must be greater than zero [47]. In the second constraint, i.e., the sum to one, the sum of the abundance values of the endmembers in each pixel must be one [36].

2.3.2. Non-linear Mixing Model (NMM)

Figure 6 (b) illustrates the interactions between the objects in the case of the Nonlinear Mixing Model. Bilinear models commonly used in NMM literature assume that the interaction between materials occurs only once. One of the first bilinear models is Nascimento Model (NM) [41], which is given as,

$$\boldsymbol{s} = \sum_{r=1}^{p} a_r \boldsymbol{e}_r + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{i,j} \boldsymbol{e}_i \odot \boldsymbol{e}_j + \boldsymbol{n}. \tag{3}$$

The given formulation describes the interaction as a sum of two terms and noise, where the first term stands for linear relation of the endmembers and the second term refers to the non-linear interaction, which is represented by the Hadamard product of the endmembers,

$$\boldsymbol{e_i} \odot \boldsymbol{e_j} = \begin{pmatrix} e_{1,i} \\ \cdots\cdots \\ \cdots\cdots \\ e_{L,i} \end{pmatrix} \odot \begin{pmatrix} e_{1,j} \\ \cdots\cdots \\ \cdots\cdots \\ e_{L,j} \end{pmatrix} = \begin{pmatrix} e_{1,i}e_{1,j} \\ \cdots\cdots \\ \cdots\cdots \\ e_{L,i}e_{L,j} \end{pmatrix}. \tag{4}$$

In (3), $\beta_{i,j}$ determines the effect of the interaction between the endmembers, i, and j. This model also has the sum to one and positivity constraints, which are given as

$$\begin{cases} a_i \geq 0 \quad \forall_i \\ \beta_{i,j} \geq 0 \quad \forall_{i \neq j} \\ \sum_{i=1}^{p} a_i + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{i,j} = 1 \end{cases}. \tag{5}$$

As another example of bilinear models, Fan *et al.* [40] proposed a model which modifies the nonlinearity coefficient, $\beta_{i,j}$, as a function of abundance coefficients, a_i and a_j,

$$\boldsymbol{s} = \sum_{r=1}^{p} a_r \boldsymbol{e}_r + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} a_i a_j \boldsymbol{e}_i \odot \boldsymbol{e}_j + \boldsymbol{n}. \tag{6}$$

The basic approach in this model is that the endmembers in each pixel affect each other with respect to their abundances. The contribution of the endmember abundances outside the present pixels is assumed zero. The Fan Model (FM) cannot be generalized to LMM as the coefficient of the Hadamard product is directly dependent on the abundances. In order to solve this problem, Halimi *et al.* [39] proposed to change the coefficients, $\beta_{i,j}$,as $\gamma_{ij} a_i a_j$,

$$\boldsymbol{s} = \sum_{r=1}^{p} a_r \boldsymbol{e}_r + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \gamma_{ij} a_i a_j \boldsymbol{e}_i \odot \boldsymbol{e}_j + \boldsymbol{n}. \tag{7}$$

In this the so-called the Generalized Bilinear Model (GBM), γ_{ij} accounts for the non-linear interactions between materials. When γ_{ij} are set to zero, the model generalizes to LMM, and alternatively, when they are set to 1, the model converts to the FM.

In the previously described models of expressions (3), (6), and (7), the terms $\boldsymbol{e}_i \odot \boldsymbol{e}_i$ are excluded since the iteration of endmembers indexes are from 1 to *(p-1)* and from *(i+1)* to *p*. So, possible interactions inside an endmember can not be included. On the other hand, the Polynomial-Post Nonlinear Multivariate (PPNM) model extends the index to all endmember combinations to include those interactions [42],[48] with expression,

$$\boldsymbol{s} = \sum_{r=1}^{p} a_r \boldsymbol{e}_r + b \sum_{i=1}^{p} \sum_{j=1}^{p} a_i a_j \boldsymbol{e}_i \odot \boldsymbol{e}_j + n \tag{8}$$

where b is a scalar coefficient utilized for the adjustment of non-linear part.

2.3.3. Intimate Mixing Model (IMM)

As the last mixing model, the intimate mixing model (IMM), which assumes that photons interact with each other many times, is shown in Figure 6 (c). One important model in defining IMM is the Hapke model [43] that is based on bidirectional reflectance theory. In this model, the interactions were formulated with the average scattering of photons in materials, the incident and emergence angles of scattering, and a simplified form of Chandrasekhar’s function [49], which describes the multiple scattering as a function of incident and emergence angles. IMM models are excluded from this research due to their complexity in applications.

2.4. Optimization Methodologies

The optimization methods used in this study can be examined under two main headings as convex optimization methods and stochastic optimization methods. While the purpose of both kinds of methods is to minimize the cost function with a set of parameters, their main difference is the way they reach the minimum point. While convex optimization algorithms usually try to reach the minimum by using the gradient direction of the cost function, stochastic optimization algorithms try to minimize the given cost function by randomly generating alternative candidate solutions.

The optimization problem for the abundance estimation can be described as finding the abundance values for endmembers, which minimizes a distance metric D between the estimated pixel and the target pixel. Such an optimization problem is given as,

$$\boldsymbol{a}_* = \arg\min_{\boldsymbol{a}} (D(\boldsymbol{s}, \boldsymbol{t})) \tag{9}$$

where $\boldsymbol{a}_*$, element of R^p , is an abundance vector, $\boldsymbol{t}$ is the actual spectral signature of the target pixel, and $\boldsymbol{s}$ is the estimated spectral signature of the target pixel. Given a mixture model M as described with one of the models in (1), (3), (6), (7) or (8), the optimization problem can be further expressed as,

$$\boldsymbol{a}_* = \arg\min_{\boldsymbol{a}} (D(M(\boldsymbol{a}, \boldsymbol{E}), \boldsymbol{t})) \tag{10}$$

where $\boldsymbol{a}$ is the abundance matrix and $\boldsymbol{E}$ is a matrix of size *Lxp*, which is composed of corresponding endmembers, $\boldsymbol{E}=[\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_p]$.

In the literature, different algorithms have been offered for both convex optimization and stochastic optimization methods as a solution to this problem. This section firstly examines sequential quadratic programming [50–52], and pattern search [53], [54] algorithms in convex optimization methods. Then, frequently used genetic algorithms [55–58] and simulated annealing [59–61] in stochastic methods are examined.

2.4.1. Convex Optimization Methods

If the constraints of a given minimization problem are convex, the problem is expressed as a convex optimization problem [62]. The most significant advantage of recognizing or formulating a problem as a convex optimization problem is that the convex optimization methods converge to the global minimum value. This section explains the sequential quadratic programming method and pattern search method, as commonly used convex optimization methods for nonlinear optimization.

2.4.1.1. Sequential Quadratic Programming (SQP)

Different mixture models such as LMM, FM, and GBM can be formulated as constrained non-linear multivariable optimization problem. Sequential Quadratic Programming is one of the main algorithms proposed for non-linear optimization problems, which is based on the active set method [63] to determine the initial guesses, and the quasi-Newton line search [64] algorithm to update these guesses [51].

Given the abundance vector as $\boldsymbol{a}$, the constraints can be further expressed as,

$$\arg\min_{\boldsymbol{a}} \left(D\left(\,M(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}\right)\right)\, such\ that \begin{cases} a \le u_b \\ l_b \le a \\ c_{eq}(a) = 1 \end{cases} \tag{11}$$

where $c_{eq}(a)$ corresponds to the sum to one constraint, $\boldsymbol{a}$ is the abundance vector, and l_b and u_b are lower and upper bounds for the abundances, which are selected as 0 and 1, respectively. The model M is selected as the linear mixture model given in (2) [34–38].

2.4.1.2. Pattern Search (PS)

Pattern search algorithm is a direct search numerical optimization method proposed by Hooke and Jeeves [65]. Later, a method considering the lower and upper boundary constraints was also developed by Findler et al. [53]. As the main difference compared to other algorithms such as gradient-based algorithms, the objective function in PS is not required to be continuous or differentiable. Further details of convergence analysis of PS, and optimality conditions, and the gradient search relation can be found in [54].

As in SQP, the PS algorithm also defines the cost function for the abundance estimation as a constrained non-linear optimization problem. However, rather than using gradient-based search methods to find the optimal point, the PS algorithm recursively tries to find an improving direction to determine the optimal point. This improving direction should not necessarily be the best direction, which in turn makes the algorithm operate on discontinuous cost functions.

2.4.2. Stochastic Optimization Methods (SQP)

This section provides a summary of leading algorithms with remarks on the critical issues related to stochastic optimization, whose popularity has grown rapidly in recent years. Stochastic optimization methods are used in nonlinear, high dimensional problems where classical deterministic optimization methods cannot be used [66]. Genetic algorithms and simulated annealing algorithms are commonly used stochastic optimization methods [67], which have been selected in this thesis for abundance estimation. The performances of such methods are compared with respect to different parameters such as complexity, endmember number and abundance estimation performances.

2.4.2.1. Genetic Algorithms (GA)

Genetic algorithms are one of the most popular stochastic optimization techniques inspired by genetic biology [68]. Given a random gene pool for a population, the algorithm iteratively finds the optimum solutions by generating new genes with mutations and crossover operations and selects the best survivals in the population with respect to a cost function. The crossover operator mixes the two selected chromosomes to generate better genes. The mutation is produced by creating a change in some genes in a chromosome. It prevents convergence to a population with a homogeneous gene pool, thus guarantees chromosome diversity.

GA algorithm is adapted to an abundance estimation problem in [57] by defining the cost function as the spectral angle [69] between the target pixel and its estimation with sum to one and positivity constraints. The authors also performed various experiments to determine the best population size.

In another study, Tong et al. utilized a GA-based approach to improve the abundance estimation to locate the materials determined in pixels, where the cost function is defined as Spectral Dependence Index [70]. Finally, Farzam et al. [55] propose a GA-based endmember and abundance estimation algorithm by assuming the data is formed by LMM. The authors define the cost function as a mean square error between the target pixel spectra and its estimation with GA and show the effect of population size on the abundance estimation.

While these previous studies mainly focus on one mixing model to define the abundance estimation, the proposed approach given in the next section will define the abundance estimation as an optimization problem with respect to more than one mixing model.

2.4.2.2. Simulated Annealing (SA)

Simulated Annealing (SA) is a stochastic based general search technique, which can deal with highly non-linear models, noisy data, and many constraints. The simulated annealing method first proposed by Metropolis et al. [71] is based on the similarity between the annealing process in physical systems and the solution process in optimization problems [72]. Annealing is the process of slowly cooling the solids after they are heated to the annealing temperature. The annealing process is called as the general heat treatments performed to relax, soften the material, and make the inner structure more usable.

The SA algorithm determines an initial candidate solution and accepts this solution as the best solution in the beginning. It then randomly chooses a valid solution at each step and measures the quality of that point. In the next steps, the first candidate solution considered as the best solution is compared with the newly produced solution. If the new result is better, the new solution is appointed as the best solution. The temperature parameter is a value initially determined in the annealing method. This value is reduced by the temperature lowering parameter at the end of the cycle created. These processes can be continued until the best solution is found, or until a specified time to run the algorithm, or until the temperature parameter is zero or less than zero. The advantage of the SA algorithm is that it can overcome local minima, unlike gradient-based optimization algorithms.

SA algorithm is used by Penn for the estimation of abundance values for LMM [59]. The author uses L2-Norm as a loss function to calculate the difference between the target pixel and the estimated pixel spectra by assuming sum to one and positivity constraints for the abundance values. Rather than using L2-norm, Debba et al. [73] also employs the SA algorithm by defining the cost function as the error of the first and second derivatives of different sets of target spectral signature and their estimations. The proposed algorithm [73] calculates the difference value by subtracting the derivative of the estimated pixel spectrum from the derivative of the target pixel. In their later studies [61], the same researchers also investigate the effect of noise on the estimation performance.

The algorithms mentioned in this section are implemented in a single optimization process to make abundance estimation with the multi-mix model. Then, the experiments performed for abundance estimation are evaluated in both synthetic and real data.

CHAPTER 3

DEEP LEARNING AND ITS APPLICATIONS ON HYPERSPECTRAL UNMIXING

Artificial neural networks are computer systems that are formed by modeling the neural network in the brain and accordingly consist of interconnected nodes called artificial neurons. Artificial neural networks have been in use since the 1950s and have gained popularity several times. However, each time there appeared computational difficulties that could not be overcome and resulted in the loss of popularity. The main reason for this was seen as the lack of processing power of the computers at that time. However, especially after the study of Hinton et al. in 2006, it regained its popularity [74].

One of the first studies on this subject, conducted by McCulloch and Pitts, can be seen in Figure 7 of the binary threshold unit created using virtual neurons [75]. The activation output of a single virtual neuron, i.e., h, is produced by taking the weighted sums of the n-dimensional input, and the final binary output is obtained by testing the activation output h against a threshold value, i.e., u.

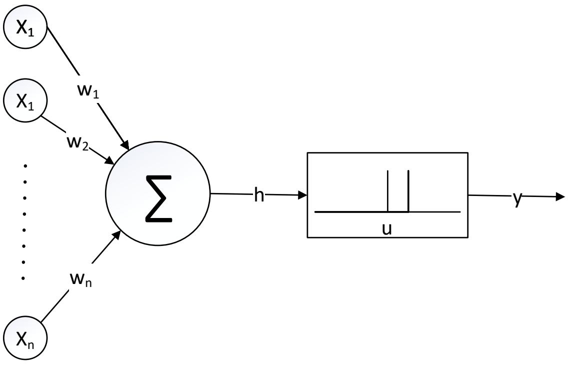


Figure 7 Model of a neuron

This process can be specified mathematically as:

$$\boldsymbol{y}_i = f_i\left(\sum_{j=1}^{n} \boldsymbol{w}_j \boldsymbol{x}_j + \boldsymbol{b}_i\right) \tag{12}$$

Where, $\boldsymbol{x}_j$ is the *j*th component of the n-dimensional input signal, $\boldsymbol{w}_j$ is the *j*th component of the weight vector, $\boldsymbol{b}_i$ is the bias of the node and $\boldsymbol{y}_i$ is the output of the function. f_i is the activation function, which is usually nonlinear [76]. Detailed information about these functions will be given in Chapter 3.1.6.

Deep learning takes its name due to its deep layers and hierarchical structure. Two factors that make deep learning techniques popular today, which has a deeper structure than other artificial neural networks, are that the data reaches very large dimensions and the development of processors that will process this data by training. In addition, parallel processing of this huge data available in current graphical processing units (GPU) has made a significant contribution to the development of deep learning. This chapter explains the general components of convolutional neural networks and autoencoders which are widely used in deep learning applications [77–86]. The hyperspectral applications of these structures are also included in Section 3.4.

3.1. Convolutional Neural Networks (CNN)

Convolutional neural networks have a special place in image analysis because they seamlessly combine the convolution-based feature extraction stage with the following pattern recognition part. In visual object recognition, CNNs generally provide high performance as a result of their high capability to represent the neighborhood relations of the image. In their seminal article that laid the foundation of the CNN framework, LeCun et al. published a multi-layered artificial neural network called LeNet-5 for handwritten digit identification where CNN structure is used to enable direct recognition of visual patterns from raw pixels [87].

Figure 8 shows the general architecture of CNNs. CNN's generally consist of the convolution layer, where convolution is applied with different filters to extract features from the image, the pooling layer where the resolution on these filters is reduced, and fully connected layers for high-level reasoning [88], [89].

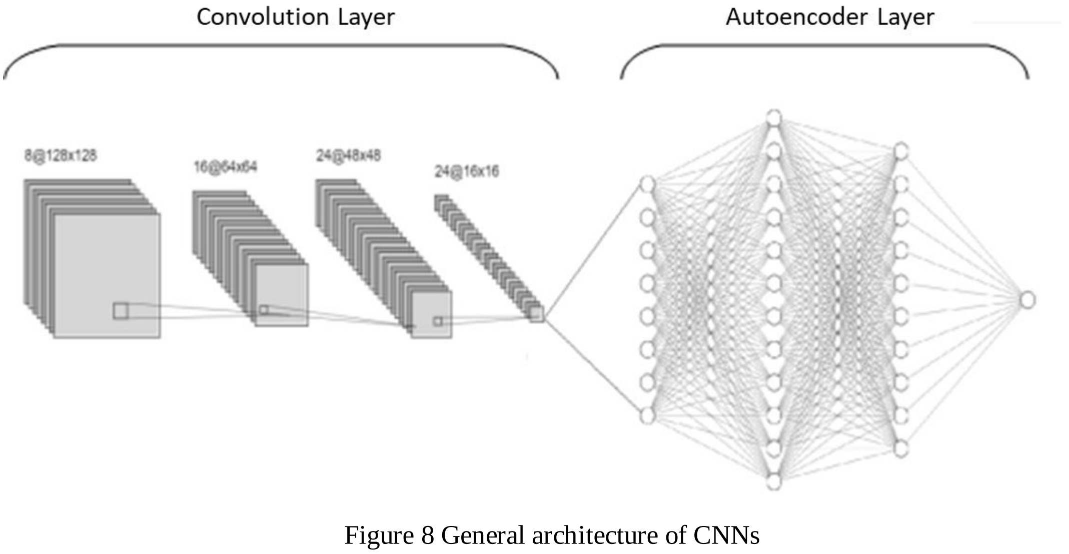


Figure 8 General architecture of CNNs

This section introduces the main CNN components. The most commonly used layers in CNN networks are the convolutional layer, pooling layer, dropout, and fully connected layer. Detailed information about batch normalization and activation functions are also presented in this section.

3.1.1. Convolution Layer

The convolution layer, which is the characteristic component of a CNN, reveals the neighborhood relations and spatial locality features on the image using different filters. While the filters slide along the image, the image pixel values that lie under the filter masks are multiplied with the values in the filter, and the resulting values are added to yield the convolution.

The sample drawing for this process is given in Figure 9. A new image called convolution attribute is obtained from the image on which the convolution layer is applied. The method used to calculate the *K* value in the figure is given in the expression (13). In the expression, *K* is the filter output, M is the location of the coordinates of the filter on the image, and F is the filter.

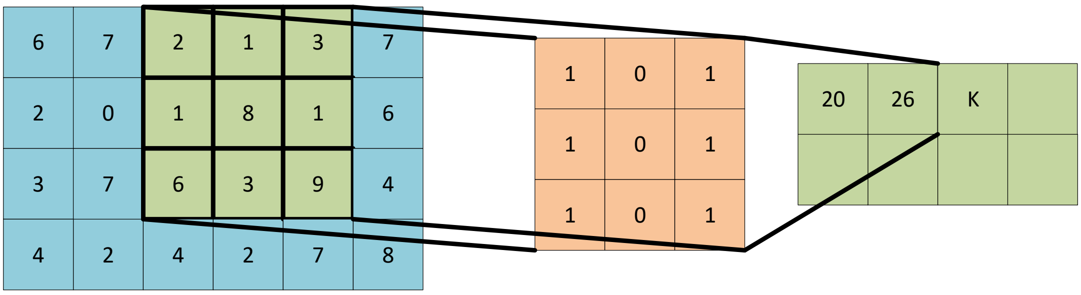


Figure 9 Convolution layer example

$$
\begin{aligned}
K = {} & \left(M_{(1,1)} * F_{(1,1)}\right) + \left(M_{(1,2)} * F_{(1,2)}\right) + \left(M_{(1,3)} * F_{(1,3)}\right) \\
& + \left(M_{(2,1)} * F_{(2,1)}\right) + \left(M_{(2,2)} * F_{(2,2)}\right) + \left(M_{(2,3)} * F_{(2,3)}\right) \\
& + \left(M_{(3,1)} * F_{(1,1)}\right) + \left(M_{(3,2)} * F_{(3,2)}\right) + \left(M_{(3,3)} * F_{(3,3)}\right)
\end{aligned} \tag{13}
$$

In addition to the filter size, there are stride and padding parameters used in convolution operation. The stride parameter specifies how often the filter is shifted while sliding over the image. The padding parameter expresses how and to what extent the edge pixels will be extended. Each filter focuses on a different feature in the image. By the use of filters, different versions of an image emphasizing different aspects are obtained.

3.1.2. Fully Connected Layer

A fully connected network example is shown in Figure 10. The fully connected layer is usually located after the feature extraction stage in convolutional neural networks. This layer works on an input where each input is connected to all neurons. The fully connected layer takes the feature maps as input and prepares them as the algorithm output as a 1D array.

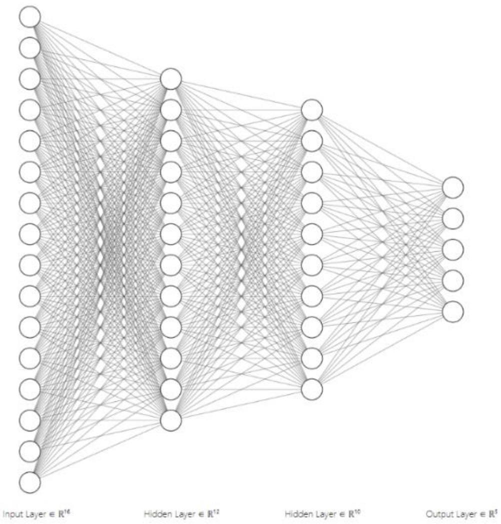
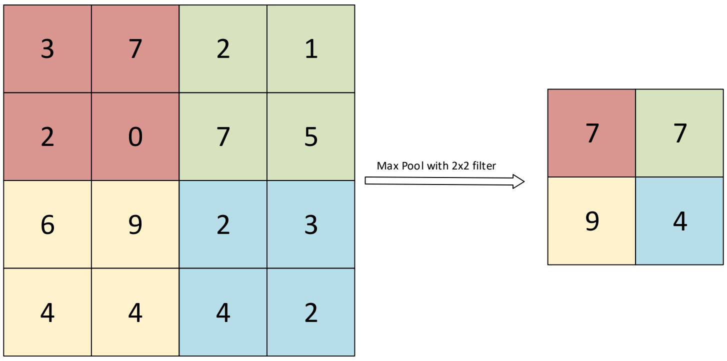


Figure 10 Sample fully connected network

3.1.3. Pooling Layer

The pooling layer often follows the convolution layer on CNNs. Its purpose is to sub-sample the image given as input to invariance to local translation in order to reduce the number of parameters and to make feature extractions with fewer parameters. Processing speed can also be increased by reducing the size of the image. This reduction is usually performed by processing the image with a square filter. There are two main filters commonly used for these processes. One of them takes the biggest value in the filter (Max Pooling), and the other takes its average value (Average Pooling). With this process, the sub-sample of the image is obtained.

Figure 11 shows 2x2 max pooling applied to a 4x4 image. In this example, the stride is selected as 2, and padding is not applied. The resulting image is shown in Figure 11 (b).



3	7	2	1
2	0	7	5
6	9	2	3
4	4	4	2

7	7
9	4

(a) (b)

Figure 11 Sample max-pooling process

3.1.4. Batch Normalization

Batch normalization (BN) is the normalization of deep neural network activation function inputs in fully connected or evolutionary layers during training [90]. BN allows higher learning rates to be used in training and more tolerance for random initiation of network parameters. Experiments have been conducted to examine the effect of batch normalization on the learning rate by comparing gradients between collectively normalized and non-normalized networks [91]. The gradients with respect to comparable parameters turned out to be often larger in normalized networks, and this difference is more significant at the lower layers of the network. It has been reported that this increase in indifference allows for high learning rates.

BN is applied as a layer in the deep neural networks and normalizes the activations of the previous layer. The mathematical formulation of BN is given in (14). Expectation and variance are computed for (pre)activation vector x, where x is a n-dimensional vector.

$$\mathrm{BN}(\mathrm{x};\, \gamma, \beta) = \beta + \gamma \odot \frac{x_i - E[x_i]}{\sqrt{var[x_i]} + \varepsilon} \tag{14}$$

where γ, β, and ε are model parameters that determine the mean, standard deviation, and the regularization parameters respectively [92].

3.1.5. Dropout Layer

The dropout layer changes the structure of the network. It has been proposed to remedy the problem of memorizing data called overfitting. Different techniques, such as regularization, try to overcome this problem by imposing restrictions on parameters and/or changing the cost function.

Sample dropout usage is given in Figure 12. In Figure 12 (a) the dropout is not applied to the network. The state after dropout is applied, as seen in Figure 12 (b). An excessive number of neurons or connections often slows the process down and may result in over-sensitivity to initial conditions. For example, a network with 50 hidden layers may have hundreds of thousands of weights. Such a large network requires both a lot of learning data and memory [93]. The dropout function enables the removal of unnecessary connections by randomly dropping units in the training phase.

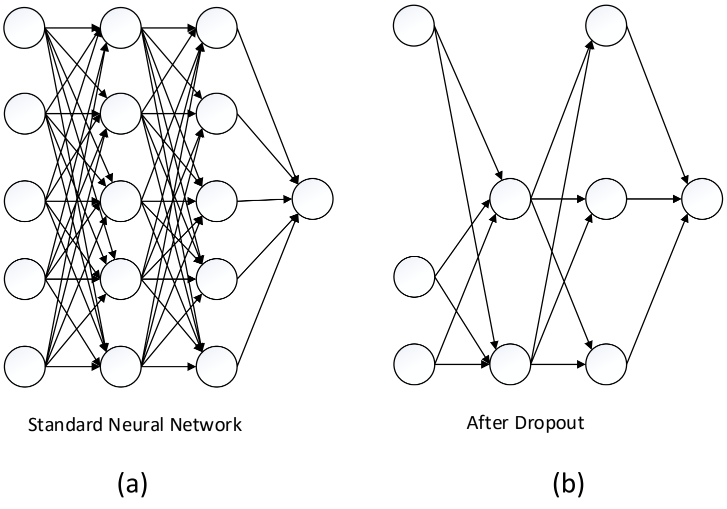


Figure 12 Sample dropout effect on feed-forward network

3.1.6. Activation Functions

The activation functions are often used to convert input signals to nonlinear output signals to assist in learning higher-order polynomials for deep networks. Differentiability is often a desirable property of nonlinear activation functions.

By taking the information coming to the layer, a linear activation determines the output by computing the weighted sum of inputs and biases. The formulation for linear activation function is

$$y_i = \sum_{j=1}^{n} w_j x_j + b_i \tag{15}$$

where y_i is the output of the ith node [94]. For nonlinear activation functions, the formulation is

$$y_i = f_i \left(\sum_{j=1}^{n} w_j x_j + b_i \right) \tag{16}$$

where $f_i$is the activation function.

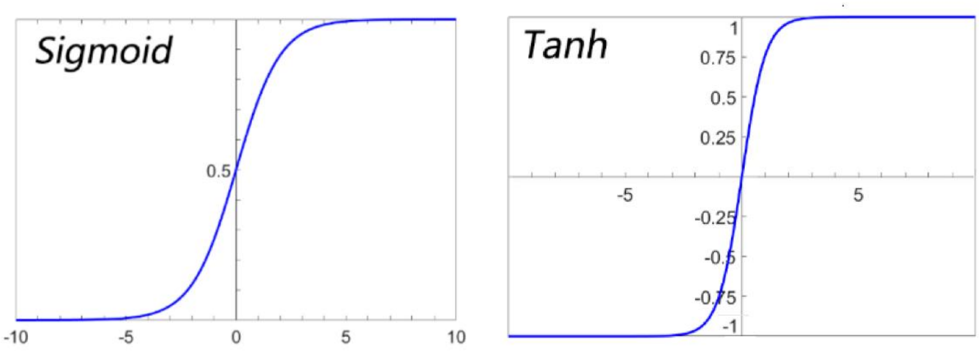
There have been quite a few studies on activation functions. Commonly used functions such as sigmoid, tanh, softmax, rectified linear unit will be examined in the following section.

3.1.6.1. Sigmoid and Tanh

The sigmoid activation function, which is sometimes called the logistic function in the literature, is a nonlinear activation function mostly used in feedforward neural networks [94], [95]. The tanh function is more preferred in multi-layer neural networks as it provides higher accuracy compared to the sigmoid function [96]. The formulation for sigmoid and tanh activation functions [94] is given in (17).

$$\begin{gathered} f_{sigmoid}(x_i) = \frac{1}{(1 + e^{-x_i})} \\ f_{tanh}(x_i) = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} \end{gathered} \tag{17}$$

Figure 13 shows the responses for the Sigmoid and tanh activation functions. The sigmoid function applies to each value of the vector given as input separately and pulls them to a value between 0 and 1, while the hyperbolic tangent function (tanh) maps the input between -1 and +1.



(a) (b)

Figure 13 Response for (a) Sigmoid Activation Function, (b) Tanh Activation Function

Logistics functions are used especially in problems involving more than one tag for an input. This function allows the inputs to be expressed as a discrete probability distribution. Softmax function, which is the generalized form of the logistics function, is used in multi-class but single-label data sets. The limitations are that for tanh and sigmoid, they are saturated because large values are assigned to 1, and small values are assigned to -1 or 0, respectively.

3.1.6.2. Rectified Linear Unit (ReLU)

To solve the limitation of tanh and sigmoid functions, a corrected linear unit (ReLU) activation function is proposed by Nair and Hinton [97]. ReLU is reported as a commonly used successful activation function [98]. By applying the Rectified Linear Unit function to the output of the feature map, a non-linear feature is added to the result. Although there are several functions occasionally offered to be used instead of ReLU in the literature, ReLU remains as the most commonly used one since the performance of other functions severely decreases when the dataset is changed.

Figure 14 shows the response for ReLU activation function. The ReLU function $F(x) = \max(0, x)$ returns 0 for values with $x \leq 0$ on each attribute value, and x for values with $x > 0$.

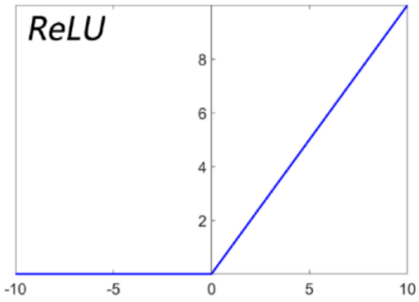


Figure 14 ReLU response

3.1.6.3. Softmax

Softmax, which is the last activation function examined in this section, is used to obtain the posterior probabilities corresponding to the input [99]. To create these probabilities, it maps input values to the probabilities between 0 and 1, where the sum of outputs of the softmax function is 1, in the output layer of the data. The formulation of softmax function is given in (18).

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{18}$$

Sigmoid and tanh functions are used for binary classification, where the softmax function is used for multiclass classification.

3.1. Autoencoders

Autoencoders are commonly used in deep learning applications and play an important role in unsupervised learning [77]. There are different variations of autoencoders in the literature, such as variational autoencoders(VAE), stacked autoencoders, denoising autoencoders [100–103]. Figure 15 shows the basic autoencoder architecture. Autoencoder is an unsupervised machine learning method that learns to copy its input into its output. Autoencoders are one of the most popular models in the deep learning area, usually consisting of two parts: Encoder and Decoder. These two parts are trained together as if they are a single model during the training phase. The difference between the input signal and the output signal is calculated as the error. The generated code is obtained by taking the output of the middle layer. This layer is called the *bottleneck layer* and is used to determine abundance values when used for hyperspectral unmixing.

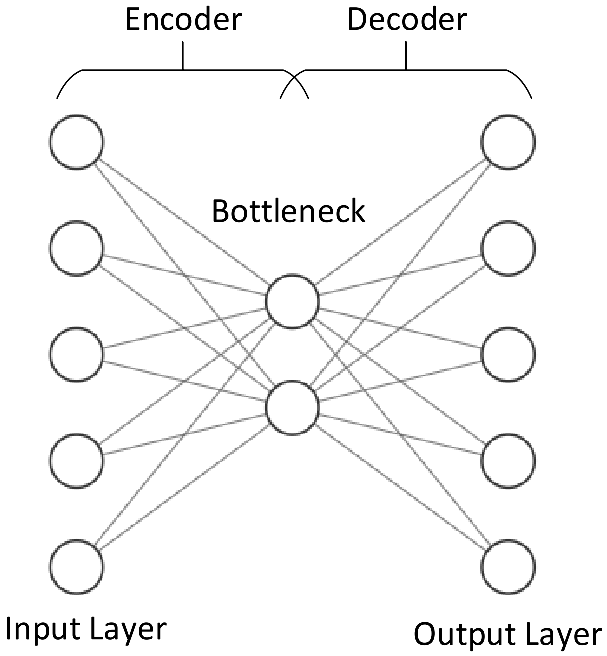


Figure 15 Basic autoencoder architecture

Stacked autoencoders are used in areas such as feature extraction, dimensionality reduction, denoising, image coloring, new data generation. However, traditional applications of autoencoders are dimensionality reduction or feature extraction. Autoencoders have recently been used mainly to learn the structure of the data. The input layer and the output layer have the same number of neurons, and the hidden layer has fewer neurons. It attempts to minimize the difference between input and output. Therefore, autoencoders are unsupervised learning models.

3.2. Descent Based Optimization Methods

Gradient descent is a first-order iterative optimization method to find the minimum of a function. To approach the minimum point, steps proportional to the negative of the gradient of the function at the current point are taken. Although there are various descent-based methods used for deep learning, only the widely used stochastic gradient descent, adaptive gradient, and adaptive movement estimation methods are explained in this section.

Learning rate or step size is a positive scalar value that multiplies the gradient during the iterations. The effect of the learning rate on the search for the optimum point is shown in Figure 16. The case where the learning rate is selected small is shown in Figure 16 (a), and the case where the learning rate is selected large in Figure 16 (b). The learning rate is a critical parameter of the gradient descent methods. A low learning rate may result in slow convergence, whereas too high learning rate may delay convergence due to oscillations around the optimum. Techniques for optimal learning rate selection have been proposed to remedy this problem [104–107].

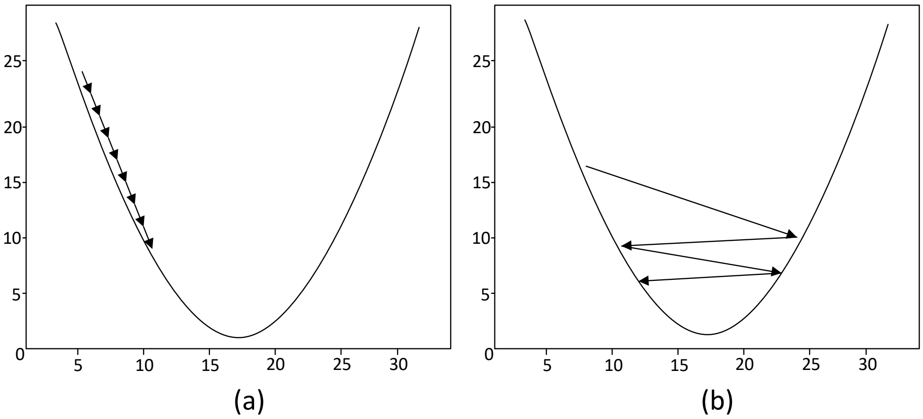


Figure 16 Learning rate effect (a) low learning rate, (b) high learning rate

3.2.1. Stochastic Gradient Descent (SGD)

There are quite different variations of the Stochastic Gradient Descent(SGD) algorithm, which has a significant impact on deep learning studies [108]. The formulation of the SGD is given in (19),

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \,.\, \nabla_{\mathrm{w}}\, D(\mathbf{z}_t, \mathbf{w}_t) \tag{19}$$

where *D* is cost function, α is step size ∇_{w} is gradient of the cost function *D* in each iteration t, z_t is randomly picked example and $\boldsymbol{w}$ is the weights. Although SGD works faster than the previous gradient-based approaches, it is relatively slow compared to the methods suggested after it, and it may converge to a local minimum.

3.2.2. Adaptive Gradients (Adagrad)

Adagrad is a method that eliminates the problems arising from the constant learning coefficient, one of the biggest problems in gradient descent methods, by updating the learning rate at each step. Formulation of Adagrad is given as,

$$g_{\mathrm{t},ij} = \frac{\partial D}{\partial\, \mathrm{w}_{t,ij}}$$

$$G_{ij} = \sum_{t=1}^{T} (g_{t,ij})^2 \tag{20}$$

$$\mathrm{w}_{t+1,ij} = \mathrm{w}_{t,ij} - \frac{n}{\sqrt{G_{ij} + \varepsilon}} \bullet \nabla_{\mathrm{w}}\, D\big(\mathrm{w}_{t,ij}\big)$$

where D is cost function, α is step size ∇_{w} is gradient of the cost function D in iteration t, n is learning rate, G_t is the sum of the squares of the gradients with respect to w_t , ε is a small positive number to avoid division by zero and $\mathrm{w}_{t,ij}$ comprises the ij'ths weights. Adagrad provides faster convergence by using a different learning rate for each parameter.

3.2.3. Adaptive Moment Estimation (Adam)

Adam is proposed for the first-order gradient-based optimization of stochastic objective functions. It is based on adaptive estimates of low-order moments [109]. This method also changes the learning rate in every iteration, like Adagrad.

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1) g_t$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2) g_t^2$$

$$\acute{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t} \tag{21}$$

$$\tilde{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t}$$

$$\mathrm{w}_{t+1} = \mathrm{w}_t - \frac{\alpha}{\sqrt{\tilde{v}_t + \varepsilon}}\ .\acute{m}_t$$

where m_t is estimate of the first moment and v_t is estimate of the second moment of the gradients. β_1 is first momentum term and β_2 is second momentum term, generally set to 0.9 and 0.999 respectively. β values are used to calculate $\acute{m}_t$ and $\tilde{v}_t$which are bias-corrected first and second moment, respectively.

3.3. Hyperspectral Unmixing with Deep Learning

Deep learning uses multilevel neural networks to achieve advances in various applications, including image classification, video analysis, speech recognition, and natural language learning. In recent years, studies have been carried out using deep learning for hyperspectral unmixing.

Although the autoencoder was not used in the first studies [110–112] for hyperspectral unmixing, the majority of the later studies [113-119] work with the encoder structure. One of the first works that did not use encoder structure is A GPU-based algorithm presented by Jiménez et al. as a spatial-spectral preprocessing method for hyperspectral unmixing [110]. In this study, the CPU-GPU differences are examined, and the changes in basic algorithms in terms of time are shown. A later algorithm that does not use the encoder structure and makes an abundance estimation in a supervised manner is presented by Xu et al. [112]. The abundance map was created using given spectral signatures using support vector regression (SVR) [120], recurrent neural networks (RNN) [121], principal component analysis network (PCANet) [122], and stacked autoencoders (SAEs) [123]. Another method that uses a convolutional network instead of encoder infrastructure and performs abundance estimation in a supervised manner is presented by Ozkan and Akar [111].

Although the layers used may differ in both encoder and decoder layers, the mainline of networks used for hyperspectral unmixing is as in Figure 17. Methods for hyperspectral unmixing have gained momentum with the success of the methods using this structure. The input parts of the models may have convolutional or fully connected layers, but the last layers of the algorithms that assume the data with LMM generally have this structure for the last parts. One of the first studies for endmember extraction and abundance estimation is proposed by Yuanchao et al. [114]. Their algorithm has two main steps. In the first stage, a non-negative sparse autoencoder is used to detect endmembers, and then abundance is determined by taking input from an endmember extraction method. Later, similar to a simple autoencoder structure, a 3-layer network using denoising autoencoder for the solution of endmember and abundance estimation problem was presented by Qu, Guo, and Qi [117].

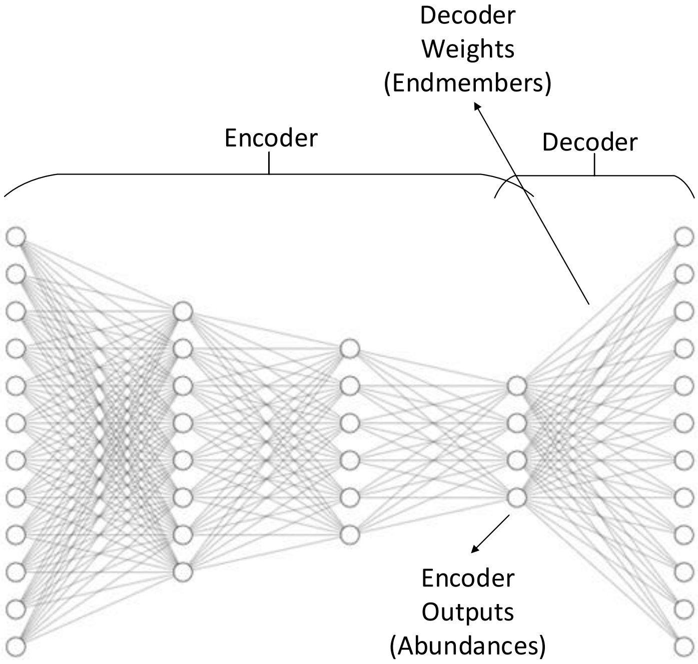


Figure 17 Simple autoencoder based hyperspectral unmixing scheme

More than one non-negative sparse autoencoder is used by Su et al. [113]. This study uses a very similar structure to the encoder-decoder structure given in Figure 17. The main reason for using more than one autoencoder is given as outlier detection. The algorithm performs outlier detection by applying a threshold with the mean and standard deviations of the spectral signals reconstructed in the last two autoencoders. Pallson et al. has made performance comparisons of autoencoders with different cost functions [124]. In this study, detailed information about autoencoder structure and parameters is given, and it is assumed that the hyperspectral data follows the linear mixing model. Another hyperspectral unmixing method presented for linear mixing is the sparsity-constrained deep NMF with total variation (SDNMF-TV) algorithm [125], which uses the Nonnegative Matrix Factorization (NMF) [126].

As mentioned before, different structures can take place in front of the encoder structure. Similarly, a convolutional-based structure has been presented by Zhang et al. [118]. Both pixel-based and cube-based convolutional neural networks are used in this method. The root-mean-square of the abundance angle distance (rmsAAD) is used as the cost function. Stacked nonnegative sparse autoencoders (NNSAEs) offered by Su et al. use more than one NNSAE to eliminate outliers in the data and then estimate abundance [113]. Root mean squared error (RMSE), root error (RE), and spectral angle distance (SAD) are used as cost functions in the algorithm. Another algorithm using both SAE and VAE is the deep autoencoder network (DAEN) [115]. This algorithm was also created with the assumption of LMM. First, spectral signatures are learned using SAEs, and endmember estimation is performed. Later, with VAE-NMF, these signatures are used for abundance estimation. Unlike other algorithms, the unmixing algorithm presented by Yan et al. first moved the data to the wavelet domain and then used an autoencoder for endmember extraction and abundance estimation [119]. Other methods using convolutional neural networks are Deep Convolutional Autoencoder Network (DCAE) [127] and the convolutional neural network autoencoder unmixing (CNNAEU) algorithm [128]. DCAE and CNNAEU methods perform hyperspectral unmixing using 1D convolutional neural networks on the spectral axis. The deep spectral convolution network (DSCN ++) algorithm [129] presented by Ozkan and Akar also uses 1D convolutional neural networks. In the study, spectral diversity, which is the case of different spectral signatures of pure pixels belonging to the same material in the data, is mentioned and the problems that may occur on the real data are given.

Deep learning methods, which have been widely used in recent years, stand out successfully. Although there are models using convolution, we have not encountered any publication that performs hyperspectral decomposition with 3D convolutional encoder structure. In addition, although there are methods recommended for nonlinear approaches, a model using endmember information has not been encountered. Therefore, a model that performs nonlinear hyperspectral unmixing with 3D convolutional encoder structure is proposed.

CHAPTER 4

EXPERIMENTAL SETUP AND DATASET

In this section, the data sets used to test the performance of the proposed methods are explained. Jasper Ridge and Samson Ridge data sets, which are frequently used for algorithms in hyperspectral unmixing in the literature, are used for experiments with real data. Detailed explanations are found by creating synthetic data that is as diverse as possible for hyperspectral abundance estimation and data created by taking into account the spatial information in order to test the model created with a 3D convolutional network. In addition, detailed information is given about the distance metrics used in the algorithm and used in performance comparisons.

4.1. Datasets Used in Experiment

Experiments are performed on both the synthetic data and the real data to evaluate the proposed approaches. Among the proposed methods, the pixel-based abundance estimation method does not use spatial information. It is thought that it would be better to include as many abundance values as possible by using various endmembers in this synthetic data created without using spatial information. On the other hand, the hyperspectral unmixing model using a 3D convolutional neural network involves the use of spatial information. For this reason, synthetic data created to test the performance of the deep learning method should be produced by taking the spatial information into consideration. For these reasons, two different synthetic data sets are created for two different methods. In order to make synthetic hyperspectral data realistic, it is planned to create synthetic data as close to real data as possible by using spectral signals from hyperspectral libraries.

Therefore, in this study, spectral signatures are randomly selected from the spectral library provided by the U.S. Geological Survey (USGS) [130]. These spectral signatures are in the range of 400-2400nm and include 224 bands.

4.1.1. Synthetic data generation for abundance estimation

Figure 18 shows randomly selected endmembers for the three endmembers from the USGS database. The first synthetic data is created for the abundance estimation algorithm and consists of 50x150 pixels. The experiments are performed using different endmembers between three and six. Three different frames are created with three different mixing models with dimensions of 50x50. The first of these frames is generated with LMM, the second with FM, and the last one with PPNM. For example, equation (T) shows an example for the synthetic data generation with 3 endmembers.

$$\boldsymbol{s} = \sum_{r=1}^{p} a_r \boldsymbol{e}_r + b \sum_{i=1}^{p} \sum_{j=1}^{p} a_i a_j \boldsymbol{e}_i \odot \boldsymbol{e}_j + n$$

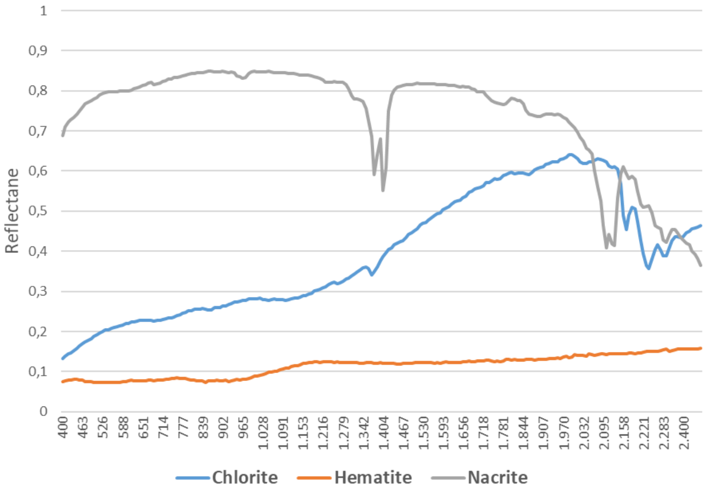
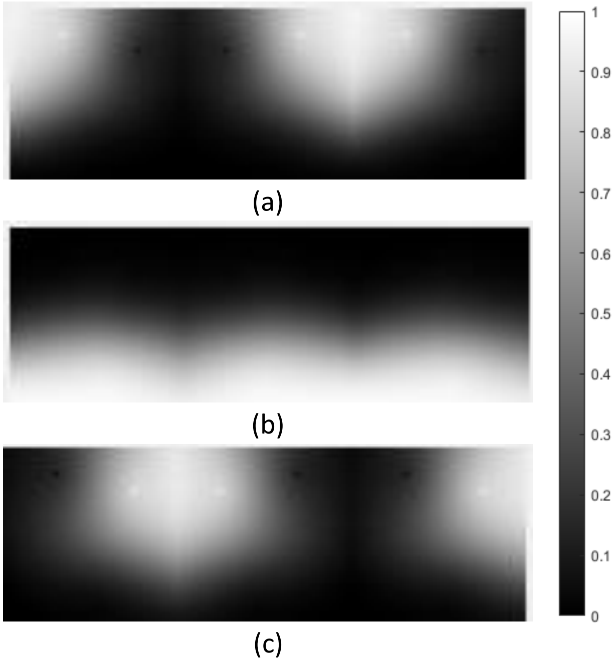


Figure 18 Examples for the selected spectral signatures from the hyperspectral library

A sample ground truth is shown in Figure 19 (a), (b), and (c) for the abundance maps, where the number of endmembers is selected as 3.


(a)

(b)

(c)

Figure 19 A sample Ground Truth for synthetic data for the abundances of three different endmembers utilized for synthetic data generation (a) abundance map for the first endmember, (b) abundance map for the second endmember, and (c) abundance map for the third endmember

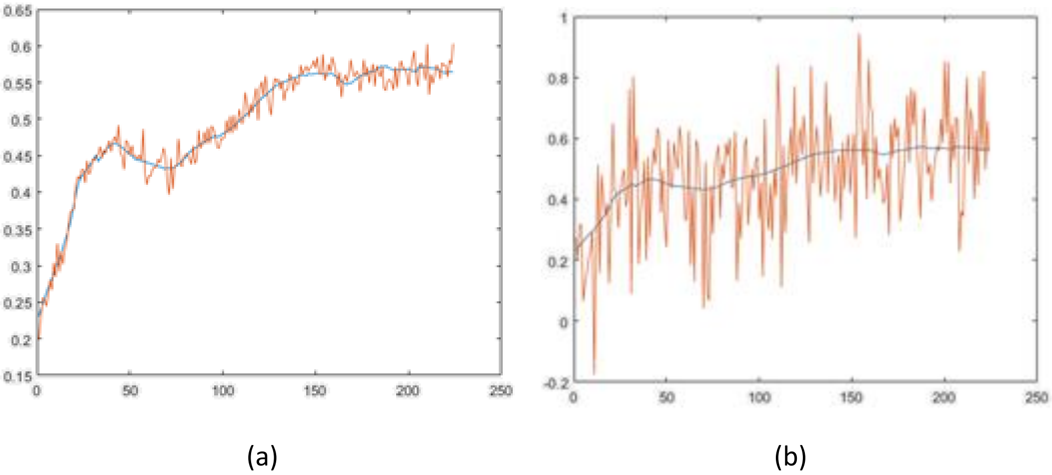
Signal to noise ratio (SNR) is used to determine the given noise levels. The formulation of noise is given as

$$Power_{signal,dB} = 10\ log_{10}\ (Power_{signal})$$

$$Power_{noise,dB} = Power_{signal}\ /\ 10^{SNR/10} \tag{22}$$

$$SNR_{dB} = 10\ log_{10}\ \left(\frac{Power_{signal,dB}}{Power_{noise,dB}}\right).$$

Experiments are conducted with three different cases as a noiseless, medium, and high noise levels, which can be seen in Figure 19. Noise added to a sample spectral signature is shown in Figure 20. The noise levels in the studies are determined as 40db for medium noise and 10db for high-level noise. It is thought that testing the experiments in high and low noise is important to see the effect of the algorithm on noise performance. Therefore, SNR values are chosen as these values.



(a) (b)

Figure 20 Noise levels on spectral signature (a) 40db Noise (b) 10db Noise

4.1.2. Synthetic data generation for deep learning

The deep learning model also benefits from the information brought by the spatial information, unlike models that make blind unmix, using 3D convolutional filters. Therefore, in order to test this model, it is necessary to generate the abundance values by using spatial information instead of generating with random abundance values as in the abundance method. Four different endmembers are used for synthetic data generated using spatial information.

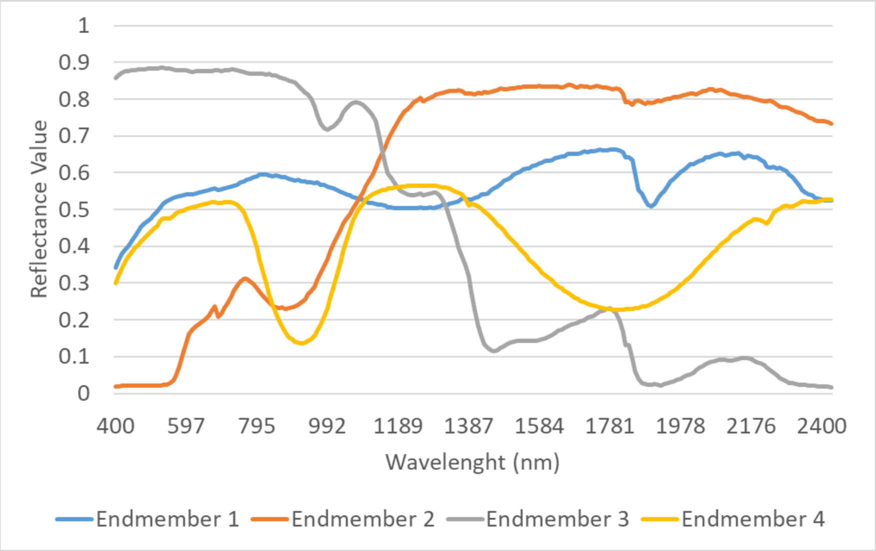
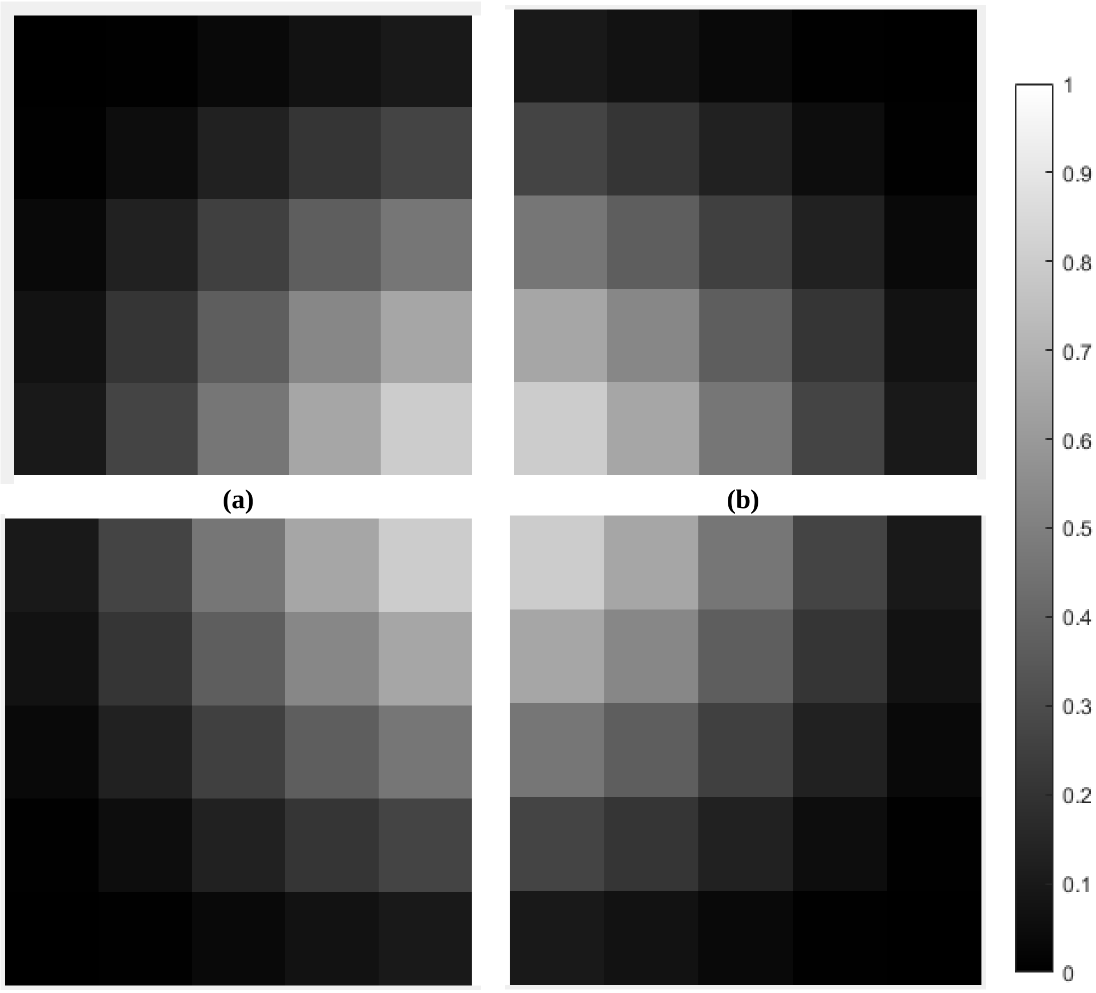


Figure 21 Spectral signature of the selected endmembers for deep learning method

The selected endmembers for these experiments are shown in Figure 21. With these endmembers, 16 frames consisting of 25x25 pixels are created. Random mixing model are chosen for each frame. The mixing models chosen for this data are determined as LMM, FM and PPNM with random b values for one frame. The main reason for this is the assumption that reflections within a segment can be varied, as can be in real data. The abundance maps consisting of these 25 pieces combined are shown in Figure 22. Different abundance values are used for each material in each frame to increase diversity.



(a) **(b)**

(c) **(d)**

Figure 22 Abundance maps for synthetic data. (a) Abundance map for endmember 1, (b) Abundance map for endmember 2, (c) Abundance map for endmember 3 and (d) Abundance map for endmember 4

4.1.3. Utilized real data for experiments

Two different hyperspectral datasets, which are widely used in the literature to measure hyperspectral unmixing and abundance estimation performance, are used in this study. RGB images and ground truth information of these data are shown in Figure 23. The first set, Samson Ridge, consists of 156 bands covering 400 nm to

900 nm. There are three kinds of materials in the data, namely soil, tree, and water, shown in Figure 23 (b), Figure 23 (c), Figure 23 (d). The second set, Jasper Ridge, consists of 224 bands covering 380 nm to 2400 nm. Due to dense water vapor and atmospheric effects, the channels with the number from 1-5, 108-112,154-166, and 220-224 are removed, and the remaining 196 bands are used. This data includes four different materials, which are Tree (Figure 23-f), Water (Figure 23-g), Soil (Figure 23-h), and Road (Figure 23-i). A more detailed description of these datasets can be found in [131].

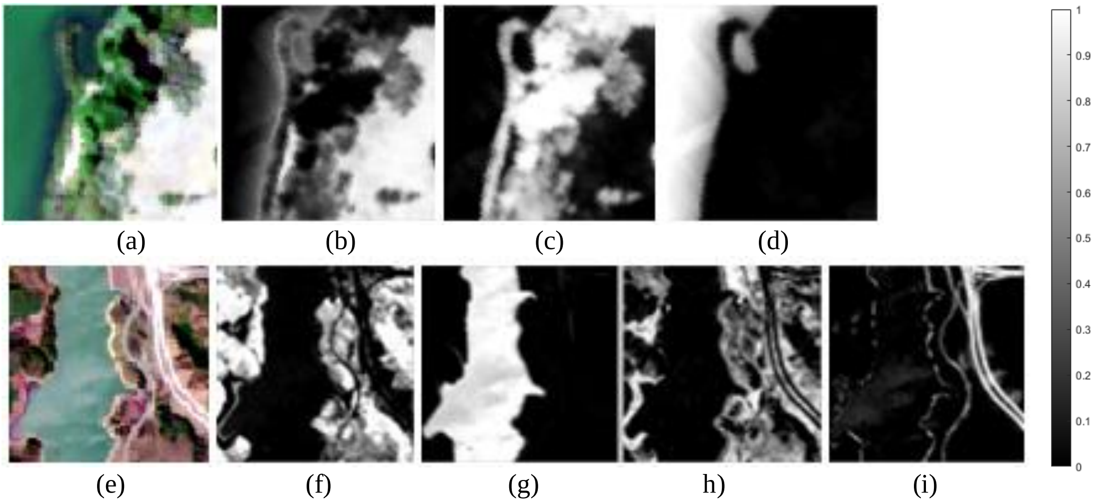

(a) (b) (c) (d)
(e) (f) (g) h) (i)

Figure 23 RGB image of Samson Ridge (a), Ground truth classes (b) Soil, (c) Tree and (d) Water and RGB image of Jasper Ridge (e) and Ground Truth Classes (f) Tree, (g) Water, (h) Soil and (i) Road

4.2. Distance Metrics

This section describes the distance metrics widely used in the literature and used in the later parts of the study. These metrics are mean absolute error, mean square error, root mean square error, spectral angle mapper, and spectral information divergence. The expressions of these methods are given as follows. In these expressions, $\hat{\boldsymbol{x}}_{\boldsymbol{p}}$ is calculated value, $\boldsymbol{x}_{\boldsymbol{p}}$ is the target value, and P is the number of bands.

L1-Norm

L1-norm, also known as the Manhattan distance, is calculated as the absolute difference of two vectors. L1-Norm is given in the expression (23).

$$Distance_{L1}(\hat{\boldsymbol{x}}_p, \boldsymbol{x}_p) = \sum_{i=1}^{P} |\, (\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)\, | \tag{23}$$

L1-Norm has been used both to determine the error value in the optimization process in detecting the abundance and as a cost function in the deep learning algorithm.

L2-Norm

The L2 norm, also known as the Euclidian norm, is given as the square root of the sum of squares error between two vectors. L2-norm is used to determine the error value in the optimization process. The expression for L2-norm is given in (24).

$$Distance_{L2}(\hat{\boldsymbol{x}}_p, \boldsymbol{x}_p) = \sqrt{\sum_{i=1}^{P} (\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)^2} \tag{24}$$

Mean Absolute Error (MAE)

The expression of mean absolute error (MAE) is given in (25). MAE is used as a cost function in deep learning.

$$\text{Mean Absolute Error(MAE)} = \frac{1}{P} \sum_{i=1}^{P} \big|| \hat{\boldsymbol{x}}_i - \boldsymbol{x}_i |\big| \tag{25}$$

Mean Squared Error (MSE)

The expression of mean squared error (MSE) is given in (26). MSE is used as a cost function in deep learning. It is also used to measure the difference between the abundance values predicted in performance evaluation and the ground truth values.

$$\text{Mean Squared Error } = \frac{1}{P} \sum_{i=1}^{P} (\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)^2 \tag{26}$$

Root Mean Square Error (RMSE)

The expression of root means squared error is given in (27). RMSE is used to measure the difference between the abundance values predicted in performance evaluation and the ground truth values.

$$\text{Root Mean Square Error} = \sqrt{\frac{1}{P} \sum_{i=1}^{P} (\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)^2} \tag{27}$$

Spectral Angle Mapper (SAM)

Spectral angle mapper (SAM), presented by Kruse et al. [132], obtains the angle between two different spectrums in terms of radians. Its formulation is given as

$$SAM = \frac{1}{N} \sum_{i=1}^{N} \arccos\left(\frac{\langle \boldsymbol{x}_i \, , \hat{\boldsymbol{x}}_i \rangle}{||\boldsymbol{x}_i||_2 ||\hat{\boldsymbol{x}}_i||_2} \right). \tag{28}$$

In the expression N is the number of the pixels. SAM is a widely used distance measurement method in hyperspectral image processing. It is a very common problem that the spectral signatures of the same material are at different levels due to the light differences. Even if SAM spectral signatures have different reflectance values, they are quite successful in determining the similarities compared to other distance measurement methods.

Spectral Information Divergence (SID)

The SID algorithm was presented by Chang to measure the spectral similarity between two spectra of pixels [133]. The formulation of SID is given as

$$SID = \frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{B} p_n \log\left(\frac{p_n}{q_n}\right) + \sum_{n=1}^{B} q_n \log\left(\frac{q}{p_n}\right) \tag{29}$$

where, $p_n = \frac{x_{i,n}}{\sum_{k=1}^{M} x_{i,k}}$ and $q_n = \frac{\hat{x}_{i,n}}{\sum_{k=1}^{M} \hat{x}_{i,k}}$.

In these equations, *N* equals the number of pixels in the data. SID also works successfully in determining the similarities of spectral signatures such as SAM from methods that are calculated directly with the difference between spectral signatures such as MSE, L1-Norm, and L2-Norm. Therefore, these methods are expected to be less affected by spectral signature changes due to light changes on real data.

CHAPTER 5

PROPOSED COARSE TO FINE ABUNDANCE ESTIMATION FOR HIGHLY MIXED HYPERSPECTRAL DATA

Hyperspectral data can be formed with more than one mixing model due to its nature. While the incident light from some pixels can be directly reflected in the scene, the reflected light from some other pixels can be formed with the interactions between different layers and materials. Mixing models presented as a solution for such problems in the literature are generally based on a single model assumption. Although these algorithms provide high performance in synthetic data created with their own assumptions, they cannot provide the same performance in real data. Therefore, it is necessary to use more than one model mixing to unmix real data.

In this section, an optimization-based algorithm has been developed that can be used in the case of multiple mixing scenarios. Multi-model minimization methods can be implemented in two different ways. The first way is to minimize each model separately and then choose the minimum among the models. The second is to define the cost function to take the minimum of all models in each iteration. The model presented in this section determines the abundance with the multi-mixing model in a single optimization process. The application of multiple models in a single optimization process can cause high costs due to parameter redundancy. With the coarse to fine approach, this cost is reduced as much as possible, at least until a certain convergence.

5.1. Proposed Method

The optimization problem for the abundance estimation can be described as finding the abundance values for endmembers, which minimizes a distance metric *D* between the estimated pixel and the target pixel. Such an optimization problem is given as,

$$\boldsymbol{a}_* = \underset{\boldsymbol{a}}{\arg\min}\,(D(\boldsymbol{s},\boldsymbol{t})) \tag{30}$$

where $\boldsymbol{a}_*$ is an element of R^p is the optimal abundance vector, $\boldsymbol{t}$ is the actual spectral signature of the target pixel, and $\boldsymbol{s}$ is the estimated spectral signature of the target pixel. Given a mixture model *M* as described with one of the models in (1), (3), (6), (7) or (8), the optimization problem can be further expressed as,

$$\boldsymbol{a}_* = \underset{\boldsymbol{a}}{\arg\min}\,(D(\,M(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t})) \tag{31}$$

where $\boldsymbol{a}$ are the abundances and $\boldsymbol{E}$ is a matrix of size *Lxp*, which is composed of corresponding endmembers, $\boldsymbol{E}=[\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots.,\boldsymbol{e}_p]$.

Such a problem is solved with different algorithms in the unmixing literature. In this section, these algorithms, namely sequential quadratic programming [50–52], pattern search [53], [54], genetic algorithms [55–58] and simulated annealing [59–61], are briefly described. While SQP, GA, and SA are previously used in existing studies, PS is newly adapted to the abundance estimation problem.

Hyperspectral data can be formed with more than one mixing model due to its nature. While the incident light from some pixels can be directly reflected in the scene, the reflected light from some other pixels can be formed with the interactions between different layers and materials. Therefore, for each pixel, an optimization-based algorithm, which can be used in the case of a multi-mixing scenario, has been developed in this section. The main underlying idea for such a solution is the blind abundance estimation according to the multiple mixture models with the assumption that there might be pixels with more than one mixture model within the image. In such a multi mixture model, the problem of abundance estimation can be redefined as the minimization of the cost function,

$$\boldsymbol{a}_* = \underset{\boldsymbol{a}}{\operatorname{argmin}} \left\{ \min \begin{pmatrix} D(M_1(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}), \\ D(M_2(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}), \\ \cdot \\ \cdot \\ D(M_n(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}) \end{pmatrix} \right\} \tag{32}$$

where M_1, M_2, and M_n represent different mixing models to sufficiently address different interactions inside and between the materials in the investigated scene.

The first analysis in such a redefined optimization problem is to investigate the behavior of the cost function for each mixing model as a function of abundances. Figure 24 illustrates a simple example for this purpose over two endmembers. The SAM metric is utilized as the distance between the generated pixel spectrum with two endmembers by using a selected mixing model and the reconstructed pixel with respect to the different abundances by using one of the three mentioned models. The cost function defined as the minimum of the three costs for three mixing models in (12) is also plotted with a dashed line in the figure as a function of abundance, a. Note that the abundance for the other endmember is ($1-a$) for the case of two endmembers, and therefore, the graphs are one-dimensional.

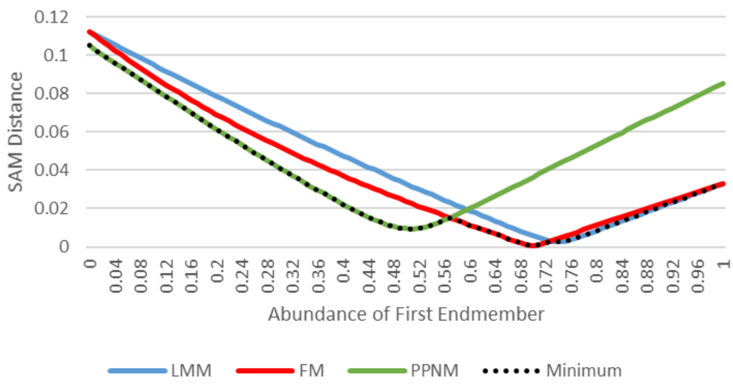


Figure 24 An example for the change of SAM distance as a function of abundance value for two different endmembers

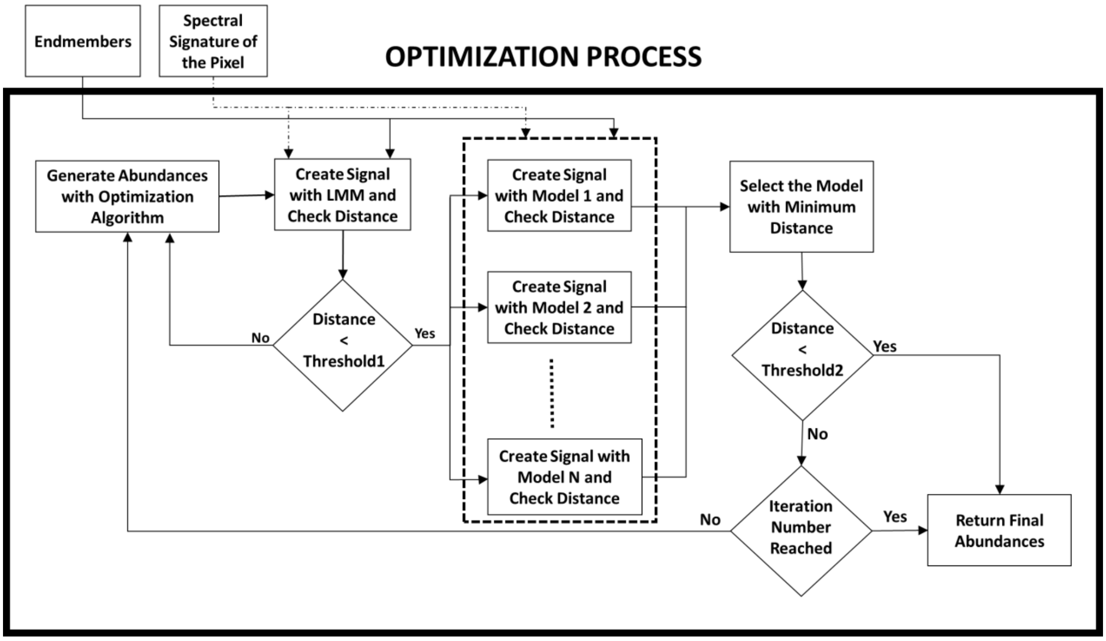

Endmembers
Spectral Signature of the Pixel
OPTIMIZATION PROCESS
Generate Abundances with Optimization Algorithm
Create Signal with LMM and Check Distance
Distance < Threshold1
No
Yes
Create Signal with Model 1 and Check Distance
Create Signal with Model 2 and Check Distance
Create Signal with Model N and Check Distance
Select the Model with Minimum Distance
Distance < Threshold2
Yes
No
Iteration Number Reached
No
Yes
Return Final Abundances

Figure 25 Flowchart of the proposed coarse to fine methodology for abundance estimation

As the first observation in Figure 24, the minima for all the mixing models are at different locations. Second, the cost function, defined as the minimum of the three mixing models and shown with the dashed line, indicates three local minima with convex characteristics, which makes the gradient descent based global search algorithms infeasible for such a problem. As the final observation, the minima for all the mixing models occur in the vicinity of each other. This implies that the search can be done for one model until the abundance estimate reaches a predefined vicinity and then continues to the minimum of all mixture models. Considering that the LMM model is the baseline coarse component in the given mixing models in Section II.A, the threshold to define such a vicinity is experimentally selected as 0.1 in terms of the SAM distance and the cost function is redefined as,

$$\boldsymbol{a}_* = \begin{cases} \arg\min\limits_{\boldsymbol{a}} \big(D(\,M_1(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t})\big), & D(\,M_1(\boldsymbol{a},\boldsymbol{E}),t) \geq th \\ \operatorname*{argmin}\limits_{\boldsymbol{a}} \begin{pmatrix} D(\,M_1(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}), \\ D(\,M_2(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}), \\ \cdot \\ \cdot \\ \cdot \\ D(\,M_n(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}) \end{pmatrix}, & D(\,M_1(\boldsymbol{a},\boldsymbol{E}),\boldsymbol{t}) < th \end{cases} \tag{33}$$

Instead of updating all models simultaneously, it is also possible to run minimization for every model individually. We observed the proposed approach in (13) provides advantages in terms of complexity. In this study, we selected the number of mixing models as three and determined these models as FM, LMM, and PPNM, as we found these models sufficient to describe possible reflections in real data. More specifically, the three models are defined as in (1), (6), and (8), respectively.

Given these observations, the proposed coarse to fine abundance estimation method first takes the target pixel spectra and the endmembers in the scene as inputs and continues with the following main stages in accordance with Figure 25:

- For a given hyperspectral pixel, the abundance values are initialized and updated with respect to the given cost function in (33) with the selected optimization algorithm.
- For each iteration of the optimization algorithm, the spectrum for the given pixel is reconstructed according to LMM with the estimated abundance values and the given endmembers, and the distance between the pixel spectrum and the reconstructed spectrum is computed.
- If the computed distance is higher than the threshold, the optimization algorithm continues to the next iteration.
- If the distance is smaller than the threshold, then the algorithm also computes the distances for FM and PPNM and selects the minimum distance from all mixture models, and then continues to the next iteration with the new abundance values.
- The algorithm terminates when the distance is sufficiently small or the maximum number of iterations is reached.

The distance, *D*, between the pixel spectrum and the reconstructed spectrum is selected as L1-Norm, L2-Norm, and SAM distance for the experiments. The L1-Norm, L2-Norm, and SAM distances. The effect of the thresholding is particularly investigated for the SAM distance by both implementing the proposed method with the given cost functions in (32) and (33). The thresholding in the cost function is not applied for L1-Norm, and L2-Norm distances as the threshold indicate high variability in these distances. In Section V, the results of the experiments with different optimization algorithms are presented and compared both for synthetic and real data.

5.2. Experimental Results and Discussion

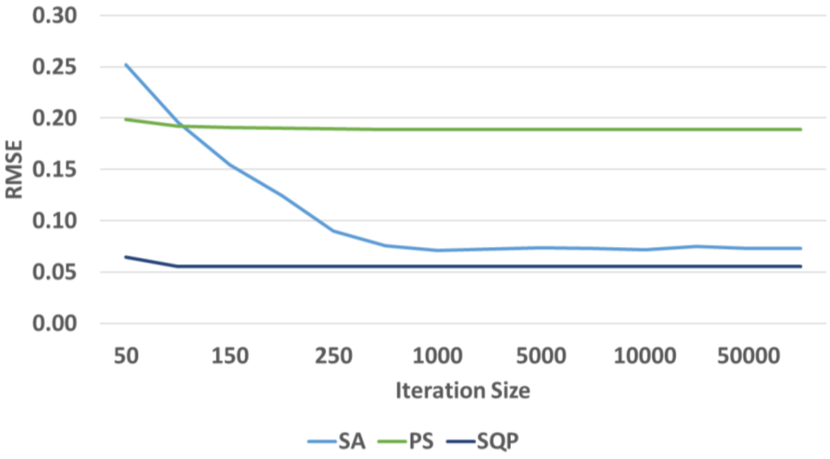
Considering the different aspects of the proposed abundance estimation method, such as the utilized mixing model, optimization algorithm, distance metrics and design parameters in all the mentioned cases, the experiments are organized as follows for a more compact presentation:

- First, the effects of the design parameters in the utilized optimization algorithms, namely, SA, PS, SQP, and GA, to the estimation performances are analyzed in Section V. B. Based on this analysis, the design parameters for a practical implementation and a fair comparison are selected in this subsection.
- Afterward, the effect of distance metrics, namely L2-norm, L1-norm, and SAM on the algorithm performances for abundance estimation are examined in Section V. B.
- As the baseline comparison, the proposed coarse to fine approach based on multi mixture model is compared with the direct search in Section V. C.
- In the next subsection, the performances of the proposed coarse to fine approach with different optimization algorithms are investigated. The ultimate method with the selected optimization algorithm is revealed.
- In Section V. E, the comparison of the proposed coarse to fine approach is compared with the baseline methods in the literature.
- Finally, in Section V. F, the performance of the proposed method on real data is presented.

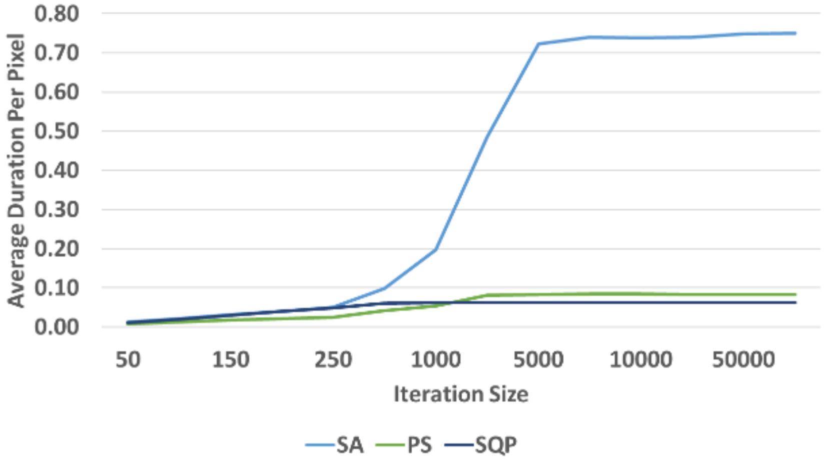
The experiments in all the sections are performed over five realizations by randomly generating five different hyperspectral data cubes with different endmember sets and estimation is presented as the average performance of all pixels over five different realizations.

5.2.1. Selection of Design Parameters

As the performance of the algorithms varies considerably with the design parameters, the first tests are performed to determine the optimized parameters for different optimization methods. In particular, while the tests on SQP, SA, and PS are conducted by changing the number of iterations, the design parameters for the GA are selected as generation and population size.



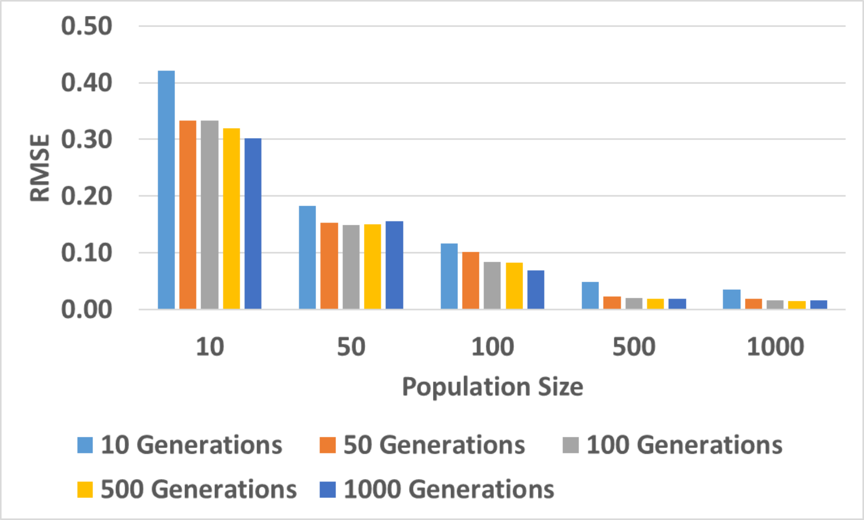
(a)



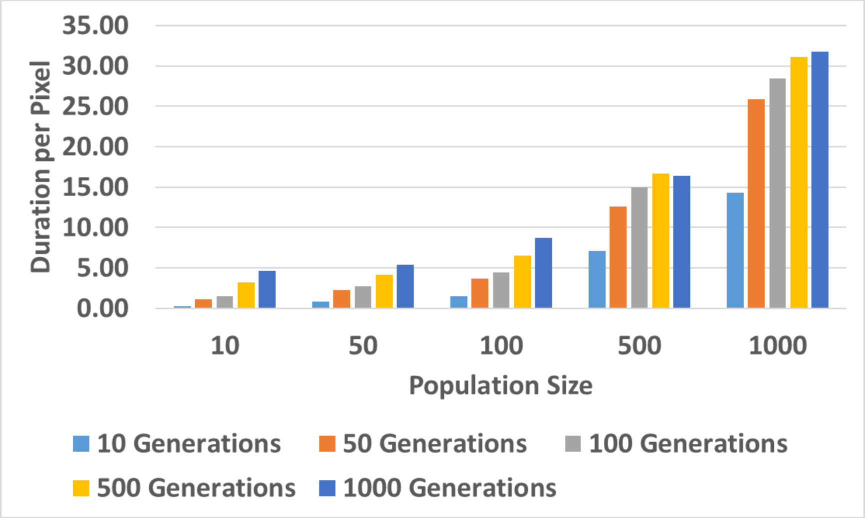
(b)

Figure 26 (a) Change of error rates of SA, PS and SQP algorithms for different iteration numbers (b) Average abundance estimation time per pixel changes of SA, PS and SQP algorithms for three different endmembers with respect to iteration number

Figure 26 (a) illustrates the performances in terms of RMSE with respect to the number of iterations for the proposed SA, PS, and SQP based coarse to fine algorithms. It should be noted that the duration in Figure 26 is given for three endmembers and can vary with the number of endmembers and noise. PS and SQP algorithms are observed to converge to the best RMSE after 100 iterations. On the other hand, the SA algorithm does not indicate any significant improvement after 1000 iterations. Figure 26 gives the duration of the algorithms with respect to the number of iterations. SA algorithm, which continues its performance improvement up to 1000 iterations as observed in Figure 26, shifts to the fine search by decreasing the damping coefficient after 1000 iterations. Then, it continues to fine search up to 5000 iterations and terminates the algorithm as there is no improvement afterward. Accordingly, the duration of the SA algorithm significantly increases after 500 iterations as it shifts to the fine search at those levels, as observed in Figure 26 (b). As a result of the experiments, the number of iterations is selected as 10.000 for fair performance evaluation. The performance differences observed in Figure 26 will be interpreted in the following sections with respect to the optimization algorithm.



(a)



(b)

Figure 27 (a) RMSE of GA with respect to population and generation size (b) Detection time per pixel change with respect to population and generation size for GA

Figure 27 (a) illustrates the changes in RMSE values for the genetic algorithm concerning the population and generation size. As the population size increases, there is a significant decrease in the RMSE values until the population size of 500.

However, the performance for the population size of 1000 is approximately in the same range as the case of 500. On the other hand, the performance for the generation size of 10 is significantly different than the other sizes of 50, 100, 500, and 1000.

Figure 27 (b) gives the durations for GA according to population and generation sizes. The duration of the algorithm has been observed to increase geometrically depending on both the population and generation sizes. As the duration of the average population can take up to 15 seconds per pixel, the usage of the GA seems not very tolerable for practical applications. However, the performance of the GA is also compared with the other algorithms for the sake of completeness. Given the performances in Figure 27, both the population and generation sizes are selected as 500 in the experiments.

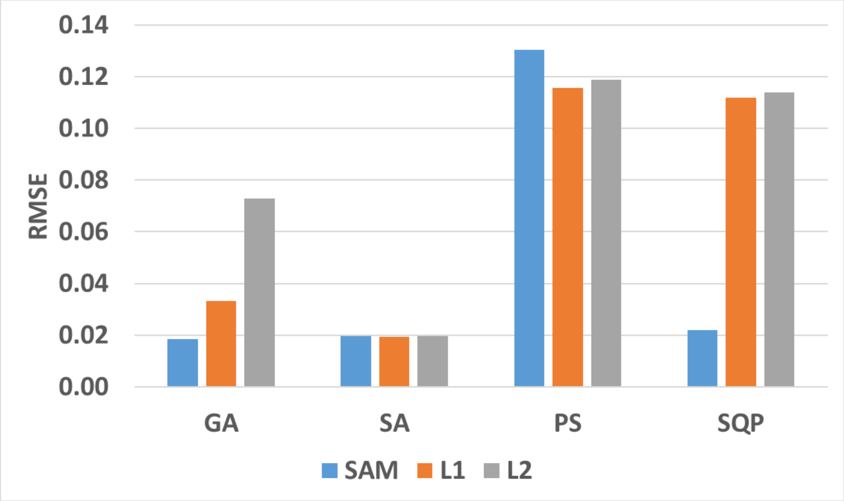


Figure 28 Comparison of GA, SA, PS, and SQP algorithms with SAM, L1 and L2-Norm as RMSE

5.2.2. Selection of Distance Metric

The distance metrics utilized as the cost function in the proposed optimization algorithms are the other aspect of the comparisons in the experiments. In accordance with the common literature, these metrics are selected as L1-Norm, L2-Norm, and SAM in the experiments, while fixing the parameters for each optimization method as given in the previous section.

Table 1 Durations of distance metrics for GA, SA, PS, and SQP algorithms

	SAM	L1-Norm	L2-Norm
GA	15,87	20,46	20,86
SA	0,61	0,68	0,72
PS	0,03	0,04	0,05
SQP	0,01	0,02	0,01

Figure 28 indicates the performances in terms of the RMSE between the estimated and original abundances for each of the metric and optimization methods. While L1-Norm reveals better performances for the PS algorithm, the superiority of SAM is quite distinguishable for the other optimization methods, including GA and SQP. The reason that the SAM metric works well in other optimization algorithms can be explained by the fact that SAM shows an observable change in small abundance changes. However, due to the algorithm structure, a different result is obtained because GA approaches all possibilities equally in the early stages. Because of its better performance, the SAM metric is selected as the distance metric of the proposed optimization algorithms in the rest of the experiments. As also illustrated in Table 1, SAM also provides shorter durations for the implementation among three distance metrics.

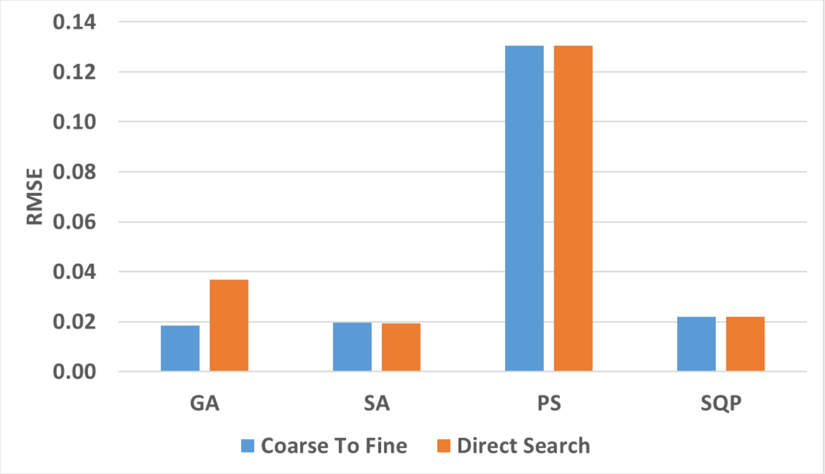


Figure 29 Comparison of coarse to fine and direct searches

5.2.3. Comparison of the Coarse to Fine and Direct Searches

The third experiment is the validation of the proposed coarse to fine search with the direct search, where the minimum is defined as the minimum of all models described in (12). Figure 29 illustrates the performances of both approaches for different optimization methods. The coarse to fine method improves the performances with the same parameters compared to the direct approach for the GA algorithm. On the other hand, the performances of the two approaches are almost at the same level for SA, PS, and SQP algorithms.

The use of coarse to fine method, on the other hand, provides significant advantages compared to the direct search in terms of complexity. Table 2 gives the average time for the estimation of abundances for one pixel for the compared four methods. While the time reduction is about 10 % for SA, it is about 28 %, 40 %, and 50 % for SA, PS, and SQP, respectively. While the GA gives the minimum results in the experiments, as indicated Table 2, its duration is quite high, which makes its practical usage infeasible.

Table 2 Duration of the optimization algorithms per pixel (in seconds) for coarse to fine and direct searches

	COARSE TO FINE APPROACH	DIRECT SEARCH APPROACH
GA	15,87	21,62
SA	0,61	0,68
PS	0,03	0,05
SQP	0,01	0,02

5.2.4. Coarse to Fine Approaches with Different Number of Endmembers

Given the distance metric of SAM and the selected coarse to fine approach, the fourth aspect of the experiments is to compare the performances with respect to different optimization algorithms. Figure 30 gives the results of the optimization algorithm for different numbers of endmembers. It can be seen that the SQP algorithm is least affected by the number of endmembers, and its performance is comparable to that of the GA algorithm with three endmembers.

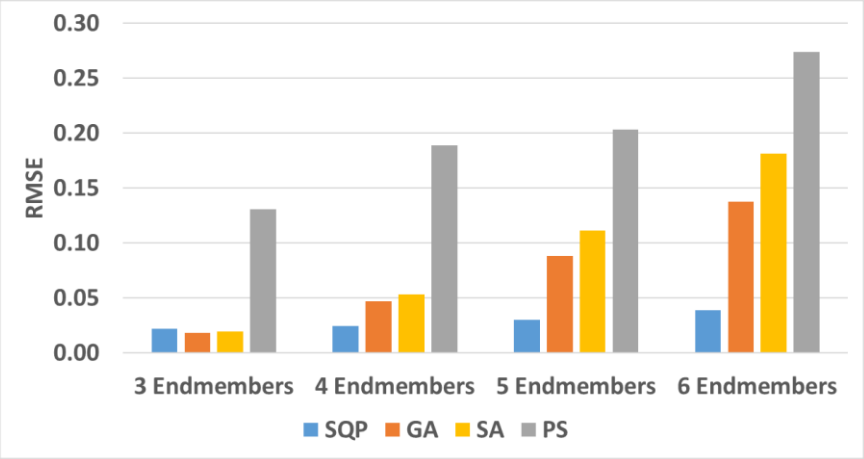
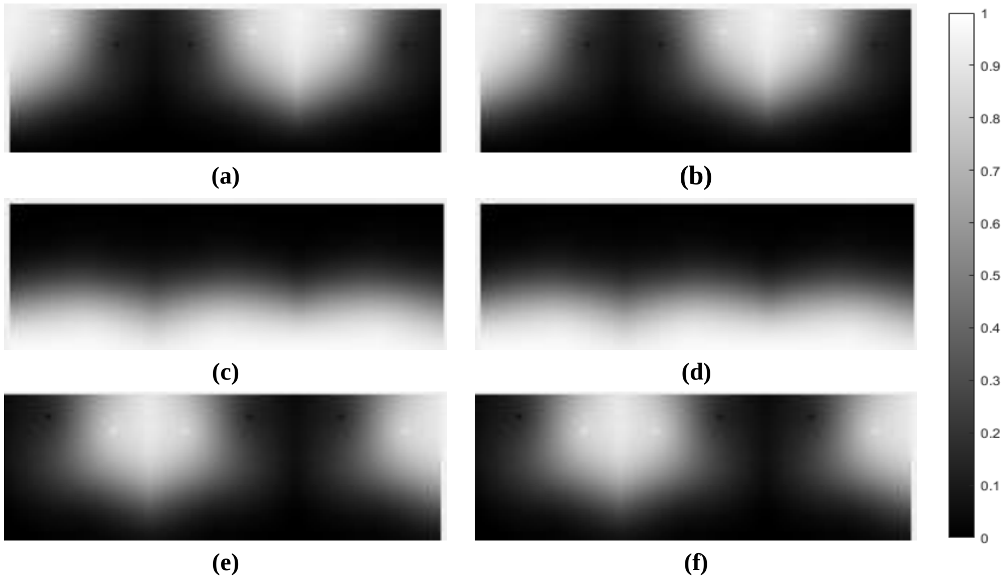


Figure 30 Comparison of optimization algorithms for different numbers of endmembers

It is observed that the algorithm, which indicates the most decreasing performance according to the number of the endmembers, is GA in the performed algorithms. This performance can still be improved by accordingly increasing the population and generation sizes in the GA algorithm. However, this increase comes with the expense of impractical implementation durations, as discussed in Section C. On the other hand, SQP is found to be quite successful in all number of endmembers among all the optimization methods. While the performances of the SA and PS algorithms for unconstrained optimization problems are quite validated in the literature [134], their performance for the case of constrained optimization for the handled abundance estimation problem is not as good as the SQP, which is specially tailored as a constrained optimization problem. Due to the mentioned performances, the SQP algorithm is further selected in the rest of the experiments.

Figure 33 illustrates an example over synthetic data for the estimated abundances along with the ground truths for the ultimate coarse to fine algorithm with SQP. Note that the synthetic data is generated as explained in Section IV, where the number of endmembers is selected as 3. The estimated abundances compared with the ground truth are quite similar for all of the endmembers, resulting in average error, which is smaller than 0.02 per pixel.



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 31 Sample ground truth and results for SQP for synthetic data (a-c-e the ground truths for endmember1, endmember2 and endmember3 respectively, b-d-f the results for corresponding endmembers)

5.2.5. Comparison of the Proposed Method with the Baseline Methods in the Literature

The methods of LMM, GBM, MLM, and PPNM are widely used in the literature for abundance estimation. Note that the mentioned abbreviations correspond to the proposed solutions in the mentioned references [17], [20], [23], [24] rather than the mixture model in this section in accordance with the convention in the literature.

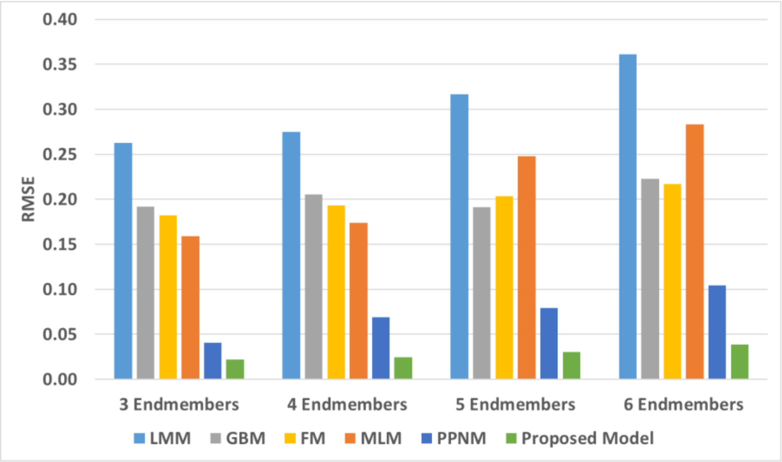


Figure 32 The comparison of the proposed method with the state of the art algorithms for different endmembers

The comparisons of these methods with the proposed approach are illustrated in Figure 32 for the generated synthetic data with a different number of endmembers. The most successful algorithm among the compared methods is found as PPNM. However, the proposed SAM and SQP based coarse to fine approach gives significantly smaller RMSE values for all the endmembers compared to the PPNM.

The given results are extended for the case of additive noise in Figure 33. The noise is added to the pixel spectra for two different levels of 10 dB and 40 dB, representing an average and extreme case. As expected, the increase in noise inversely affects the performance of the algorithms. It has been observed that the performance of the MLM algorithm is more affected by noise than other algorithms. As in the noiseless situation given in Figure 33, the increase in the number of endmembers negatively affects the performance. On the other hand, the method which is least affected by the noise is observed to be the FM. The proposed algorithm shows the best performance among the compared methods for both noise levels.

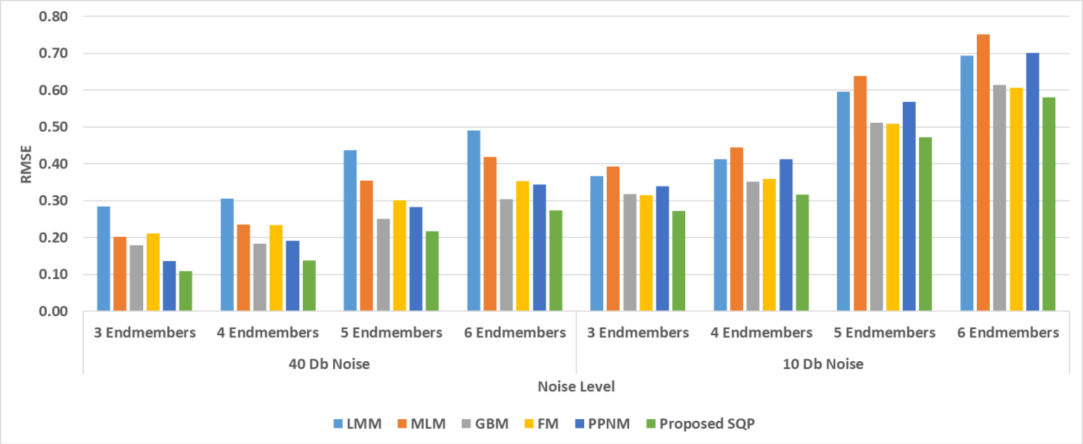


Figure 33 Noise effects on the algorithms. Results for 40db and 10db noisy data with different endmembers

As an example, Figure 33 indicates the spectra of a sample pixel, its noisy versions, and the constructed pixel spectra with the proposed approach. The reconstructed signal depicted in gray almost completely overlaps with the original signal in orange, even at high noise levels.

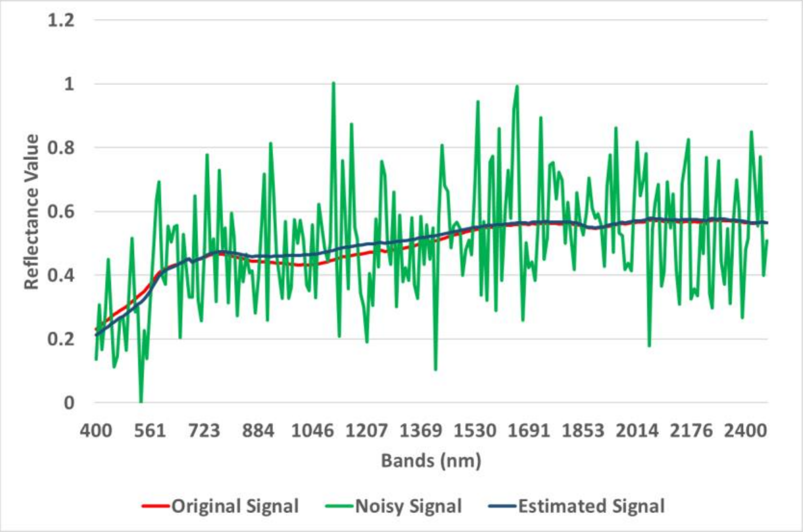


Figure 33 Sample signature with and without 10db noise and estimated signature

Table 3 Performance of the algorithms for the data with different mixing models (LMM, FM and PPNM)

	Data with LMM				Data with Fan Model				Data with PPNM			
	3 EMs	4 EMs	5 EMs	6 EMs	3 EMs	4 EMs	5 EMs	6 EMs	3 EMs	4 EMs	5 EMs	6 EMs
LMM	0,000	0,000	0,000	0,000	0,116	0,121	0,177	0,200	0,235	0,240	0,262	0,299
MLM	0,000	0,000	0,000	0,000	0,065	0,079	0,121	0,132	0,145	0,155	0,215	0,249
GBM	0,000	0,000	0,000	0,000	0,009	0,007	0,008	0,006	0,187	0,145	0,191	0,222
FM	0,098	0,100	0,135	0,144	0,002	0,000	0,000	0,002	0,139	0,137	0,149	0,159
PPNM	0,011	0,012	0,015	0,023	0,033	0,043	0,045	0,067	0,001	0,007	0,017	0,032
Proposed SQP	0,002	0,004	0,009	0,016	0,021	0,021	0,025	0,029	0,005	0,009	0,011	0,017

While previous results are cumulatively given over highly mixed data containing three different models, Table 3 shows the results separately generated with only one model, as another aspect of the comparison. For instance, the presented results on the left side of the table show the performance of the proposed and baseline methods for the data generated with only LMM. It has been observed that LMM and GBM algorithms are very successful in synthetic data created with LMM. Although PPNM and the proposed algorithm also give low error rates, LMM and GBM are observed to be more successful in the data created with LMM.

As a second case where the data is generated with the Fan model, the performance of other algorithms is decreased, and the FM has outperformed. It is also observed that GBM has also achieved successful results on this data. Finally, PPNM and the proposed algorithm have been observed to outperform the other algorithms on synthetic data created with PPNM, as expected. However, the performances of the algorithms decrease when pixels with different mixing models exist in the utilized data. In addition, it is observed that the performances of the algorithms generally decrease with the increase in the number of endmembers.

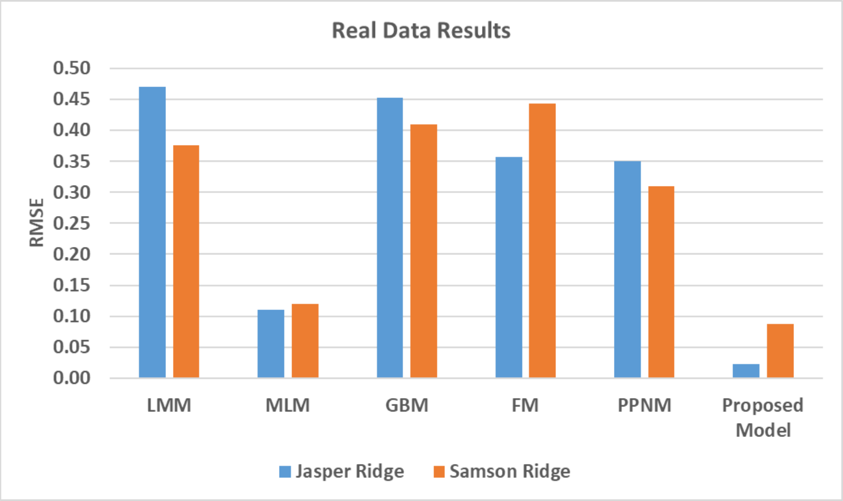


Figure 34 Abundance estimation error for Samson Ridge and Jasper Ridge data sets

As expected, the overall results reveal better performances when the utilized algorithm have the same mixing model with the generated data or covers the mixing model that can be obtained with parameter changes. For instance, the performance of GBM is satisfactory for data generated with LMM, as GBM turns into LMM when the coefficient for the inner product is set to 0 in (7). The experiments highlight that an algorithm based only one mixing model can fail when the actual mixing model of the data is different.

5.2.6. Comparisons with the Baseline Literature on Real Data

As the final experiment, Figure 34 shows the performance of the algorithms on the real data: Samson Ridge, and Jasper Ridge. All algorithms, which are operating quite smoothly on synthetic data, reveal decreased performances on real data. The MLM algorithm's performance, on the other hand, is relatively better on real data compared to other algorithms in the literature. The performance of the proposed algorithm is found to be quite successful among all the algorithms due to its searching methodology for all the possible mixing models and the utilized distance metric.

Figure 35 and Figure 36 shows the output of the SQP algorithm for Jasper Ridge and Samson Ridge data. It is seen that the obtained results are quite similar to the ground truths for both images. When the results are analyzed in detail, it is also observed that, in addition to the use of multiple mixing models, selecting the distance unit as SAM significantly increases the performance. Note that the pixels with LMM, FM, and PPNM are coded with black, gray, and white colors, respectively. In comparison with the presented RGB images, the PPNM model is more common in trees and water regions, where the incident light is more likely to be reflected after the interactions with different layers. The LMM model is mostly seen in soil, rocks, and asphalt regions where the materials are mostly uniform and not expected to reflect with each other.

Finally, the FM model is seen more frequently in sloping areas or inclined regions conforming to the assumption that the material does not interact on its own. The proposed method in fact achieves to catch the variations in real scenes with its multi mixture model based coarse to fine approach. In addition, the use of the SAM metric could successfully compensate for the light changes, which can frequently occur in real scenes.

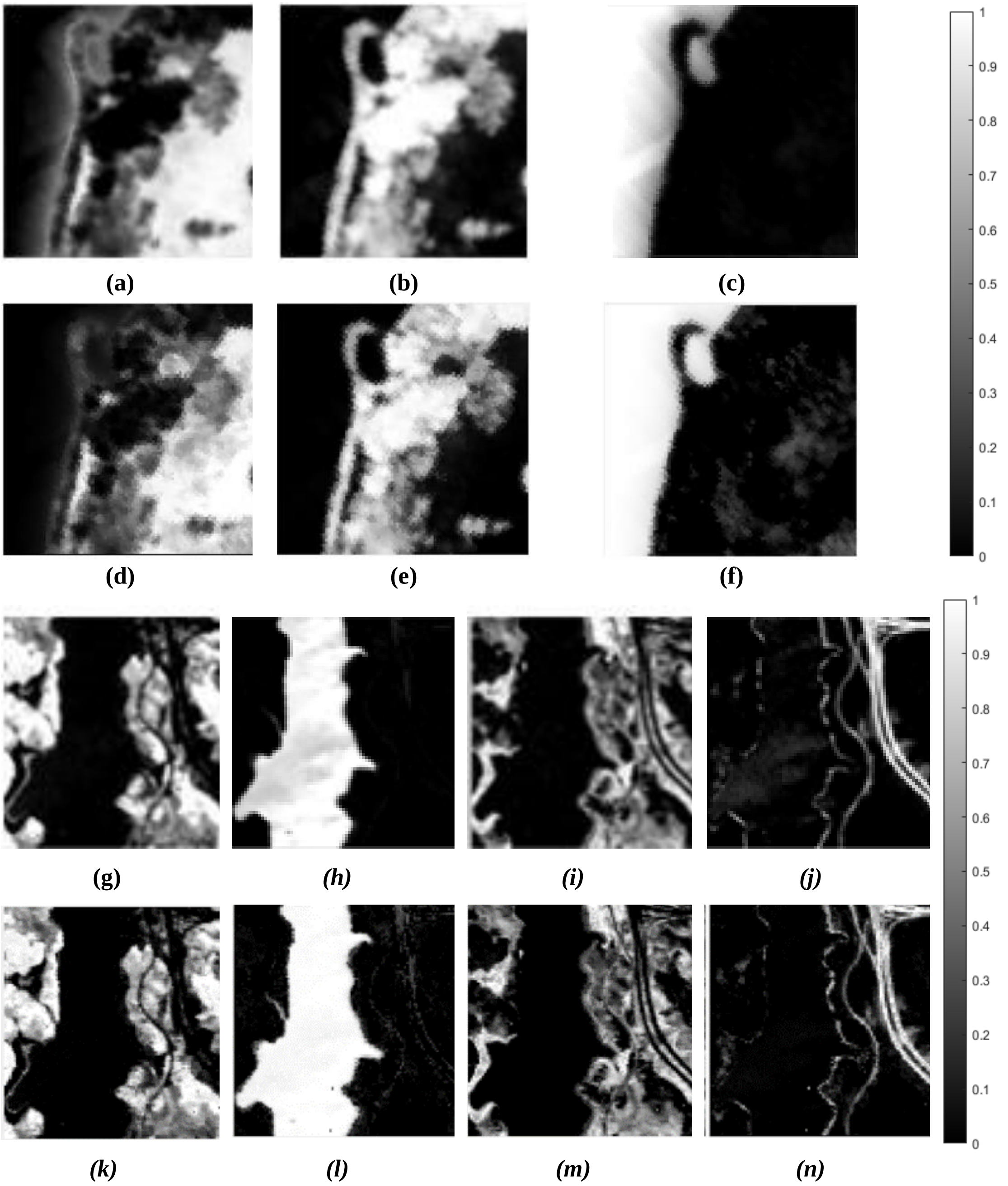


Figure 35 The results for Jasper Ridge and Samson Ridge. (a), (b) and (c) are the ground truths for the abundances for soil, tree and water, respectively and (d), (e) and (f) are the estimated abundances with the proposed algorithm for Samson Ridge data. (g),(h),(i) and (j) are the ground truths for the abundances for tree, water, soil and road, respectively, and (k), (l),(m) and (n) are the estimated abundances with the proposed algorithm for Jasper Ridge data

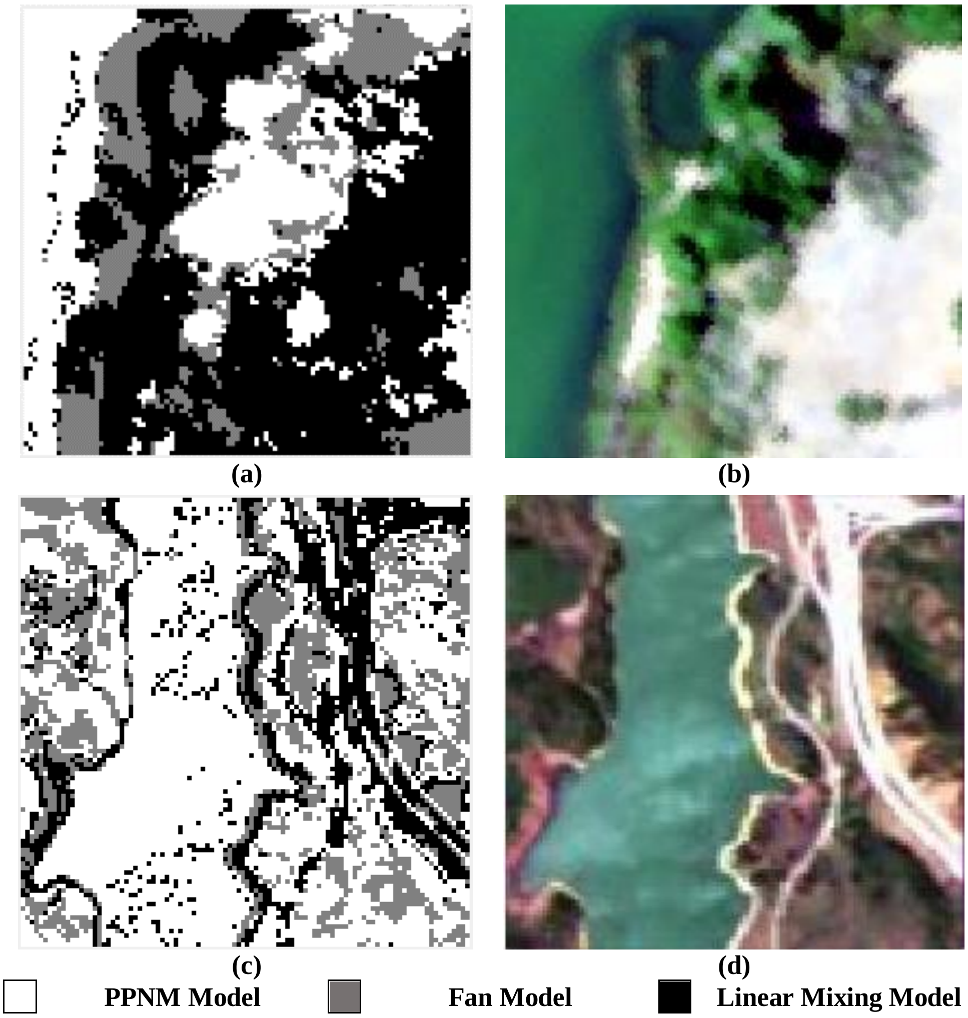


Figure 36 Displaying the models of the results obtained with the proposed algorithm on the basis of pixels LMM(Black), FM(Grey), PPNM(White) (a-Model estimation results for Samson Ridge, b-RGB image of Samson Ridge, c- Model estimation results of Jasper Ridge and d-RGB image of Jasper Ridge data

CHAPTER 6

PROPOSED NONLINEAR UNMIXING WITH 3D CONVOLUTIONAL ENCODER (3DCE)

This chapter presents a 3D convolutional encoder (3DCE) based hyperspectral unmixing algorithm. The use of the autoencoder structure has been observed in the deep learning-based unmixing algorithms that are widely used recently. Although this structure is successful in extracting the features in the data, it does not contain spatial information. Accordingly, 3D convolutional networks using neighborhood information together with spectral information is considered as an effective means to improve unmixing performance. In addition, observing that the related algorithms in the literature are generally tailored for LMM without taking more complex interactions in real data that can be modeled with nonlinear mixtures, a nonlinear part is integrated to 3D convolutional encoder structure to include such interactions in real data.

The chapter first examines the effect of optimization algorithms on the proposed 3D convolutional encoder based hyperspectral unmixing model. The parameters of the optimization algorithms are decided with respect to the performance changes of the examined model. The optimization algorithms are chosen as Adam, SGD, and Adagrad due to their high acceptance in the literature [116-124]. The performances of different distance units with these optimization algorithms are also investigated. Moreover, the experiments are conducted with different learning rates and batch sizes with different optimization methods and distance units. The comparisons of the presented model with the baseline algorithms in the literature are also given in this chapter.

6.1. Proposed 3D Convolutional Encoder Based Hyperspectral Unmixing

Conventional hyperspectral unmixing methods generally focus on a single problem, such as endmember estimation or abundance estimation. The performance of the algorithms for abundance estimation generally varies according to the endmember estimation performance. Deep learning-based hyperspectral unmixing methods have been proposed as an alternative to endmember estimation and abundance estimation methods [107-116],[121-124]. These methods can solve these two problems simultaneously. However, it has been observed that existing methods especially neglect spatial neighboring. In this context, a 3D convolutional encoder based method has been proposed for the use of spectral information together with spatial information. In addition, a nonlinear part has been added for the unmixing of nonlinear mixtures with better performance, which has an important place in the hyperspectral unmixing with real data.

The proposed model consists of three main parts. The first part illustrated in Figure 37 is the convolution part. This layer consists of three different 3D convolution layers and a flatten layer. 3D convolutional filters are used in 3 dimensions to contain both spectral and spatial information in the process. In the figure, P is the input channel of the hyperspectral data, which corresponds to the number of spectral bands in the hyperspectral image. L1, L2, and L3 are the spectral dimensions of 3D convolution filters. The output of this part is a flatten layer formed of 1D vectors, which has been converted from 3D inputs. These 1D vectors enter the next part as input.

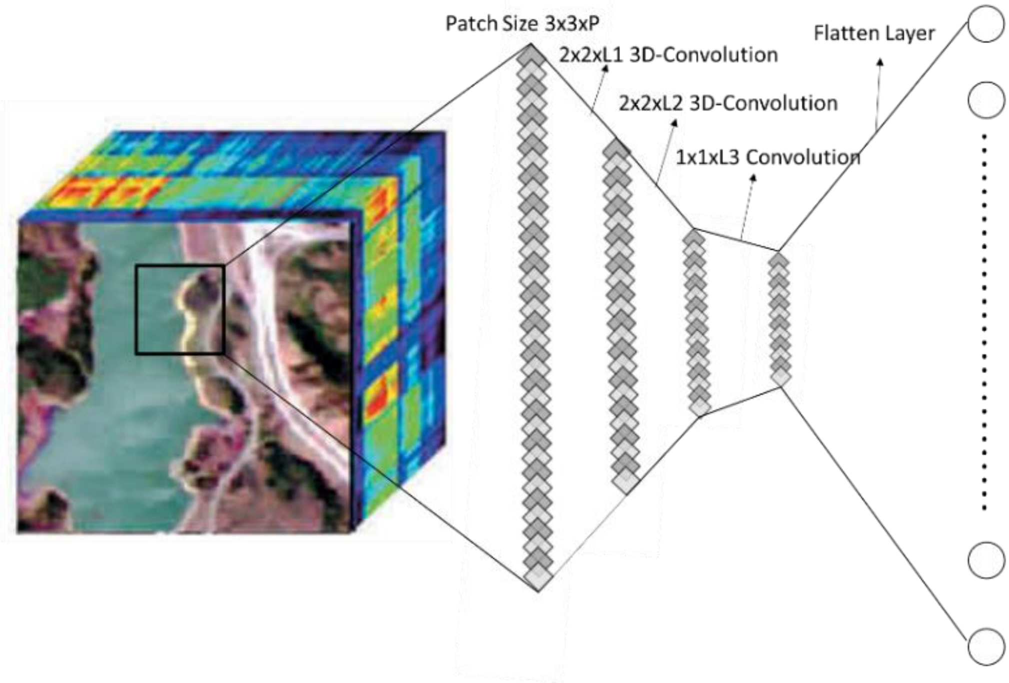


Figure 37 The convolution part of the proposed 3DCE model

The *x, y, z* parameters in the description of the filters correspond to the spatial coordinates in *x* and *y* directions and the spectral coordinate in the third *z* dimension. For a data with 220 spectral bands, an example for the application of a 3D convolution is given in Figure 38. The first filter set of size 2x2x21 is applied to the first patch of size 3x3x220 with different weights as the number of filters. The filter size is defined as 21 for this filter. As a result, a 2x2x200x21 size feature map is obtained. Then a second filter set of 2x2x11 size is applied on this feature map. The number of filters for this filter set is set to 11. As a result, after this filter, another 1x1x190x11 size feature map is obtained. The last filter is 1x1x7 in size and consists of 7 filters. As a result of this process, a 1x1x183x7 feature map is obtained. The flatten layer returns the last feature map into a one-dimensional vector. As a result, an output of 1288x1 for this example is obtained. This output is given as input to the autoencoder part. Convolution filters are used to extract low and high level features in the image in each layer.

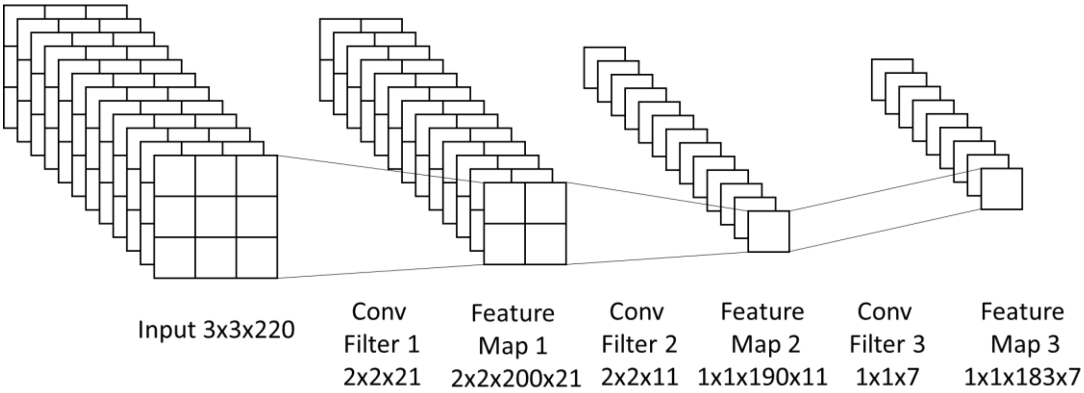


Figure 38 An example for the application of filtering for 3D convolutional part

The second part of the proposed model, namely the autoencoder part, illustrated in Figure 39, consists of two main stages, which are encoder and decoder. Autoencoders are generally used to reduce the data size they receive as input to a smaller size and learn the data structures by bringing them back to the input size. This structure is used in hyperspectral unmixing by reducing the received spectral signal to the number of endmember size and generating a signal again. With the restrictions provided in this layer, restrictions such as sum to one and positivity in hyperspectral unmixing can be enforced. This layer output is then transformed into the input signal again in the decoder layer, and both abundance estimation and endmember estimation operations are performed. In the proposed model, the 3D convolution part's output is given as an input to the encoder section. Then, the encoder-decoder structure, which provides the abundance estimation and endmember estimation processes, is established. For this process, the widely accepted encoder-decoder structures in the literature are used [124], [135].

In the autoencoder part, the most important factor affecting the performance stands out as the normalization layer applied before the output of the encoder. This layer enables the application of the two most important constraints in hyperspectral unmixing. These are the constraint of positivity, which is the positive encoder output provided by ReLU, and the constraint of sum to one. This layer is applied as,

$$\boldsymbol{W}_{output,i} = \frac{\boldsymbol{w}_{input,i}}{\sum_{j=1}^{T} \boldsymbol{w}_{input,j}} \tag{34}$$

where i is the index of the node, $\boldsymbol{w}$ is the weight matrix, and T is the length of the node.

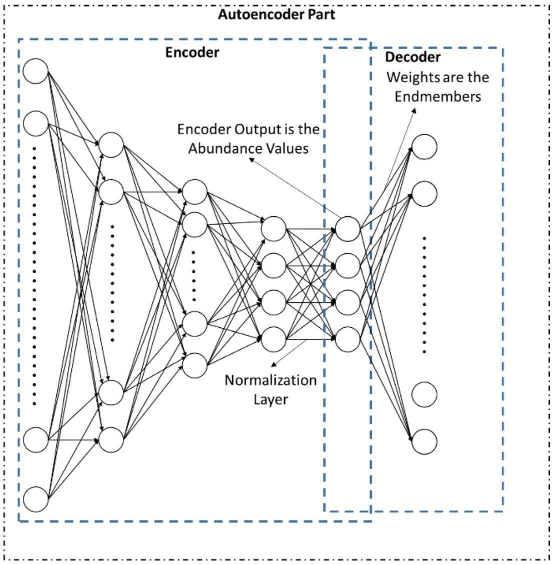

Autoencoder Part
Encoder
Decoder
Weights are the Endmembers
Encoder Output is the Abundance Values
Normalization Layer

Figure 39 The autoencoder part of the proposed 3DCE model

The given autoencoder structure is made for the hyperspectral unmixing process for linear mixing models, but it cannot unmix nonlinear mixtures. Figure 40 shows the nonlinear part of the presented 3DCE model, which enables hyperspectral unmixing for nonlinear mixing models. This model has a single external node as input. The model can adjust the nonlinearity coefficients by optimizing the weights connected to this node. In the output layer, the summation of the output of the autoencoder part and the output of the nonlinear part is provided.

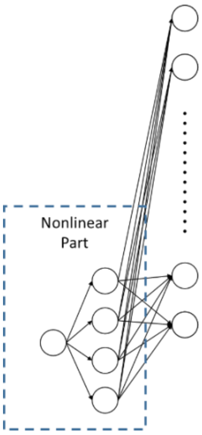


Figure 40 Nonlinear part of the Proposed 3DCE Model

The whole structure of the proposed 3DCE model with the autoencoder and nonlinear parts for nonlinear hyperspectral unmixing is presented in Figure 41. As mentioned, P is the input channel of the hyperspectral data. L1, L2, and L3 are the spectral dimensions of 3D convolution filters.

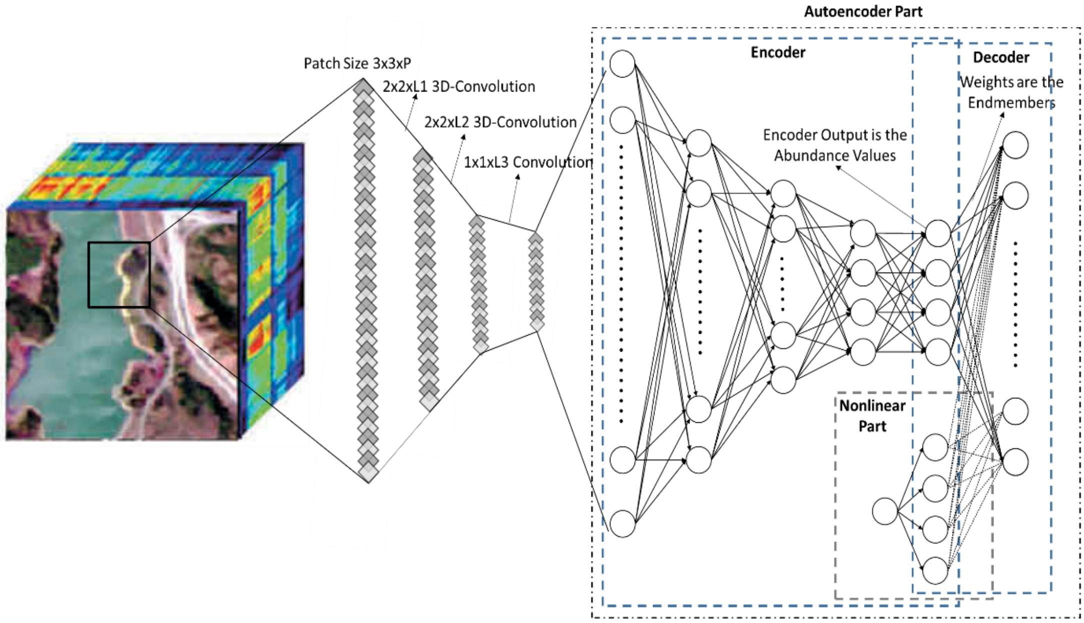


Figure 41 Proposed 3D convolutional autoencoder based deep learning model (3DCE) structure

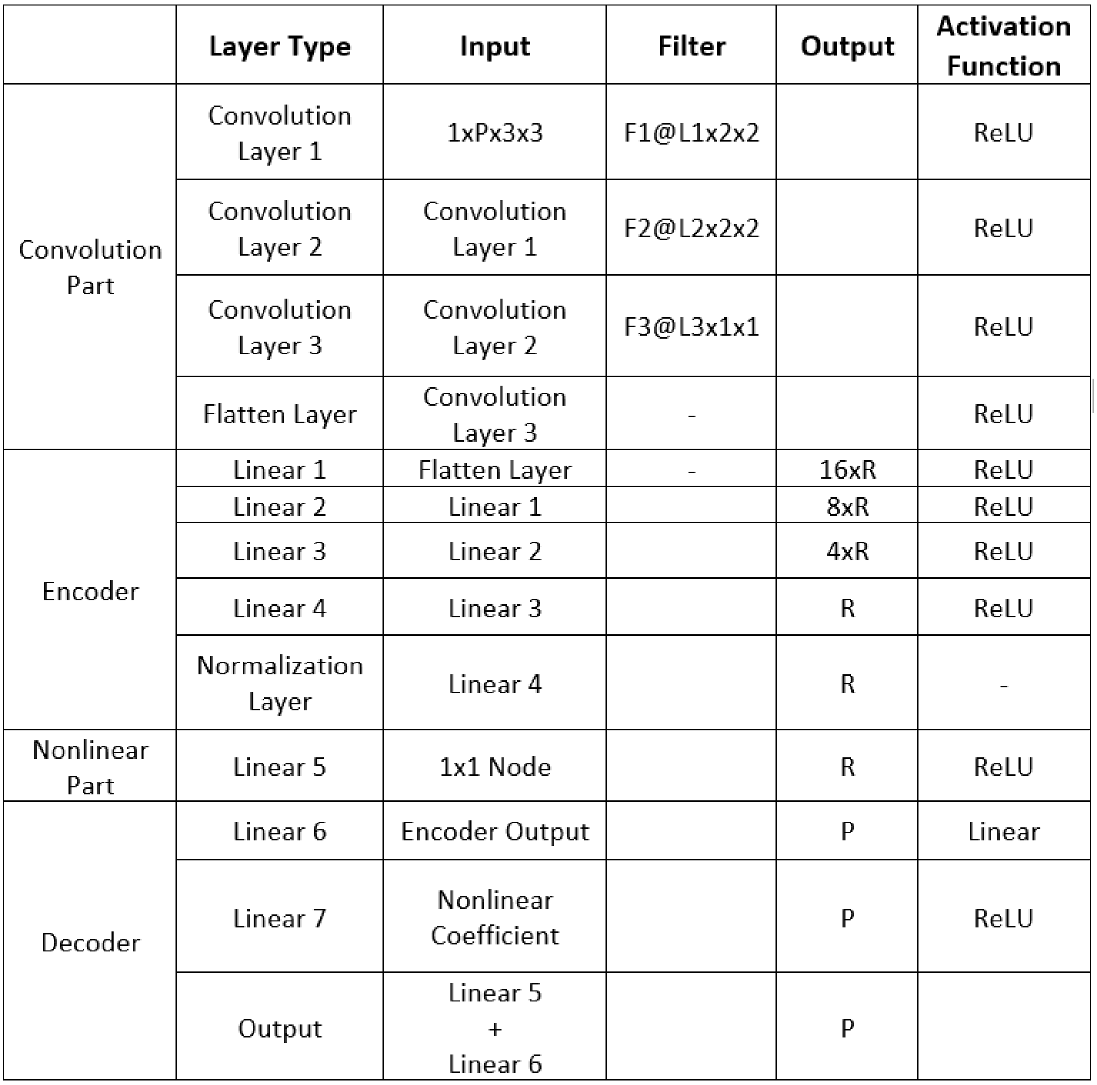
The detailed structure of the proposed model is shown in Table 4, for each part of the model, including layer type, input, filter type, output, and activation functions. The convolution part has three filters, and a flatten layer. This layer, which is the input of the network, receives a 3-dimensional matrix, including neighbor pixels as input. The encoder layer behind the convolution layer consists of 4 fully connected layers and a normalization layer. The connection with the section where nonlinearity is defined is made in the decoder layer, which is the model's output section. The presented network's activation function is chosen as ReLU, whose details are given in Chapter 3.1.6.2.

In the table, P is the input channel of the network, and R is the number of endmembers. While F1, F2, and F3 are the number of filters applied at each layer. Note that these filters are applied separately for each patch during the realization of the network. L1, L2, and L3 are given as the filter size applied to the spectral band. The parameters F1, F2, F3 and L1, L2, L3, are both determined as 21,11,7 in the experiments conducted in this study. The initial network values are determined with the kaiming initializer provided by He et al. [136], which is widely used in the literature [87], [99]. Besides, the endmembers in the encoder-decoder structure are initialized with the VCA output. The proposed method is implemented by using the python programming language and PyTorch [137]. The main steps of the implementation can be summarized as follows:

- Each pixel on the image is given as a block of 3x3xP with its 3x3 neighbors as input to the convolution layer.
- The convolution output is then given as input to the encoder decoder structure.

- The difference (error) between the estimated pixel spectra at the output of the autoencoder and original pixel spectra is calculated with respect to a distance metric.
- The total number of the pixel determines the number of updates in each iteration. When batch is used, the number of updates is equal to the total number / batch size. In this case the error is determined as the average distance for each batch.
- Learning process is completed after the last iteration.
- After the last iteration, the ultimate weights between the encoder-decoder structure and the encoder output are determined as endmembers and abundance values.

Table 4 Main structure and parameters of 3DCE



	Layer Type	**Input**	**Filter**	**Output**	**Activation Function**
Convolution Part	Convolution Layer 1	1xPx3x3	F1@L1x2x2		ReLU
	Convolution Layer 2	Convolution Layer 1	F2@L2x2x2		ReLU
	Convolution Layer 3	Convolution Layer 2	F3@L3x1x1		ReLU
	Flatten Layer	Convolution Layer 3	-		ReLU
Encoder	Linear 1	Flatten Layer	-	16xR	ReLU
	Linear 2	Linear 1		8xR	ReLU
	Linear 3	Linear 2		4xR	ReLU
	Linear 4	Linear 3		R	ReLU
	Normalization Layer	Linear 4		R	-
Nonlinear Part	Linear 5	1x1 Node		R	ReLU
Decoder	Linear 6	Encoder Output		P	Linear
	Linear 7	Nonlinear Coefficient		P	ReLU
	Output	Linear 5 + Linear 6		P	

6.2. Experimental Results and Discussion

The experiments for the proposed 3DCE model are divided into three main groups as the experiments to evaluate the utilized optimization method, observe the effect of different cost functions, and understand the effect of the learning rate and batch size. The experimental comparisons on the utilized optimization methods for the proposed 3DCE include Adam, SGD, and Adagrad optimizers. The cost functions for the comparisons are selected as MSE, L1-norm, SAM, and SID. In addition, the experiments with different learning rates and batch sizes are performed, and their effects on the endmember and abundance estimation are investigated. Finally, the proposed 3DCE method based hyperspectral unmixing method is compared with the baseline endmember estimation and abundance estimation algorithms in the literature. VCA and SISAL algorithms, which are found successful from the endmember estimation algorithms, are chosen for comparisons. Similarly, LMM, MLM, and PPNM algorithms among the abundance estimation algorithms are chosen for comparisons. Furthermore, the proposed 3DCE based hyperspectral unmixing algorithm is also compared with the proposed optimization-based coarse to fine abundance estimation algorithm in Chapter 5. In this section, the performance of abundance estimation is evaluated in terms of the RMSE between the estimated abundance values and the ground truth abundance values.

6.2.1. Experiments for the Utilized Optimization Method and Cost Function

The performances of optimization algorithms such as Adam, SGD, and Adagrad with different learning rates and different distance metrics such as SAM, SID, MSE, and L1-norm are examined. Detailed information on these optimization algorithms and distance metrics is given in Chapter 3.2 and Chapter 4.2. The experiments are conducted with different batch sizes for each algorithm and different learning rates. The range of the learning rate for the experiments is chosen from 0.1 to 0.0000001. The error in the experiments in this section is reported as RMSE between the original and estimated abundance values and the spectral angle between the original endmember and estimated endmembers. The number of iterations is determined as 50 for the experiments.

Table 5 3DCE with Adam optimizer abundance estimation performance for synthetic data with batch size 1 in terms of RMSE

Learning Rate	Adam-SAM	Adam-MSE	Adam-L1 Norm	Adam-SID
0.1	0.27	0.23	0.23	0.23
0.01	0.21	0.24	0.17	0.20
0.001	0.18	0.13	0.11	0.12
0.0001	0.13	0.09	0.10	0.11
0.00001	0.07	0.08	0.08	0.09
0.000001	0.07	0.08	0.09	0.07
0.0000001	0.13	0.15	0.15	0.13

Table 5 shows the results for Adam optimizer results with different learning rate and distance metrics as the first experiment. Adam optimizer is more successful at low learning rates. Moreover, the algorithm performs better with 0.00001 and 0.000001 learning rates. This is because the change in reconstruction error between iterations for each distance unit has a different effect on the cost function. It has been observed that higher performance is obtained at low learning rates for Adam optimizer. It is also seen that the performance drops considerably when a very high learning rate, such as 0.1, is selected. The performance of Adam optimizer with SAM is higher than with MSE and L1-Norm in the experiments.

Table 6 shows the results for SGD, which is tested as the second optimizer. Similar to the Adam algorithm, the learning rate for the SGD algorithm has a great impact on performance. While MSE and L1-Norm give a minimum value for 0.1 learning rate, SID give a minimum at 0.01 learning rate and SAM at 0.0001 learning rate. It is also observed that SGD-MSE and SGD-L1 reach their minimum values for 0.1 learning rate. Since this value is the final value in the selected range, SGD-MSE and SGD-L1 experiments are also repeated for 0.2 learning rate. The resulting values are obtained as 0.09 and 0.13. These different learning rates with minimum values also indicate that the performance of SGD is more affected by cost functions. Also, the SGD optimization algorithm works successfully with a higher learning rate than Adam optimizer. The main reason for such a behavior is that the algorithm iterates in random searches initially. The experiments have reveals that SGD has higher performance with SAM and SID.

Table 6 3DCE with SGD optimizer abundance estimation performance for synthetic data with batch size 1 in terms of RMSE

Learning Rate	SGD-SAM	SGD-MSE	SGD-L1 Norm	SGD-SID
0.1	0.12	0.08	0.07	0.16
0.01	0.15	0.09	0.08	0.07
0.001	0.08	0.11	0.08	0.08
0.0001	0.07	0.15	0.11	0.10
0.00001	0.15	0.17	0.16	0.16
0.000001	0.15	0.25	0.23	0.21
0.0000001	0.25	0.25	0.25	0.25

The results for the Adagrad algorithm can be seen in Table 7. Similar to the SGD algorithm, the Adagrad algorithm reaches minimum error rates at different learning rates with different distance metrics. It achieves a minimum error rate with a learning rate of 0.0001 in SID and SAM and 0.01 in L1-Norm and MSE distance metrics. It has been observed that Adagrad gives more successful results with MSE. Adagrad algorithm also achieves higher performance with a low learning rate like Adam algorithm. The minimum value in MSE has been observed to be lower than other distance measures with the Adagrad algorithm. It has also been observed that, unlike other algorithms, the Adagrad algorithm achieves higher performance with MSE.

Table 7 3DCE with Adagrad optimizer abundance estimation performance for synthetic data with batch size 1 in terms of RMSE

Learning Rate	Adagrad-SAM	Adagrad-MSE	Adagrad-L1 Norm	Adagrad-SID
0.1	0.15	0.10	0.16	0.15
0.01	0.10	0.06	0.11	0.16
0.001	0.10	0.10	0.11	0.09
0.0001	0.08	0.12	0.11	0.08
0.00001	0.15	0.16	0.17	0.15
0.000001	0.24	0.24	0.25	0.25
0.0000001	0.25	0.25	0.25	0.25

The minimum error rates obtained by the algorithms in the case of batch size 1 shows that SAM for Adam, SID for SGD, and MSE for Adagrad are the distance metrics with the lowest error. The main reason for this is that the effect of the difference between spectral signatures in the endmember learning process on the abundance value is less affected by distance metrics such as SAM and SID than distance measurement units such as MSE and L1 norm. When SAM is used as a distance metric, it is possible to estimate with the same abundance value in cases where spectral signatures have altitude difference, but this is not possible for other distance metrics.

6.2.2. Batch Size Experiments

The experiments are conducted to examine the effect of changing batch size on algorithm performance. The effect of batch size on algorithm performance has been included in several studies. There are studies indicating that the algorithms with larger batch size might achieve better performance with deep learning models [138–140]. Table 8 shows the best results of the Adam algorithm obtained for different batch sizes. Although the results are obtained with different learning rates, it is observed that the increase in batch size affects the performance positively when the distance unit is SAM. When the results obtained for the Adam algorithm are examined, it is observed that using the learning rate of 0.001 or 0.0001 achieves the best results. Although there are minor changes in other algorithms, the obtained values are very close to each other.

Table 8 3DCE with Adams algorithm abundance estimation performance change on the batch size in terms of RMSE

Batch Size	Adam-SAM	Adam-MSE	Adam-L1 Norm	Adam-SID
1	0.065	0.075	0.075	0.065
50	0.055	0.080	0.080	0.065
100	0.045	0.080	0.070	0.065
250	0.045	0.085	0.070	0.070

The results for the SGD algorithm can be seen in Table 9. Considering the batch size results for the SGD algorithm, the use of low batch positively affects the performance in the table. As a result of these experiments, the SGD algorithm achieves the best results when SID is used as the distance metric. SGD algorithm converges with more iterations than other algorithms due to its structure. Therefore, when high batch is used, network renewal decreases significantly between each iteration. Performance decreases in high batches due to the choice of constant iteration and high learning rate in these experiments. However, when the number of epochs is increased, more successful results are obtained in high batches. In the experiments repeated for 100 batches, it is observed that when the iteration number is set to 250, the performance increases to 0.50 on average for SGD-SAM.

Table 9 3DCE with SGD algorithm abundance estimation performance change on the batch size in terms of RMSE

Batch Size	SGD-SAM	SGD-MSE	SGD-L1 Norm	SGD-SID
1	0.070	0.075	0.070	0.065
50	0.075	0.080	0.065	0.070
100	0.090	0.090	0.075	0.075
250	0.115	0.130	0.150	0.105

Table 10 shows the results of the Adagrad algorithm. Unlike other algorithms, batch size change in the Adagrad algorithm does not cause an evident change in the results, but it has been observed that the best results are obtained using one batch and MSE. Adagrad indicates higher performance with MSE, which shows more variation between iterations.

Table 10 3DCE with Adagrad algorithm abundance estimation performance change on the batch size in terms of RMSE

Batch Size	Adagrad-SAM	Adagrad-MSE	Adagrad-L1 Norm	Adagrad-SID
1	0.075	0.055	0.105	0.075
50	0.080	0.070	0.075	0.110
100	0.065	0.060	0.080	0.055
250	0.095	0.055	0.105	0.100

6.2.3. Endmember Extraction Performance

In the proposed network structure, the weights between the encoder and decoder layer correspond to the endmembers, as explained in Section 6.1. Experiments have been carried out for endmember estimation, which is an essential step in hyperspectral unmixing. First of all, endmember extraction experiments for optimization algorithms with synthetic data are performed. The endmembers extracted from the results are compared with the original endmembers used in creating synthetic data. Error rates for endmember estimation are given as the SAM result in radians. The endmembers extracted by the Adam algorithm are shown in Figure 42. The average SAM error between estimated endmembers and the ground truth for this result is 0.04 radians.

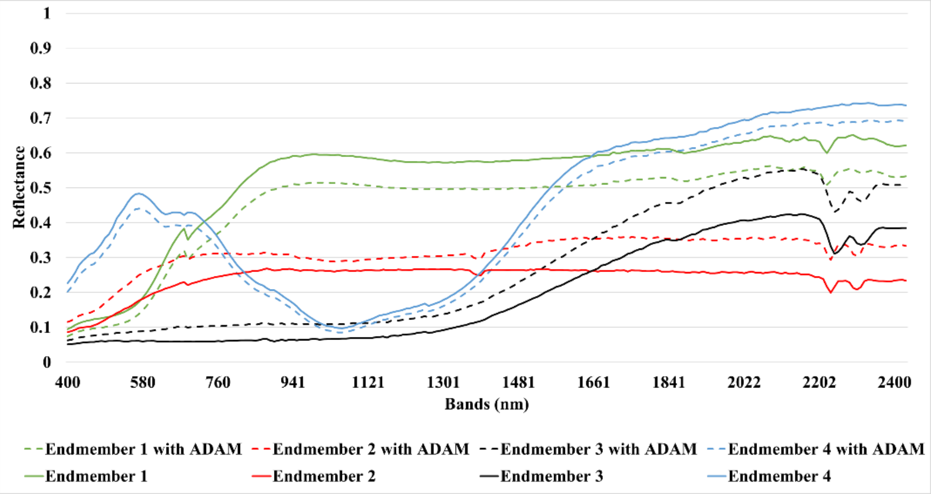


Figure 42 The endmembers extracted with 3DCE using Adams algorithm

It has been observed that the difference between extracted endmembers and ground truth is low for all algorithms. The endmembers extracted by the SGD algorithm are shown in Figure 43. The average SAM error between estimated endmembers and ground truth for the SGD algorithm is 0.04 radians.

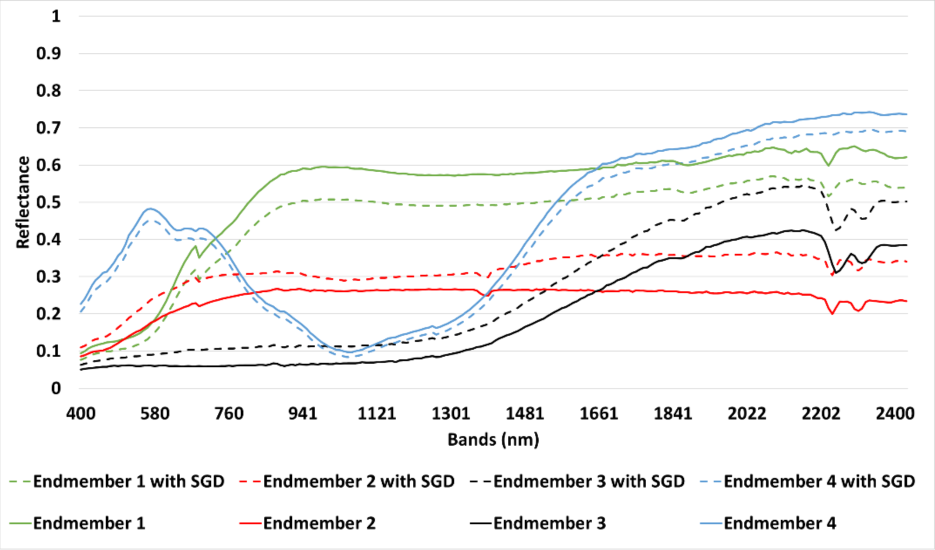


Figure 43 The endmembers extracted with 3DCE using SGD algorithm

The endmembers extracted by the Adagrad algorithm are shown in Figure 44. The average SAM error between estimated endmembers and ground truth for the Adagrad algorithm is 0.03 radians.

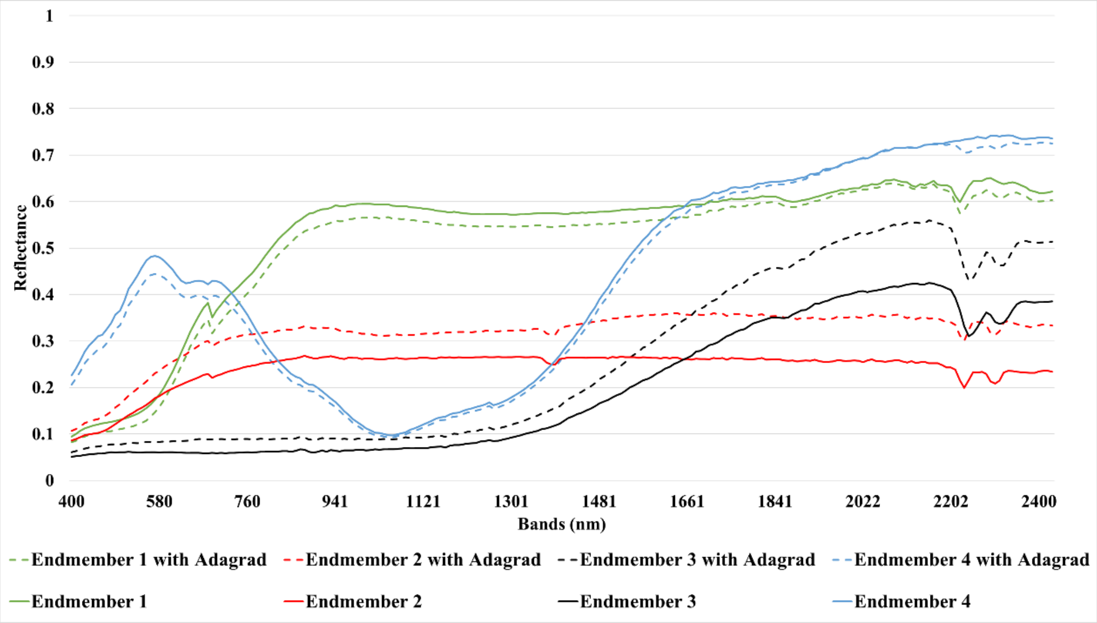


Figure 44 The endmembers extracted with 3DCE using Adagrad algorithm

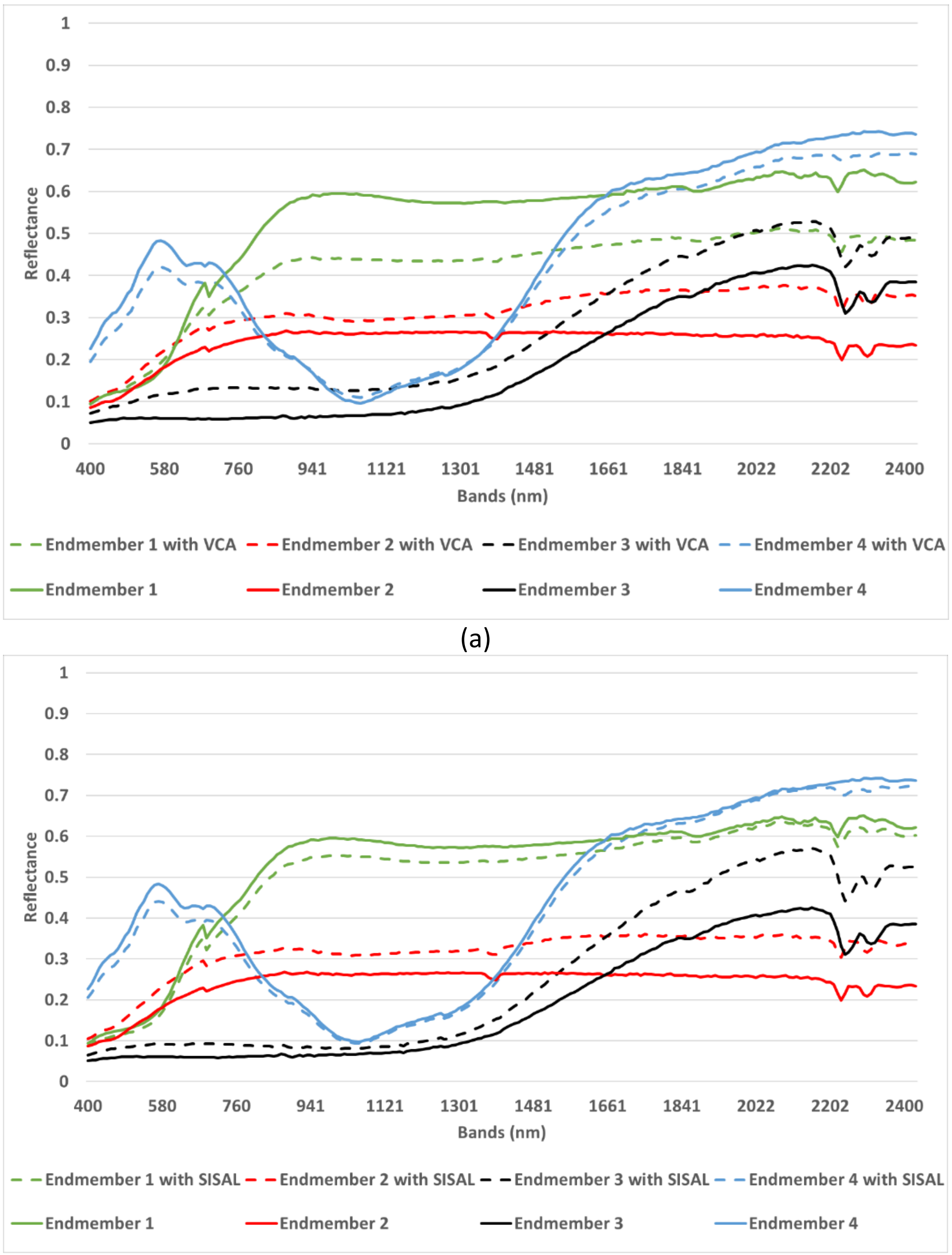
When the results are examined, it is seen that the error is lower when SAM and SID are used as distance metrics. The main difference between these metrics compared to the others is that they are less affected by noise when calculating errors. The extracted endmembers with these metrics have also been observed more successfully. Therefore, it is decided to use the Adam algorithm with SAM in the proposed 3DCE method.

6.3. Comparison of abundances with baseline methods in the literature

In the experiments performed in this section, different comparisons are made for endmember extraction and abundance estimation. While the distance unit used for endmember comparisons is SAM, the distance unit used for abundance estimation is given as RMSE. The experiments are performed 10 times, and the averages of these experiments are given in the final tables. The images show the best results obtained in these experiments.

The previous experiments show that 3DCE with Adam as an optimizer and SAM as a distance metric achieves better results. Therefore, synthetic and real data comparisons are made with these methods. VCA and SISAL algorithms are used for endmember estimation as the well-known standard methods for hyperspectral unmixing. LMM, MLM, and PPNM algorithms, which are successful in previous experiments, are used for abundance estimation. Endmember estimation and abundance estimation performance are evaluated separately for both real and synthetic data.

For synthetic data, the estimated endmembers by VCA and SISAL algorithms are given in Figure 45. The average error between estimated endmembers and ground truth for VCA is 0.070, and for SISAL 0.035 in radians. The abundance estimation processes are performed with these endmembers. Since there is no pure pixel in the synthetic data, the SISAL algorithm is expected to perform better. Figure 42 illustrates the endmember obtained with the proposed 3DCE algorithm. The average estimation error between estimated endmembers and ground truth for the proposed 3DCE algorithm is 0.038 radians. As can be seen, the proposed method and SISAL yield similar results for synthetic data. The formed endmembers are characteristically observed very close to each other.



(a)

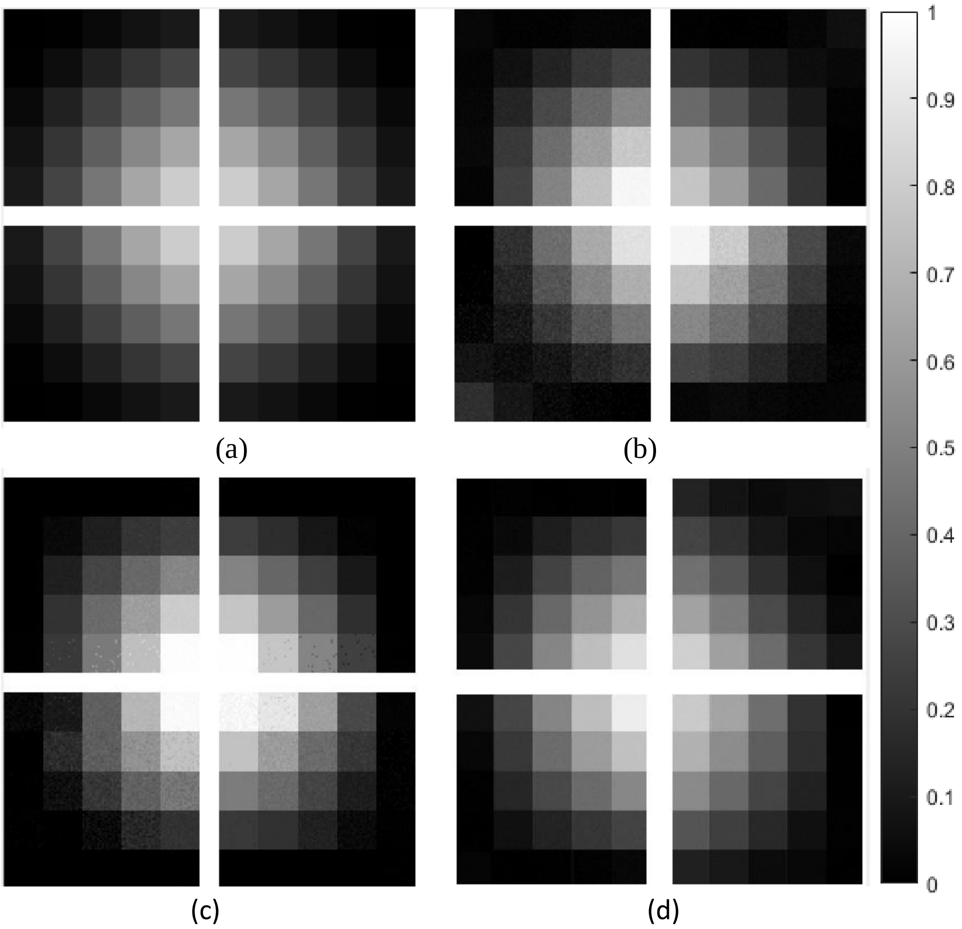
(b)

Figure 45 Extracted endmember with VCA (a) and SISAL(b) with synthetic data

The second aspect of the comparisons is determined as the comparison of abundance estimation performances. In Table 11, the average errors of the LMM, MLM, and PPNM algorithms are given for synthetic data with the extracted endmembers with VCA and SISAL. The abundance estimation performances are given as RMSE.

Table 11 Abundance Estimation Performance with the Extracted Endmembers in terms of RMSE

	LMM	MLM	PPNM	Proposed Abundance Estimation Algorithm	3DCE
Endmembers with VCA	0.1595	0.1438	0.1336	0.1413	**0.0498**
Endmembers with SISAL	0.0600	0.0600	0.0550	0.0533	



(a) (b)

(c) (d)

Figure 46 (a) Ground Truth and the results for (b) VCA+PPNM, (c) SISAL+PPNM and (d) Proposed 3DCE Method

Similar to the experiments in Chapter 5, the PPNM algorithm's performance with synthetic data is observed more successfully than other algorithms in these experiments. When used with the SISAL algorithm, the proposed algorithm has been observed that the proposed algorithm performs with an error rate of 0.055. However, when analyzed for this data, LMM and MLM algorithm performances are also at a rate close to 0.06. Considering the analyzes in Chapter 5, these experiments' performance is highly dependent on the performance of the endmember extraction algorithm. In the experiments performed with the algorithm presented in Chapter 5, successful results are obtained in synthetic data when using endmember extracted with SISAL. However, as in other algorithms, the performance is relatively low when using VCA. The performance of the proposed 3DCE for abundance estimation is 0.049. Compared to other results, an improvement of 10% has been achieved. Considering that it does not contain any pure pixels on the synthetic data and is created with a non-linear mixing model, the abundance estimation result obtained is considered quite successful.

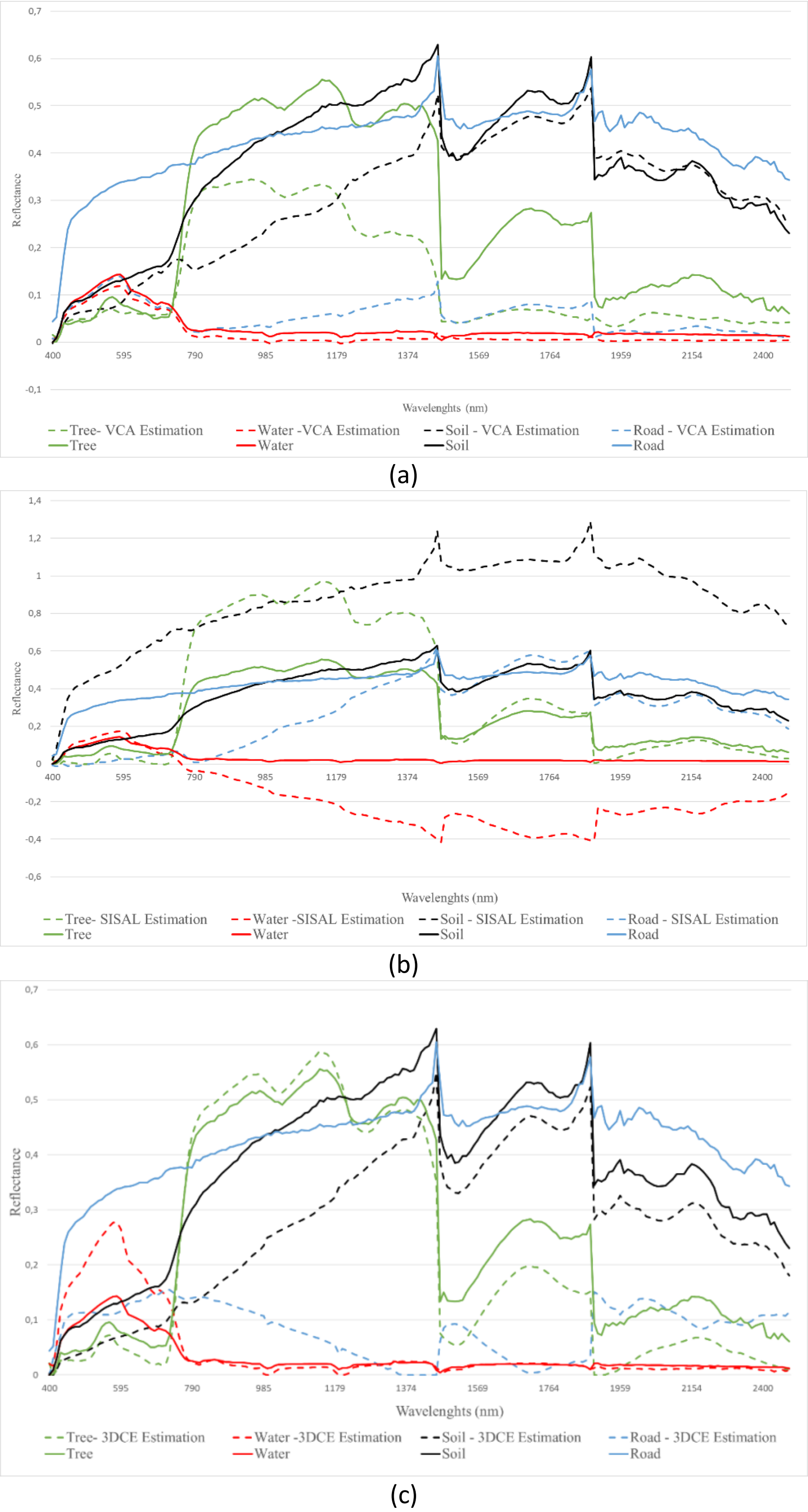
In Figure 46 (a), the ground truth for the abundance of endmembers in synthetic data, in Figure 46 (b), Figure 46 (c) and Figure 46 (d), the VCA + PPNM result, the SISAL + PPNM result, and the proposed 3DCE method result are given, respectively. The differences that occur due to the endmember estimation are noticeable, even if the differences are small. When the errors in the presented algorithm are examined in detail, it is observed that the errors are generally located in the border regions. Also, in cases where the endmember estimation error is high, the error rate is higher than the average in regions where the endmember's abundance value is high.

6.4. Real Data Experiments

Similar to the analysis in Chapter 5, two different real data sets are used for real data experiments. For Jasper Ridge and Samson Ridge datasets, the results of VCA and SISAL algorithms combined with LMM, MLM, and PPNM algorithms are obtained. In addition to these conventional hyperspectral endmember estimation and abundance estimation models, the comparisons are also performed with the autoencoder based deep learning method proposed by Palsson *et al.* [124]. The method uses encoder-decoder structure for hyperspectral unmixing. As the final investigation, the proposed method with the nonlinear layer is compared with the autoencoder structure without nonlinear layer to reveal the effect of the integrated layer to the unmixing performance.

6.4.1. Comparisons with Literature

Real data comparisons are performed with Jasper Ridge and Samson Ridge data tests. Extracted endmembers with VCA and SISAL algorithms for Jasper Ridge data are given in Figure 47 (a) and Figure 47 (b). Although VCA and SISAL algorithms work with high performance on synthetic data, their performance may change with real data. In the experiments, the performances of VCA and SISAL algorithms in real data have been observed. Although there are pure pixels in the data, the VCA algorithm has achieved endmember extraction performance with an average error rate of 0.32 radians. The SISAL algorithm, which achieved a higher performance rate on synthetic data, indicates deficient performance in the Jasper data set. The error rate for this data is determined as 0.59 radians. As can be seen from Figure 48 (b), since the algorithm does not enforce the exclusion of a negative value, it detects one endmember with a negative value for Jasper Ridge data. Algorithms have been run by taking the negative values of this endmember as zero due to the positivity assumptions of the abundance estimation algorithms. The endmembers obtained with the proposed 3DCE method are given in Figure 47 (c). The average error rate of these endmembers compared with the endmembers in the ground truth are determined as 0.29 radians, a value which is better than those with VCA and SISAL but still considered as a high value. While the tree, water, and soil signatures are found to be quite successful, the road signature is obtained with lower performance.

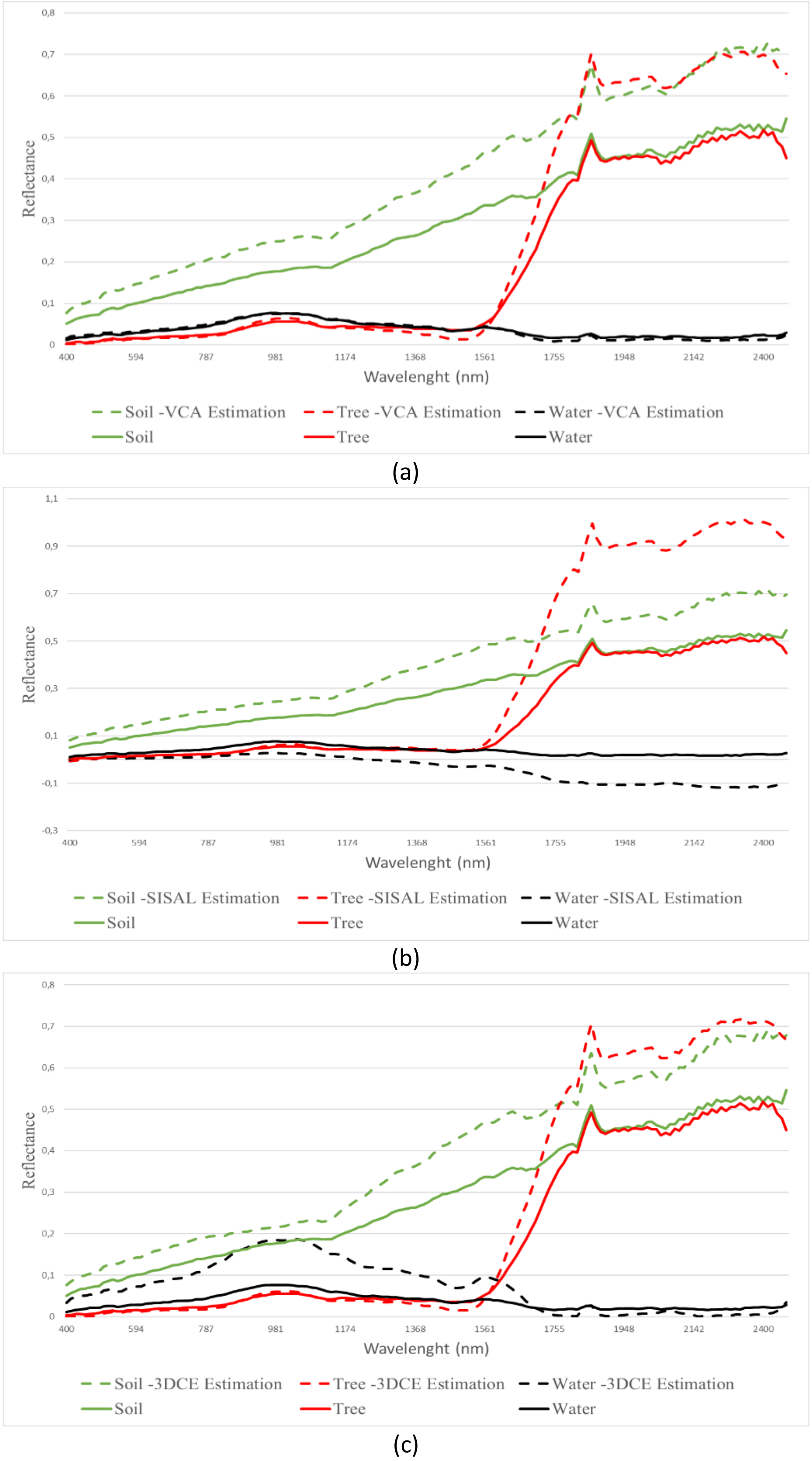


(a)

(b)

(c)

Figure 47 Extracted Endmembers with (a) VCA, (b) SISAL and (c) 3DCE for Jasper Ridge dataset



(a)

(b)

(c)

Figure 48 Extracted Endmembers with (a) VCA, (b) SISAL, and (c) proposed 3DCE for Samson Ridge dataset

Extracted endmembers with VCA and SISAL algorithms for the Samson Ridge dataset are given in Figure 48 (a) and Figure 48 (b). In contrast to the results obtained in the Jasper Ridge data set, the endmember extractions are found more successful in this data set for the VCA algorithm. The SISAL algorithm has extracted one endmember with a negative value, as in the Jasper Ridge data set. In abundance estimation experiments, negative values in these endmembers are also set to zero. The average error for endmember extraction is calculated as 0.06 radians for the VCA algorithm and 0.26 radians for the SISAL algorithm. The extracted endmembers with the proposed 3DCE method are given in Figure 48 (c). The error for these endmembers is calculated as 0.11 radians. Considering the endmember estimation experiments, the proposed algorithm performs endmember estimation successfully in both synthetic and real data.

Spectral diversity explains why the performance of both the proposed algorithm and the algorithms, such as VCA that chooses endmember from the data differs from the ground truth. Ozkan and Akar revealed that the pure pixels of the endmember in these datasets consist of different spectral signals [129]. Therefore, it is considered to be acceptable if the signals found are different from the ground truth. However, it has been observed that the experiments performed with the method proposed in Chapter 5, where the endmember information is available, yields better abundance estimation results.

After the endmember estimations, the abundance estimation performances by using these endmembers are examined. Table 12 shows the abundance estimation results with the endmembers extracted by VCA and SISAL algorithms for the Jasper Ridge dataset. The best result in this section is seen with VCA + MLM as 0.17. Moreover, when the abundance estimation algorithm presented in Chapter 5 is used together with VCA, it provides a performance increase of around 10%. The performance of the autoencoder based deep learning model is 0.18 similar to the other algorithms in the literature. Palsson's model, on the other hand, showed a slightly lower performance in this data than conventional methods, achieving 0.18. The proposed 3DCE algorithm provides a performance increase of around 20% compared to the best result for the Jasper Ridge dataset. Although it does not have a low error rate as in synthetic data, it has been seen to have an average error rate of 0.14.

Table 12 The performance of the algorithms for Jasper Ridge dataset as RMSE

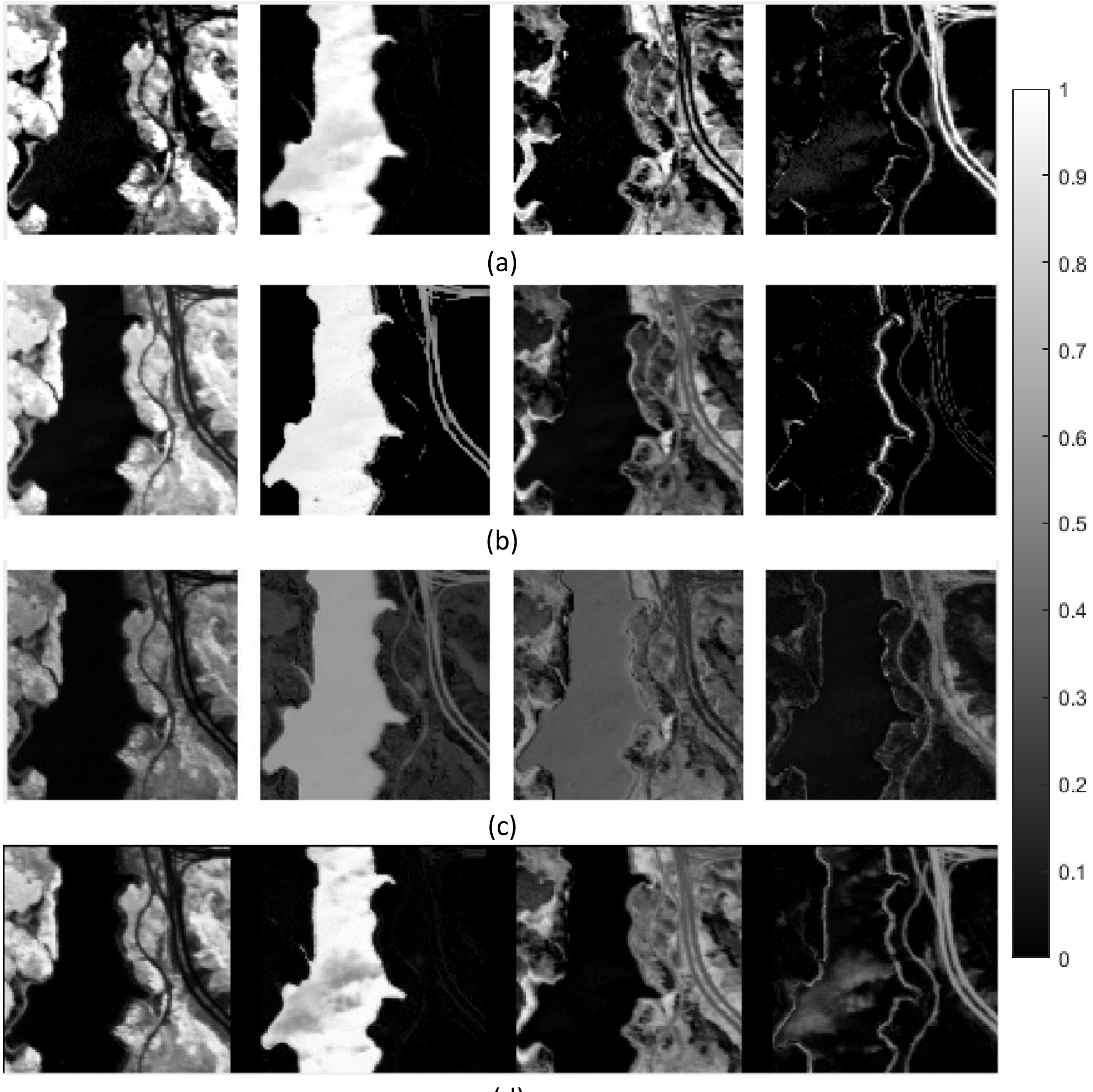
	LMM	MLM	PPNM	Proposed Abundance Estimation Algorithm	Palsson's Autoencoder Model	**Proposed 3DCE**
VCA	0.20	0.17	0.20	0.15	0.18	**0.14**
SISAL	0.23	0.21	0.23	0.22		

Table 13 shows the abundance estimation results with the endmembers extracted by VCA and SISAL algorithms for the Samson Ridge dataset. As with the Jasper Ridge dataset results, the combination of VCA + MLM algorithms has achieved a successful result for this dataset. Similarly, the abundance estimation algorithm presented in Chapter 5 provides a 10% performance increase when used with VCA. It has been observed that Palsson’s autoencoder based deep learning method performs better than Jasper Ridge data in this model. For Samson Ridge data, the performance of Palsson’s method is observed as 0.15. The model performance presented for the Samson Ridge dataset, where the endmember extraction performance is much better than the Jasper Ridge dataset performance, is obtained as 0.12. For this dataset, as in the other datasets, an improvement of about 20% is observed in the proposed 3DCE method compared to other methods in the literature.

Table 13 The performance of the algorithms for Samson Ridge dataset as RMSE

	LMM	MLM	PPNM	Proposed Abundance Estimation Algorithm	Palsson's Autoencoder Model	**Proposed 3DCE**
VCA	0.24	0.20	0.32	0.17	0.15	**0.12**
SISAL	0.25	0.25	0.28	0.25		

Figure 49 shows the ground truth and the abundance estimation results for the Jasper Ridge dataset. Figure 49 (a) shows the ground truth for the Jasper Ridge dataset. In Figure 49 (b), the results for VCA-MLM, in Figure 49 (c), the results for SISAL-MLM, and in Figure 49 (d), the results obtained with the proposed 3DCE method are given. Although the abundance estimation errors are not close to zero, when the results are examined visually, the main classes are predicted quite successfully with the proposed 3DCE method. When the results are examined for VCA + MLM combination, which is one of the successful methods, it is seen that the road and water classes are mixed, and a large part of the land is detected as soil.



(a)

(b)

(c)

(d)

Figure 49 The abundance estimation results for Jasper Ridge dataset (a) ground truth, (b) the results for VCA + MLM (c) the results for SISAL+MLM and (d) the results with proposed 3DCE

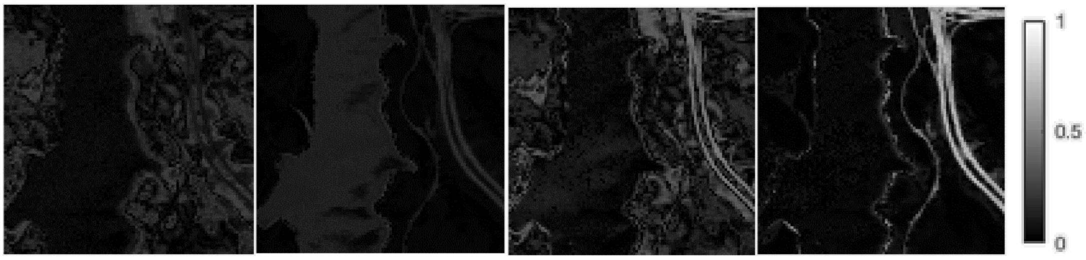


Figure 50 Error map showing the difference between the abundance map derived by the proposed 3DCE method and the ground truth for Jasper Ridge dataset

Figure 50 illustrates the error map between ground truth and the results of 3DCE model for Jasper Ridge dataset. Although the results are similar, it is observed that the rate of mixing road and soil materials with each other is higher than the other materials. The main reason for this is that there are very few road pixels and the neighborhood of the road and soil pixels is too high. Therefore, the presence of both road and soil in the area calculated for the road in the 3x3 filter used for the neighborhood is one of the reasons for this mixture.

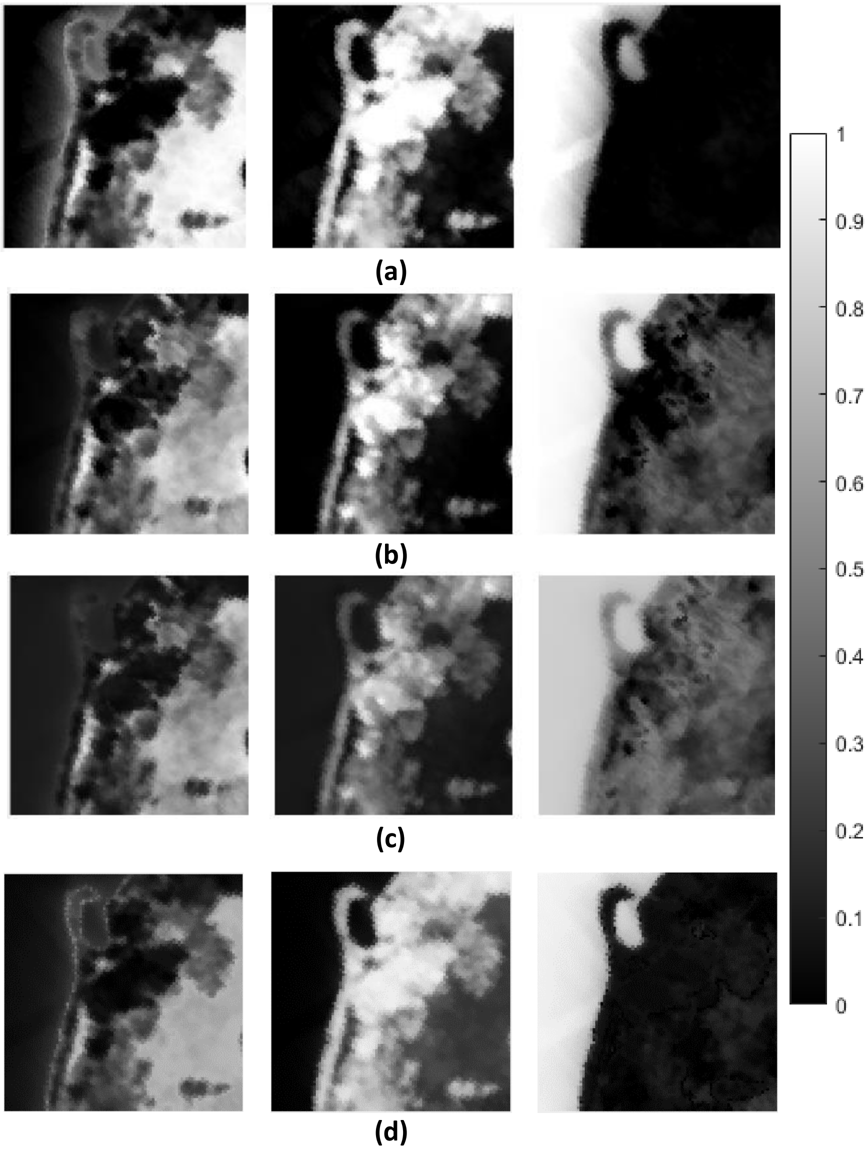


Figure 51 The abundance estimation results for Samson Ridge dataset (a) ground truth, (b) the results for VCA + MLM (c) the results for SISAL+MLM, and (d) the results with proposed 3DCE method

Figure 51 shows the ground truth and the abundance estimation results for the Samson Ridge dataset. In Figure 51 (a) ground truth of Samson Ridge dataset, in Figure 51 (b) VCA-MLM results, in Figure 51 (c) SISAL-MLM results, and in Figure 51 (d) the results obtained with the proposed 3DCE method are given. When these results are examined visually, it is seen that although the tree detection is quite successful with almost all methods, soil and water may not be detected as well. The abundance estimation for the proposed 3DCE algorithm is more apparent than other algorithms.

Figure 52 illustrates the error map between ground truth and the results of 3DCE model for Samson Ridge dataset. In this data set, it is observed that the errors are mostly between the tree and water pixels. The similarity of tree and water spectral signatures between 400-1500nm is another reason for this error. Also the abundance values of the other classes and the soil class have little confusion with the other classes.

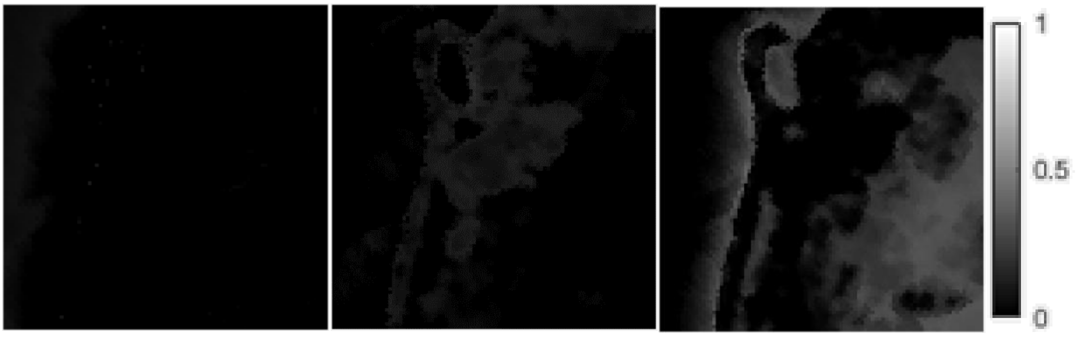


Figure 52 Error map showing the difference between the abundance map derived by the proposed 3DCE method and the ground truth for Samson Ridge dataset

In the experiments performed for the presented 3D encoder based hyperspectral unmixing algorithm, it is observed that the performance increase is achieved compared to the mainstream algorithms used in the literature. Experiments have been successfully performed on synthetic and real data for both endmember estimation and abundance estimation. It has been observed that the presented method works quite successfully for both data.

6.4.2. Comparison with Respect to Mixing Models and Nonlinear Layer

The pixel models in the Samson Ridge and Jasper Ridge datasets are shown in Figure 54. Experiments have been conducted to measure the performance of the presented 3DCE model on different mixing models. It is given as the average error over the previous experiments. Selected models are determined as LMM, FM and PPNM. The errors given in this section are given as RMSE of the pixels marked with that model.

In the experiments made for Jasper Ridge data, it is seen that the error value of the pixels marked as LMM is 0.17. This value is taken as 0.15 for FM and 0.10 for PPNM. For Samson Ridge data, the error values are determined as 0.12, 0.10 and 0.11 for LMM, FM and PPNM, respectively. It has been observed that the pixels with nonlinear mixtures are estimated more successfully than the pixels with the linear mixture model. Since the number of pixels modeled as the total nonlinear mixture model in these data is higher than the linear model, the overall performance has been achieved more successfully than using only the linear model. It has been observed that the nonlinear part positively affects the algorithm performance, since the complex interactions in the real data are too complex to be modeled with a single model. However, it is thought that the algorithm presented can work with lower performance on data created with only linear mixtures.

Therefore, separate experiments are carried out to measure the effect of nonlinear part on performance. In this section, the nonlinear part is removed from the model and the encoder-decoder structure after the 3D convolution layer is included. When the results are observed with this model, it is seen that the performance decreased from 0.14 to 0.17 for Jasper ridge and from 0.12 to 0.15 for Samson Ridge datasets. Although better estimation of linear models is achieved, the estimation of other models in the real data is more successful when nonlinear part is included to model. Considering that the number of pixels in different mixing models in the data are close to each other, the more successful separation of nonlinear mixing models, the better performance in predicting the vast majority of data.

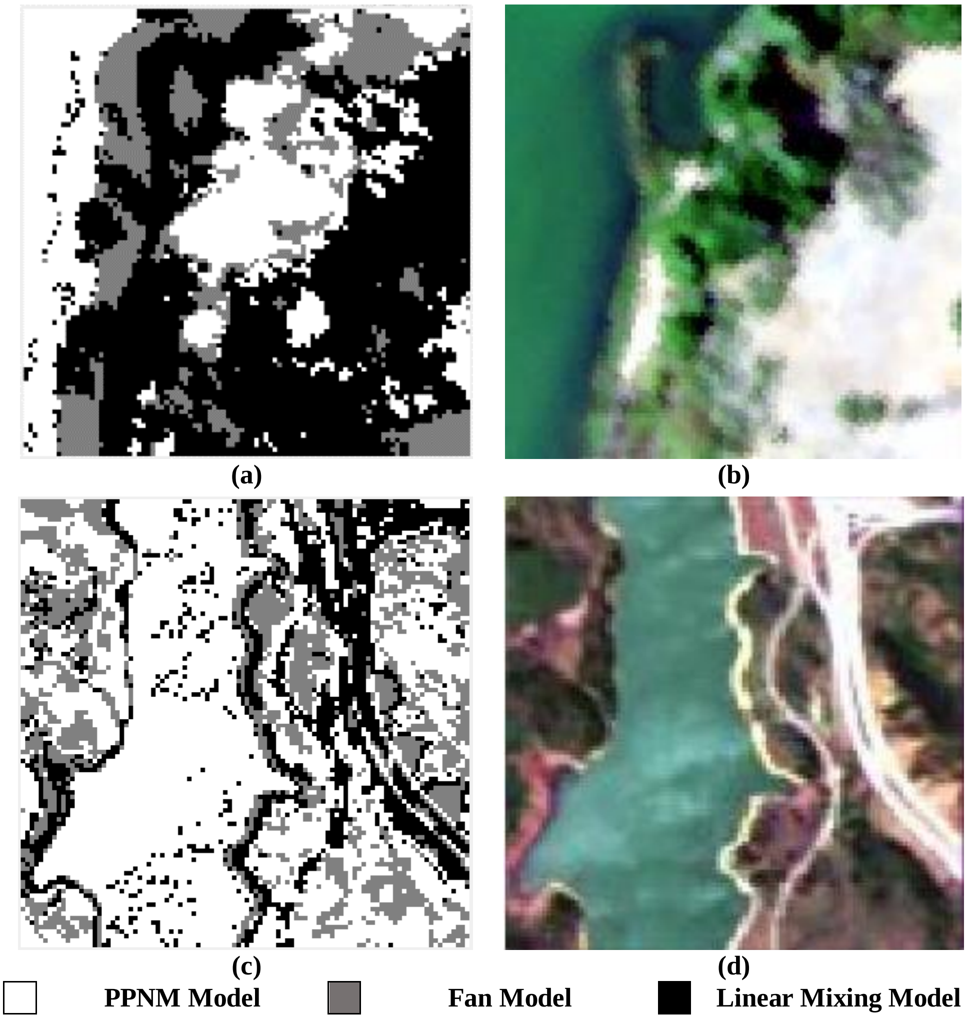


Figure 53 Displaying the models of the Samson Ridge and Jasper Ridge Data, LMM(Black), FM(Grey), PPNM(White) (a-Results for Samson Ridge, b- RGB image of Samson Ridge, c- Results of Jasper Ridge and d-RGB image of Jasper Ridge

6.4.3. Cross-validation of the Proposed Unmixing Model

The presented deep learning based hyperspectral unmixing method requires a separate learning process for each data. The main reason for this is that the spectral differences in each image and the materials that can be included in each data may differ. In other words, unlike traditional deep learning applications, the presented algorithm does not use the weights of a model by training a model for the hyperspectral unmixing process, but uses the encoder outputs and decoder weights in the final iteration.

The algorithm presented in this section is also tested with 10-fold cross-validation in order to observe performance measurements similar to traditional deep learning algorithms of the presented method. The cross-validation process is performed by dividing the picture into 10 parts and performing the training process of 9 parts at each time and using the unused part in the test process. In other words, considering the size of the Jasper Ridge data, 9000 pixels in 10000 pixels in Jasper Ridge data are used for learning process and 1000 pixels are used for testing. In Samson Ridge data, 8325 pixels in 9250 total pixels are used for the learning process and 925 pixels are used for the test process. During each training, a different section is separated as the test data and the average RMSE values of the operation on the whole data are given. Experiments are performed on Jasper Ridge and Samson Ridge datasets.

In the experiments performed for the Jasper Ridge dataset, when all data are used, the average error is determined as 0.14. In cross-validation experiments, the error value is obtained as 0.15. It is also observed that the error value of 0.12 in Samson Ridge dataset is found as 0.15 in cross-validation experiments. One of the main reasons for this decrease in the abundance estimation performance is that the proposed 3D convolutional encoder based unmixing method is tested on a data that the model has never seen before. Another reason for this decrease is that the learning process takes place with only 90% of the data. It is expected that performance changes will be observed as the percentage of the data used for training decreases.

As mentioned before, the presented model performs the learning process for the received hyperspectral data only by training. The main reason for this is that the weights coincide with the endmember during both endmember estimation and abundance estimation processes. It has been observed that the presented model works quite successfully in the presented cross-validation experiments to perform the abundance determination process on a different data by using the trained weights of the autoencoder. This application is valid only if it is applied for abundance detection for images containing the same materials.

CHAPTER 7

CONCLUSIONS

In this thesis, two different studies have been performed for the hyperspectral unmixing problem. The first study is the abundance estimation in the presence of endmember information, and the second study is the deep learning-based 3D convolutional encoder based hyperspectral unmixing method for the cases where the endmember information is unavailable. Comparative performance evaluations have been performed in both synthetic and real data for both models.

The first study is tailored for the abundance estimation for the cases where endmember information is available. In the performed study, a model on abundance estimation using the multiple mixture model in a single optimization process is presented. Performance variation of the presented model is investigated with SA, PS SQP, and GA optimization methods. In the experiments, the number of iterations for SA, PS, and SQP and the population and generation sizes for GA and performance interactions are examined. With this experiment, the parameters are tried to be determined for each method for a fair comparison without explicitly biasing the algorithm performances. Despite the high performance of GA among the methods, it has been revealed that it is not a practical method due to its very high processing time. The SQP algorithm, which provides both high performance and short processing time, is chosen for the proposed model. In addition, the performances are examined with respect to different distance metrics, namely L1-Norm, L2-Norm, and SAM. The experiment has indicated that the performance of the SAM metric is better than the other two metrics. The main reason for this is that the same material can be included in different spectral signatures, which causes the performance of the algorithm to decrease during the abundance estimation. During the method development process, the time and performance change between the coarse to fine approach and the direct approach has been observed. It has been noticed that the presented coarse to fine approach has faster convergence. At the end of these experiments, a coarse to fine approach using SQP as the optimization algorithm and SAM as the distance unit is proposed.

The performance of the method is tested and compared with different noise levels and a different number of endmembers. Comparisons are made with both synthetic and real data with algorithms such as LMM, GBM, FM, MLM, and PPNM in the literature. It has been observed that these models yield successful results if synthetic data are created with their own assumptions. However, it is noticed that the performance decreases in synthetic data created with another mixing model. It has also been observed that the proposed method is more successful than all methods in cases where the data is created with a multiple mixing model. Although MLM is the most successful method among the methods in the literature in experiments with real data, the performance of the proposed coarse to fine approach exceeds that.

In the second study, a 3D convolutional encoder based algorithm is proposed for the cases where endmember information is not available. Adams, SGD, and Adagrad algorithms are used in the experiments as the optimization algorithm for the selection of the presented model. The response of these algorithms to different learning rates and batch sizes has been examined, and the results are reported. Experiments have been performed on this synthetic data created at a low noise level and two different real data. For comparison, among the successful algorithms in the literature, pure pixel-based (VCA) and non-pure pixel assumption (SISAL) algorithms are selected for endmember estimation. Endmember extracted with the proposed 3DCE model has been seen as more successful than these methods. SISAL is more successful at endmember extraction in synthetic data without pure pixels, whereas VCA is better at real data containing pure pixels. In the experiments conducted for abundance estimation, it is observed that the MLM algorithm, among the algorithms in the literature, achieves more successful results on real data. When used for abundance estimation, the performance of the algorithm presented in Chapter 5 is higher than other algorithms when using the endmembers obtained with VCA in the real data. Nevertheless, the presented 3DCE model has achieved more successful results than these algorithms in both synthetic data without pure pixels and real data. In comparisons of abundance estimation for real data, it is seen that Palsson's method obtain similar performances to other methods in the literature. As a result, the algorithm performance is 10-20% higher in comparison with the experiments performed with the selected Adam optimizer. In addition, it is observed that the proposed 3DCE method obtain better results than the baseline autoencoder-based unmixing methods in the literature. It has been validated with experiments that especially the nonlinear layer provides good performance in data with nonlinear mixtures.

As a result, it has been observed that the proposed 3DCE method is more successful when endmember information is not available, but the abundance estimation method recommended in Chapter 5 should be used when there is prior endmember signature information. The future studies can be focused on more sophisticated models including intimate mixing to better represent the interactions in complex real scenes. Another aspect that needs attention is the adaptation of other optimization algorithms to coarse to fine approach. Furthermore, although deep learning methods are used frequently in the literature, the most important feature of such methods is the performance differences with the change of the data. Besides the data change, the need for parameter changes is a very important factor. Future work in this direction, may focus on the development of nonlinear hyperspectral unmixing models that can adapt to data change for instance with Automatic Machine Learning (AutoML). Moreover, studies can also be focused into nonlinear mixing models with deep learning models.

REFERENCES

[1] E. M. Mikhail, J. S. Bethel, and J. C. McGlone, “Introduction to modern photogrammetry,” *New York*, p. 19, 2001.

[2] J. A. Richards, “Sources and characteristics of remote sensing image data,” in *Remote Sensing Digital Image Analysis*, Springer, 1993, pp. 1–37.

[3] G. Joseph, *Fundamentals of remote sensing*. Universities Press, 2005.

[4] P. K. Varshney, M. K. Arora, and others, *Advanced image processing techniques for remotely sensed hyperspectral data*. Springer Science & Business Media, 2004.

[5] J. A. Goodman, “Hyperspectral remote sensing of coral reefs: Deriving bathymetry, aquatic optical properties and a benthic spectral unmixing classification using AVIRIS data in the Hawaiian Islands.,” 1977.

[6] L. Parra, C. Spence, P. Sajda, and G. M. D. First, “Unmixing Hyperspectral Data,” *Adv. Neural Inf. Process. Syst.*, pp. 942–948, 2000.

[7] D. Heinz, C.-I. Chang, and M. L. G. Althouse, “Fully constrained least-squares based linear unmixing [hyperspectral image classification],” in *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS’99 (Cat. No. 99CH36293)*, 1999, vol. 2, pp. 1401–1403.

[8] N. Keshava, J. P. Kerekes, D. G. Manolakis, and G. A. Shaw, “Algorithm taxonomy for hyperspectral unmixing,” in *Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI*, 2000, vol. 4049, pp. 42–64.

[9] C.-I. Chang and Q. Du, “Estimation of number of spectrally distinct signal sources in hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 608–619, 2004.

[10] J. C. Harsanyi, W. Farrand, and C.-I. Chang, “Detection of subpixel spectral signatures in hyperspectral image sequences,” in *Annual Meeting, Proceedings of American Society of Photogrammetry & Remote Sensing*, 1994, pp. 236–247.

[11] J. M. Bioucas-Dias and J. M. P. Nascimento, “Hyperspectral subspace identification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, 2008.

[12] P. Chen and J. Y. Tourneret, “Toward a sparse Bayesian Markov random field approach to hyperspectral unmixing and classification,” *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 426–438, 2017, doi: 10.1109/TIP.2016.2622401.

[13] M. E. Winter, “N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data,” no. October 1999, pp. 266–275, 1999, doi: 10.1117/12.366289.

[14] J. W. Eoardmanl, F. a Kruselj, and R. O. Green, “Mapping target signatures via partial unmixing of AVIRIS data,” *Jet Propuls.*, no. 23, pp. 23–26, 1993.

[15] C. Chang, S. Member, C. Wu, and S. Member, “A New Growing Method for Simplex-Based Endmember Extraction Algorithm,” vol. 44, no. 10, pp. 2804–2819, 2006.

[16] C.-I. Chang, C.-C. Wu, and H.-M. Chen, “Random pixel purity index,” *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 2, pp. 324–328, 2009.

[17] X. Wu, B. Huang, A. Plaza, Y. Li, and C. Wu, “Real-time implementation of the pixel purity index algorithm for endmember identification on GPUs,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 5, pp. 955–959, 2013.

[18] C.-I. Chang and C.-C. Wu, “Iterative pixel purity index,” in *2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2012, pp. 1–4.

[19] J. Gu, Z. Wu, Y. Li, Y. Chen, Z. Wei, and W. Wang, “Parallel optimization of pixel purity index algorithm for hyperspectral unmixing based on spark,” in *2015 Third International Conference on Advanced Cloud and Big Data*, 2015, pp. 159–166.

[20] R. Heylen, M. A. Akhter, and P. Scheunders, “A fast alternative for the pixel purity index algorithm,” in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 1781–1784.

[21] J. M. P. Nascimento and J. M. B. Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, 2005, doi: 10.1109/TGRS.2005.844293.

[22] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, “A Convex Analysis-Based Minimum-Volume Enclosing Simplex Algorithm for Hyperspectral Unmixing,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4418–4432, 2009, doi: 10.1109/tsp.2009.2025802.

[23] J. Li and J. M. Bioucas-Dias, “Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data,” *Int. Geosci. Remote Sens. Symp.*, vol. 3, no. 1, pp. 250–253, 2008, doi: 10.1109/IGARSS.2008.4779330.

[24] J. M. Bioucas-Dias, “A variable splitting augmented Lagrangian approach to linear spectral unmixing,” *WHISPERS ’09 - 1st Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*. 2009, doi: 10.1109/WHISPERS.2009.5289072.

[25] R. Neville and K. Staenz, “Automatic endmember extraction from hyperspectral data for mineral exploration,” *Proc. 21st ...*, no. June, pp. 21–24, 1999, [Online]. Available: ftp://s5-bsc-faisan.cits.rncan.gc.ca/pub/geott/ess_pubs/219/219526/4637.pdf.

[26] E. M. T. Hendrix, I. García, J. Plaza, G. Martín, A. Plaza, and S. Member, “A New Minimum-Volume Enclosing Algorithm for Endmember Identification and Abundance Estimation in Hyperspectral Data,” vol. 50, no. 7, pp. 2744–2757, 2012.

[27] A. Ambikapathi, T. H. Chan, W. K. Ma, and C. Y. Chi, “Chance-constrained

robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11 PART 1, pp. 4194–4209, 2011, doi: 10.1109/TGRS.2011.2151197.

[28] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. F. Huntington, “ICE: A statistical approach to identifying endmembers in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2085–2095, 2004, doi: 10.1109/TGRS.2004.835299.

[29] A. Zare and P. Gader, “Sparsity promoting iterated constrained endmember detection with integrated band selection,” *Int. Geosci. Remote Sens. Symp.*, vol. 4, no. 3, pp. 4045–4048, 2007, doi: 10.1109/IGARSS.2007.4423737.

[30] S. E. Yuksel, S. Kucuk, S. Member, and P. D. Gader, “SPICEE : An Extension of SPICE for Sparse Endmember Estimation in Hyperspectral Imagery,” vol. 13, no. 12, pp. 1910–1914, 2016.

[31] B. Somers, G. P. Asner, L. Tits, and P. Coppin, “Remote Sensing of Environment Endmember variability in Spectral Mixture Analysis : A review,” *Remote Sens. Environ.*, vol. 115, no. 7, pp. 1603–1616, 2011, doi: 10.1016/j.rse.2011.03.003.

[32] R. Heylen, M. Parente, and P. Gader, “A review of nonlinear hyperspectral unmixing methods,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, 2014, doi: 10.1109/JSTARS.2014.2320576.

[33] Y. Altmann, N. Dobigeon, S. Member, and J. Tourneret, “Unsupervised Post-Nonlinear Unmixing of Hyperspectral Images Using a Hamiltonian Monte Carlo Algorithm,” vol. 23, no. 6, pp. 2663–2675, 2014.

[34] D. C. Heinz, others, and C. I. Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, 2001, doi: 10.1109/36.911111.

[35] O. Eches, N. Dobigeon, C. Mailhes, and J.-Y. Tourneret, “Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery,” *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1403–1413, 2010.

[36] J. J. Settle and N. A. Drake, “Linear mixing and the estimation of ground cover proportions,” *Int. J. Remote Sens.*, vol. 14, no. 6, pp. 1159–1177, 1993, doi: 10.1080/01431169308904402.

[37] N. Dobigeon, J.-Y. Tourneret, and C.-I. Chang, “Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2684–2695, 2008.

[38] L. Drumetz, M.-A. Veganzones, S. Henrot, R. Phlypo, J. Chanussot, and C. Jutten, “Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability,” *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3890–3905, 2016.

[39] A. Halimi, Y. Altmann, N. Dobigeon, J. Tourneret, and S. Member, “Nonlinear Unmixing of Hyperspectral Images Using a Generalized Bilinear Model,” vol. 49, no. 11, pp. 4153–4162, 2011.

[40] W. Fan, B. Hu, J. Miller, and M. Li, “Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data,” *Int. J. Remote Sens.*, vol. 30, no. 11, pp. 2951–2962, 2009, doi: 10.1080/01431160802558659.

[41] J. M. P. Nascimento and J. M. Bioucas-Dias, “Nonlinear mixture model for hyperspectral unmixing,” *Image Signal Process. Remote Sens. XV*, vol. 7477, no. September 2009, p. 74770I, 2009, doi: 10.1117/12.830492.

[42] Y. Altmann, A. Halimi, N. Dobigeon, and J. Y. Tourneret, “Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery,” *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3017–3025, 2012, doi: 10.1109/TIP.2012.2187668.

[43] B. Hapke, “Bidirectional reflectance spectroscopy: 1. Theory,” *J. Geophys. Res. Solid Earth*, vol. 86, no. B4, pp. 3039–3054, 1981.

[44] J. M. P. Nascimento and J. M. Bioucas-Dias, “Unmixing hyperspectral intimate mixtures,” *Image Signal Process. Remote Sens. XVI*, vol. 7830, no. October 2010, p. 78300C, 2010, doi: 10.1117/12.865118.

[45] R. S. Rand, R. G. Resmini, and D. W. Allen, “Modeling linear and intimate mixtures of materials in hyperspectral imagery with single-scattering albedo and kernel approaches,” *J. Appl. Remote Sens.*, vol. 11, no. 1, p. 016005, 2017, doi: 10.1117/1.jrs.11.016005.

[46] N. Dobigeon, J. Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, “Nonlinear unmixing of hyperspectral images: Models and algorithms,” *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 82–94, 2014, doi: 10.1109/MSP.2013.2279274.

[47] A. A. Nielsen, “Linear Mixture Models And Partial Unmixing in Multi- and Hyperspectral Image Data,” in *1st EARSeL Workshop on Imaging Spectroscopy*, 1998, no. 1, pp. 165–172.

[48] N. Dobigeon, L. Tits, B. Somers, Y. Altmann, and P. Coppin, “A comparison of nonlinear mixing models for vegetated areas using simulated and real hyperspectral data,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 1869–1878, 2014, doi: 10.1109/JSTARS.2014.2328872.

[49] S. Chandrasekhar, *Radiative transfer*. Courier Corporation, 2013.

[50] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.

[51] D. F. Shanno, “Conditioning of Quasi-Newton Methods for Function Minimization,” *Math. Comput.*, vol. 24, no. 111, p. 647, 2006, doi: 10.2307/2004840.

[52] J. Chen, X. Jia, W. Yang, and B. Matsushita, “Generalization of subpixel analysis for hyperspectral data with flexibility in spectral similarity measures,”

IEEE Trans. Geosci. Remote Sens., vol. 47, no. 7, pp. 2165–2171, 2009, doi: 10.1109/TGRS.2008.2011432.

[53] N. V Findler, C. Lo, and R. Lo, “Pattern search for optimization,” *Math. Comput. Simul.*, vol. 29, no. 1, pp. 41–50, 1987, doi: https://doi.org/10.1016/0378-4754(87)90065-6.

[54] C. Audet and J. E. Dennis Jr, “Analysis of generalized pattern searches,” *SIAM J. Optim.*, vol. 13, no. 3, pp. 889–903, 2002.

[55] M. Farzam, S. Beheshti, and K. Raahemifar, “Calculation of abundance factors in hyperspectral imaging using genetic algorithm,” *Can. Conf. Electr. Comput. Eng.*, pp. 837–841, 2008, doi: 10.1109/CCECE.2008.4564653.

[56] X. Tong *et al.*, “A New Genetic Method for Subpixel Mapping Using Hyperspectral Images,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, no. 9, pp. 4480–4491, 2016, doi: 10.1109/JSTARS.2015.2496660.

[57] S. Chowdhury, J. Zhang, K. Staenz, and D. Peddle, “Spectral mixture analysis of hyperspectral data using Genetic Algorithm and Spectral Angle Constraint (GA-SAC),” *Work. Hyperspectral Image Signal Process. Evol. Remote Sens.*, no. 3, pp. 1–4, 2012, doi: 10.1109/WHISPERS.2012.6874227.

[58] Y. Guijun, P. Ruiliang, Z. Chunjiang, H. Wenjiang, and W. Jihua, “Estimation of subpixel land surface temperature using an endmember index based technique: A case examination on ASTER and MODIS temperature products over a heterogeneous area,” *Remote Sens. Environ.*, no. 115, pp. 1202–1219, 2011.

[59] B. S. Penn, “Using simulated annealing to obtain optimal linear end-member mixtures ofhyperspectral data,” *Comput. Geosci.*, vol. 28, no. 1, pp. 809–817, 2002, doi: 10.1016/S0006-8993(99)01227-5.

[60] A. Villa, J. Chanussot, J. A. Benediktsson, M. Ulfarsson, and C. Jutten, “Super-Resolution: An efficient method to improve spatial resolution of hyperspectral images,” *Int. Geosci. Remote Sens. Symp.*, pp. 2003–2006, 2010, doi: 10.1109/IGARSS.2010.5654208.

[61] P. Debba, “Abundance estimation of spectrally similar minerals,” *Int. Geosci. Remote Sens. Symp.*, vol. 2, pp. II-298-II–301, 2009, doi: 10.1109/IGARSS.2009.5418069.

[62] S. P. Boyd and L. Vandenberghe, “Convex optimization,” vol. 1, pp. 147–168, 2004.

[63] P. T. Boggs and J. W. Tolle, “Sequential Quadratic Programming,” *Acta Numer.*, vol. 4, no. January, pp. 1–51, 1995, doi: 10.1017/S0962492900002518.

[64] R. H. Byrd, J. Nocedal, and Y.-X. Yuan, “Global convergence of a cass of quasi-Newton methods on convex problems,” *SIAM J. Numer. Anal.*, vol. 24, no. 5, pp. 1171–1190, 1987.

[65] R. Hooke and Jeeves T.A., “‘Direct search’ solution of numerical and

statistical problems,” *J. Assoc. Comput. Mach.*, vol. 8, pp. 212–229, 1961.

[66] T. P. Teng, “Distress Identification Manual for the Long-Term Pavement Performance Program,” no. June, 2003.

[67] J. C. Spall, *Stochastic Optimization*. 2012.

[68] D. E. Goldberg, “Genetic algorithms in search,” *Optim. Mach.*, 1989.

[69] Z. Ben Rabah, I. R. Farah, G. Mercier, and B. Solaiman, “A new method to change illumination effect reduction based on spectral angle constraint for hyperspectral image unmixing,” *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 6, pp. 1110–1114, 2011, doi: 10.1109/LGRS.2011.2157890.

[70] J. Verhoeye and R. De Wulf, “Land cover mapping at sub-pixel scales using linear optimization techniques,” *Remote Sens. Environ.*, vol. 79, no. 1, pp. 96–104, 2002, doi: 10.1016/S0034-4257(01)00242-5.

[71] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[72] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[73] P. Debba, E. J. M. Carranza, F. D. Van Der Meer, and A. Stein, “Abundance estimation of spectrally similar minerals by using derivative spectra in simulated annealing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3649–3658, 2006, doi: 10.1109/TGRS.2006.881125.

[74] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[75] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[76] X. Yao, “Evolving artificial neural networks,” *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[77] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 37–49.

[78] Y. Burda, R. Grosse, and R. Salakhutdinov, “Importance weighted autoencoders,” *arXiv Prepr. arXiv1509.00519*, 2015.

[79] A. Ng and others, “Sparse autoencoder,” *CS294A Lect. notes*, vol. 72, no. 2011, pp. 1–19, 2011, doi: 10.1088/0953-4075/31/3/001.

[80] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv Prepr. arXiv1511.05644*, 2015.

[81] J. Masci, U. Meier, D. Cire\csan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in *International conference on artificial neural networks*, 2011, pp. 52–59.

[82] P. Y. Simard, D. Steinkraus, J. C. Platt, and others, “Best practices for convolutional neural networks applied to visual document analysis.,” in *Icdar*, 2003, vol. 3, no. 2003.

[83] A. G. Howard *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv Prepr. arXiv1704.04861*, 2017.

[84] A. Vedaldi and K. Lenc, “Matconvnet: Convolutional neural networks for matlab,” in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 689–692.

[85] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems,* 2012, pp. 1097–1105.

[86] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv Prepr. arXiv1408.5882,* 2014.

[87] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[88] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–14.

[89] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” pp. 818–833, 2014.

[90] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[91] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, “Understanding Batch Normalization,” no. NeurIPS, 2018.

[92] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, “Recurrent batch normalization,” in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings,* 2017, no. Section 3, pp. 1–13.

[93] G. Hinton, “Dropout : A Simple Way to Prevent Neural Networks from Overfitting,” vol. 15, pp. 1929–1958, 2014.

[94] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions : Comparison of Trends in Practice and Research for Deep Learning,” pp. 1–20.

[95] J. Turian, J. Bergstra, and Y. Bengio, “Quadratic Features and Deep Architectures for Chunking,” no. June, pp. 245–248, 2009.

[96] B. Karlik and O. Vehbi, “Performance Analysis of Various Activation Functions in Artificial Neural Networks,” in *Journal of Physics: Conference Series*, 2019, vol. 1237, no. 2, doi: 10.1088/1742-6596/1237/2/022030.

[97] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” 2010.

[98] B. Zoph and Q. V Le, “Searching for activation functions,” in *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018, pp. 1–13.

[99] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[100] C. Xing, L. Ma, and X. Yang, “Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images,” *J. Sensors*, vol. 2016, 2016, doi: 10.1155/2016/3632943.

[101] C. Mellon and U. C. Berkeley, “Tutorial on Variational Autoencoders,” pp. 1–23, 2016.

[102] H. C. H.-C. Shin, M. R. R. Orton, D. J. J. Collins, S. J. J. Doran, and M. O. O. Leach, “Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, 2013, doi: 10.1109/TPAMI.2012.277.

[103] J. Chen, W. Zhu, T. Yong, Q. Yu, Y. Zheng, and L. Huang, “Hyperspectral anomaly detection based on stacked denoising autoencoders,” *J. Appl. Remote Sens.*, vol. 11, no. 3, p. 036007, 2017, doi: 10.1117/1.JRS.11.

[104] S. Boutami, N. Zhao, and S. Fan, “Determining the optimal learning rate in gradient-based electromagnetic optimization using the Shanks transformation in the Lippmann--Schwinger formalism,” *Opt. Lett.*, vol. 45, no. 3, pp. 595–598, 2020.

[105] Y. Zhu, Y. Chen, D. Huang, B. Zhang, and H. Wang, “Automatic, Dynamic, and Nearly Optimal Learning Rate Specification by Local Quadratic Approximation,” *arXiv Prepr. arXiv2004.03260*, 2020.

[106] S.-B. Lin, Y. Lei, and D.-X. Zhou, “Boosted Kernel Ridge Regression: Optimal Learning Rates and Early Stopping.,” *J. Mach. Learn. Res.*, vol. 20, pp. 41–46, 2019.

[107] T. Moncur, “Optimal Learning Rates for Neural Networks,” 2020.

[108] P. Netrapalli, “Stochastic Gradient Descent and Its Variants in Machine Learning,” *J. Indian Inst. Sci.*, vol. 99, no. 2, pp. 201–213, 2019, doi: 10.1007/s41745-019-0098-4.

[109] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[110] L. I. Jimenez *et al.*, “GPU Implementation of Spatial-Spectral Preprocessing for Hyperspectral Unmixing,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 11, pp. 1671–1675, 2016, doi: 10.1109/LGRS.2016.2602227.

[111] S. Ozkan and G. B. Akar, “Deep Spectral Convolution Network For

Hyperspectral Unmixing," pp. 3313–3317, 2018.

[112] X. Xu, Z. Shi, and B. Pan, "A supervised abundance estimation method for hyperspectral unmixing," *Remote Sens. Lett.*, vol. 9, no. 4, pp. 383–392, 2018, doi: 10.1080/2150704X.2017.1415471.

[113] Y. Su, A. Marinoni, J. Li, J. Plaza, and P. Gamba, "Stacked nonnegative sparse autoencoders for robust hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1427–1431, 2018, doi: 10.1109/LGRS.2018.2841400.

[114] Y. Su, A. Marinoni, J. Li, A. Plaza, and P. Gamba, "Nonnegative sparse autoencoder for robust endmember extraction from remotely sensed hyperspectral images," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 205–208.

[115] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "DAEN: Deep Autoencoder Networks for Hyperspectral Unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4309–4321, 2019, doi: 10.1109/TGRS.2018.2890633.

[116] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and Y. Huang, "Deep Auto-Encoder Network For Hyperspectral Image Unmixing," *IGARSS 2018 - 2018 IEEE Int. Geosci. Remote Sens. Symp.*, pp. 6400–6403, 2018, doi: 10.1109/IGARSS.2018.8519571.

[117] Y. Qu, R. Guo, and H. Qi, "Spectral unmixing through part-based non-negative constraint denoising autoencoder," *Int. Geosci. Remote Sens. Symp.*, vol. 2017-July, no. July, pp. 209–212, 2017, doi: 10.1109/IGARSS.2017.8126931.

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

[119] B. Yan, Z. Wu, H. Liu, Y. Xu, and Z. Wei, "Hyperspectral Unmixing Via Wavelet Based Autoencoder Network," in *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2019, pp. 1–5.

[120] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[121] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[122] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?," *IEEE Trans. image Process.*, vol. 24, no. 12, pp. 5017–5032, 2015.

[123] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," *Tech. Univ. Denmark*, vol. 5, 2012.

[124] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson,

"Hyperspectral Unmixing Using a Neural Network Autoencoder," *IEEE Access*, vol. 6, pp. 25646–25656, 2018, doi: 10.1109/ACCESS.2018.2818280.

[125] X. R. Feng, H. C. Li, J. Li, Q. Du, A. Plaza, and W. J. Emery, "Hyperspectral unmixing using sparsity-constrained deep nonnegative matrix factorization with total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6245–6257, 2018, doi: 10.1109/TGRS.2018.2834567.

[126] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," vol. 401, no. October 1999, pp. 788–791, 1999.

[127] M. M. Elkholy, M. Mostafa, H. M. Ebied, and M. F. Tolba, "Hyperspectral unmixing using deep convolutional autoencoder," *Int. J. Remote Sens.*, vol. 41, no. 12, pp. 4797–4817, 2020, doi: 10.1080/01431161.2020.1724346.

[128] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Convolutional Autoencoder For Spatial-Spectral Hyperspectral Unmixing," *IGARSS 2019 - 2019 IEEE Int. Geosci. Remote Sens. Symp.*, no. 978, pp. 357–360, 2019.

[129] S. Ozkan and G. B. Akar, "Improved Deep Spectral Convolution Network For Hyperspectral Unmixing With Multinomial Mixture Kernel and Endmember Uncertainty," no. 1, pp. 1–11, 2018, [Online]. Available: http://arxiv.org/abs/1808.01104.

[130] R. N. Clark *et al.*, "USGS digital spectral library splib06a," *US Geol. Surv. Digit. data Ser.*, vol. 231, p. 2007, 2007.

[131] F. Zhu, "Hyperspectral Unmixing: Ground Truth Labeling, Datasets, Benchmark Performances and Survey," *arXiv Prepr. arXiv1708.05125*, 2017.

[132] F. A. Kruse, A. B. Lefkoff, J. W. H. Boardman, A. T. Shapiro, P. J. Barloon, and A. F. H. Goetz, "The Spectral Image Processing System (SIPS)," 1992.

[133] C.-I. Chang and Chein-I Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1927–1932, 2002, doi: 10.1109/18.857802.

[134] V. Granville, M. Kfivanek, and J. Rasson, "Simulated Annealing: A Proof of Convergence," vol. 16, no. 6, pp. 652–656, 1994.

[135] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014, doi: 10.1109/JSTARS.2014.2329330.

[136] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1026–1034, 2015, doi: 10.1109/ICCV.2015.123.

[137] A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.

[138] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy,

“On large-batch training for deep learning: Generalization gap and sharp minima,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–16, 2017.

[139] P. Doll, R. Girshick, and P. Noordhuis, “Accurate, large minibatch SGD,” *Arxiv*, 2017.

[140] E. Hoffer, I. Hubara, and D. Soudry, “Train longer, generalize better: Closing the generalization gap in large batch training of neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, pp. 1732–1742, 2017.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	: Özdemir, Okan Bilge
Nationality	: Turkish
Marital Status	: Married
Phone	: 0 544 870 55 33
e-mail	: oozdemir@metu.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
MS	Information Systems, *METU*	2013
BS	Computer Engineering, Hacettepe University	2016
BS	Computer Education, *Muğla University*	2009
High School	Balgat Anadolu Teknik Lisesi, Ankara	2002

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2019-Present	Information Systems, Artvin Çoruh University	Research Assistant
2009-2019	Information Systems, METU	Research Assistant

PROJECTS

Year	**Name**
2020	Hyperspectral Gas Detection with Deep Learning
2020	Illegal Use Detection and Consumption Estimation System with Image Processing
2018	Hyperspectral Chemical Substance Imaging and Automated Detection Phase 2
2017	Hyperspectral Chemical Substance Imaging and Automated Detection Phase 1
2017	Investigation of Archaeological Remains in Hyperspectral

	Images
2016	Development of Mobile Malware Analysis System
2015	TUYGUN
2015	Image Enhancement and Object Classification in Hyperspectral Images
2014	Material Detection with Hyperspectral Images

PUBLICATIONS

- Özdemir, O.B.; Soydan, H.; Yardımcı Çetin, Y.; Düzgün, H.Ş. Neural Network Based Pavement Condition Assessment with Hyperspectral Images. Remote Sens. 2020, 12, 3931.

- Özdemir, O. B., & Çetin, Y. Y. (2017, May). Improved hyperspectral vegetation detection using neural networks with spectral angle mapper. In Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII (Vol. 10190, p. 101901A). International Society for Optics and Photonics.

- Özdemir, O. B., Soydan, H., Çetin, Y. Y., & Düzgün, Ş. (2016, July). Hyperspectral unmixing based vegetation detection with segmentation. In Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International (pp. 6998-7001).

- Soydan, Hilal, H. Şebnem Düzgün, and Okan Bilge Özdemir. "Analysis of land use land cover changes for an abandoned mine site." Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. IEEE, 2015.

- Özdemir, O. B., Soydan, H., Çetin, Y. Y., & Düzgün, H. Ş. (2015, May). Signature based vegetation detection on hyperspectral images. In 2015 23nd Signal Processing and Communications Applications Conference (SIU) (pp. 2501-2504). IEEE.

- Fatih Omruuzun, Okan Bilge Ozdemir, and Yasemin Yardimci Cetin, Spel: Development Of A New Spectral Signature Library For Food Products, Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry Petra Perner (Editor) 6th International Conference, MDA 2014 St. Petersburg, Russia

- Ozdemir, O.B., Gedik E., Cetin, Y.Y., “Hyperspectral Classification Using Stacked Autoencoders with Deep Learning”, Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2014.

- Ozdemir, O. B., & Cetin, Y. Y. (2014, April). Improvement of hyperspectral classification accuracy with limited training data using meanshift segmentation. In Signal Processing and Communications Applications Conference (SIU), 2014 22nd (pp. 1794-1797). IEEE.

- Ozdemir, O. B., & Cetin, Y. Y. (2013, April). The effect of training data on hyperspectral classification algorithms. In Signal Processing and Communications Applications Conference (SIU), 2013 21st (pp. 1-4). IEEE.

- Ozdemir, O.B.; Cetin, Y.Y., “Improvements On Hyperspectral Classification Algorithms”, Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2013.