LEARNING-BASED METHODS FOR MULTI-MODAL AND MULTI-SPECTRAL DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAVAŞ ÖZKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2020

Approval of the thesis:

LEARNING-BASED METHODS FOR MULTI-MODAL AND MULTI-SPECTRAL DATA

submitted by **SAVAŞ ÖZKAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Gözde Bozdağı Akar
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU _____

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU _____

Assist. Prof. Dr. Osman Serdar Gedik
Computer Engineering, Yıldırım Beyazıt University _____

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

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering, METU _____

Date: 21.09.2020

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, Surname: Savaş Özkan

Signature :

ABSTRACT

LEARNING-BASED METHODS FOR MULTI-MODAL AND MULTI-SPECTRAL DATA

Özkan, Savaş
Ph.D., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Gözde Bozdağı Akar

September 2020, 122 pages

Data-driven solutions have become essential parts of our daily lives. These solutions generally consume a large amount of supervised data. Additionally, a large-body of learnable parameters must be trained to be able to reach the level of human knowledge accurately. However, these dependencies can be overcome by making full use of domain-specific features via specialized learning structures.

This thesis addresses unsupervised and multimodal data by utilizing different sensor types and application domains in particular. Our first direction is to prove that robust representations can be computed from data without supervision if data-specific features can be appropriately modeled. For this purpose, multispectral data is used, which provides rich content information. Spectral material signatures and their fractional abundances are obtained from the data blindly. The second direction is to show that multimodal data offers theoretical and practical benefits to solve complicated visual tasks. In particular, we propose a novel NN model that is able to segment biomedical data from diverse visual modalities. The superiority is that no information about data modalities is provided at both train and test phases. This feature inevitably improves the applicability of this model. Furthermore, the number of learnable pa-

rameters is decreased since the solution is combined within a single model. For this purpose, we present novel modifications in the NN structure, and GAN is used that aims to cluster and adapt domains for different modalities. Lastly, we apply multimodal learning for the regression optimization of multispectral data, where a large amount of data is mainly required for more accurate solutions. This limitation is eliminated by solving the regression and the classification tasks concurrently. Similarly, experimental results show that both methods significantly improve the performance compared to all baselines.

Keywords: Unsupervised Learning, Multimodal Data, Multispectral Data, Biomedical Data

ÖZ

ÇOKLU-MODEL VE ÇOKLU-SPEKTRAL VERİLERDE ÖĞRENME TABANLI YÖNTEMLER

Özkan, Savaş
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar

Eylül 2020 , 122 sayfa

Veriye dayalı çözümler günlük hayatımızın vazgeçilmez bir parçası haline geldi. Bu çözümler genellikle büyük miktarda denetlenen veri tüketmektedir. Ek olarak, insan bilgi düzeyine doğru bir şekilde ulaşabilmek için geniş bir öğrenilebilir parametreler bütünü eğitilmelidir. Bununla birlikte, bu bağımlılıklar, özel öğrenme yapıları aracılığıyla alana özgü özelliklerin tam olarak kullanılmasıyla aşılabilir.

Bu tez, özellikle farklı sensör tiplerini ve uygulama alanlarını kullanarak denetimsiz ve çok modlu verileri ele almaktadır. İlk yönümüz, veriye özgü özellikler uygun şekilde modellenebilirse, sağlam temsillerin denetim olmaksızın verilerden hesaplanabileceğini kanıtlamaktır. Bu amaçla, zengin içerik bilgisi sağlayan multispektral veriler kullanılır. Spektral malzeme imzaları ve bunların karışım oranları verilerden kör olarak elde edilirmektedir. İkinci yön, çok modlu verilerin karmaşık görsel görevleri çözmek için teorik ve pratik faydalar sunduğunu göstermektir. Özellikle, biyomedikal verileri çeşitli görsel modalitelerden bölümlere ayırabilen yeni bir NN modeli öneriyoruz. Üstünlük, hem eğitim hem de test aşamalarında veri modaliteleri hakkında

hiçbir bilginin sağlanmamasıdır. Bu özellik, kaçınılmaz olarak bu modelin uygulanabilirliğini artırmaktadır. Ayrıca çözüm tek bir modelde birleştirildiği için öğrenilebilir parametrelerin sayısı azalmıştır. Bu amaçla, NN yapısında yeni modifikasyonlar sunuyoruz ve farklı modaliteler için alanları kümelemeyi ve uyarlamayı amaçlayan GAN kullanılır. Son olarak, daha doğru çözümler için büyük miktarda veriye ihtiyaç duyulan multispektral verilerin regresyon optimizasyonu için multimodal öğrenmeyi uygulanmaktadır. Regresyon ve sınıflandırma görevlerini aynı anda çözerek bu sınırlama ortadan kaldırılır. Benzer şekilde, deneysel sonuçlar, her iki yöntemin de bütün önceki yöntemlere kıyasla performansı önemli ölçüde artırdığını göstermektedir.

Anahtar Kelimeler: Denetimsiz Öğrenme, Çoklu Modlu Veriler, Multispektral Veriler, Biyomedikal Veriler

To my family...

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Prof. Dr. Gözde Bozdağı Akar for her tremendous patience, wisdom, support and guidance in the first two stages of my research life, master and doctorate studies. I am grateful for her technical knowledge and helps that she generously shared with me. Probably, I could not be who I am today professionally if she was not my supervision. I will always remember that it has been a pleasure and a chance for me to work with her.

I would like to express my gratitude to Prof. Dr. İlkay Ulusoy and Assist. Prof. Dr. Osman Serdar Gedik for their valuable comments that help to improve the quality of the thesis. Furthermore, I would like to thank Assoc. Prof. Dr. Seniha Esen Yüksel and Assist. Prof. Dr. Elif Vural, for being a part of my examiner committee and their valuable feedbacks.

I would like to thanks Tuğrul Ateş, Dr. Engin Tola, Dr. Ersin Esen and Dr. Medeni Soysal, who are contributed to me to enlarge a huge backpack of knowledge on various technical fields. I would like to thanks to all my friends in TUBITAK UZAY, METU-Multimedia Lab, STM and FORD-OTOSAN, who created a joyful environment in this period. Lastly, I have special thanks for Tayfun Ateş, who regularly loses the FIFA matches and still does not give up :).

Last but not least, I would like to express my gratitude for my family, Serap, Arif and Akın. They have always been there for me with their support and patience. Undoubtedly, this thesis is dedicated to them for their unconditional love.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xvi

LIST OF FIGURES xviii

LIST OF ABBREVIATIONS xxi

CHAPTERS

1 INTRODUCTION 1

1.1 Background and Motivation 1

1.2 Contributions for These Questions 3

1.2.1 Learning representations with no supervision 4

1.2.2 Learning representations from multimodal tasks 6

1.2.3 Restructuring learning models for different domains 8

1.3 The Outline of the Thesis 8

2 FOUNDATIONS 11

2.1 History of Artificial Neural Networks 11

2.2 Foundations of Deep Models 13

2.2.1 Learning-Models 14
2.2.2 Cost-Functions 15
2.2.3 Gradient-Based Learning 16
2.2.4 Dense Layers 17
2.2.5 Convolution Layers 18
2.2.6 Activation Layers 19
2.2.7 Regularization Layers 21
2.2.8 Normalization Layers 21
2.2.9 Pooling Layers 22
2.3 Deep Network Research 22
2.3.1 AutoEncoder Model 23
2.3.2 Adversarial Model 25
2.3.3 Skip-Connections 26
2.3.4 Attention Model 27
3 SPECTRAL UNMIXING AND ENDMEMBER EXTRACTION 29
3.1 Literature Review 32
3.1.1 Endmember Extraction 32
3.1.2 Spectral Unmixing 33
3.2 Sparse Auto-Encoder Network for Endmember Extraction 34
3.2.1 Zero-biased Filters 37
3.2.2 Sparsity and Nonlinearity 37
3.2.3 Robust Latent Representations 39
3.2.4 Parameter Learning 42

3.2.5 Extra Step for Overfitting and Generalization 43
3.2.6 Fractional Abundance Estimation 43
3.3 Deep Spectral Convolution Network for Multi-Spectral Unmixing . . 44
3.3.1 Stack of Spectral Convolutions 45
3.3.2 Multinomial Mixture Kernel For Abundance Maps 48
3.3.3 Parameter Updates for Multinomial Mixture Kernel 50
3.3.4 Extra Steps for Spectral Variability 52
3.3.5 Implementation Details 53
3.4 Experiments 55
3.4.1 Multispectral Datasets 55
3.4.1.1 Synthetic Data 56
3.4.1.2 Urban Data 56
3.4.1.3 Jasper Data 56
3.4.1.4 Samson Data 57
3.4.1.5 Cuprite Data 57
3.4.1.6 University of Pavia Data 57
3.4.1.7 Mississippi Gulfport Data 57
3.4.2 Metrics 58
3.4.3 Baselines 58
3.4.4 Sparse Autoencoder Network for Endmember Extraction (Endnet) 59
3.4.4.1 Parameter Setting 59
3.4.4.2 Computational Complexity 61

3.4.4.3 Tests on Datasets 61

3.4.5 Deep Spectral Convolution Network for Multi-Spectral Unmixing (DSCN++) 65

3.4.5.1 Parameter Setting 65

3.4.5.2 Test on Datasets 72

4 MULTIMODAL LEARNING 77

4.1 Literature Review 78

4.1.1 Semantic Segmentation 79

4.1.2 CrossModal Learning 80

4.1.3 MultiModal Learning 81

4.2 MultiModal Learning For Biomedical Data 82

4.2.1 Proposed Method 83

4.2.2 DCN Structure 85

4.2.3 Parameter Learning 86

4.2.4 Data Preparation and Post-Processing 89

4.3 Multi-Modal Learning For MultiSpectral Data 89

4.3.1 Proposed Method 90

4.3.2 Parameter Learning 91

4.4 Experiments 91

4.4.1 Biomedical Data 92

4.4.2 Visible-Range Data 94

4.4.3 Multispectral Data 94

4.4.4 Evaluation Metric 95

4.4.5 MultiModal For Biomedical Data 96

4.4.6 MultiModal For Visible-Range Data 98

4.4.7 Multi-Modal For MultiSpectral Data 99

5 CONCLUSION 105

5.1 Conclusions and Limitations 105

5.2 Future Directions 107

REFERENCES 109

LIST OF TABLES

TABLES

Table 3.1 Structure of DSCN++. Conv1D, SN and BN denote 1D convolution, spectral normalization and batch normalization respectively. 46

Table 3.2 Structure of discriminative network. 52

Table 3.3 Average computation time for methods on three datasets. 60

Table 3.4 SAD and RMSE results on Urban dataset. Mean and standard deviation are reported. Best results are illustrated in bold. 66

Table 3.5 SAD and RMSE results on Samson dataset. Mean and standard deviation are reported. Best results are illustrated in bold. 67

Table 3.6 SAD and RMSE results on Jasper Ridge dataset. Mean and standard deviation are reported. Best results are illustrated in bold. 68

Table 3.7 SAD results on Cuprite dataset. Mean and standard deviation are reported. Best results are illustrated in bold. 69

Table 3.8 SAD results on University of Pavia dataset. Mean and standard deviation are reported. Best results are illustrated in bold. 70

Table 3.9 SAD results on Mississippi Gulfport dataset. Mean and standard deviation are reported. Best results are illustrated in bold. 71

Table 3.10 RMSE performance on synthetic dataset for different configurations. Best results are indicated as Bold. 72

Table 3.11 RMSE performance on synthetic dataset for baselines. Best results are indicated as Bold-Red and second best results are shown as Bold. . . . 73

Table 3.12 RMSE performance on Urban dataset for different configurations. Best results are indicated as Bold. 74

Table 3.13 RMSE performance on Urban dataset for baselines. Best results are indicated as Bold-Red and second best results are shown as Bold. 75

Table 3.14 RMSE performance on Jasper dataset for different configurations. Best results are indicated as Bold. 76

Table 3.15 RMSE performance on the Jasper dataset for baselines. Best results are indicated as Bold-Red and second best results are shown as Bold. . . . 76

Table 4.1 Percentage of responses forwarding to next layers. 86

Table 4.2 Performance for individual modalities and a multimodal method. . . 97

Table 4.3 Precision and IOU results on Covid-19 dataset for multimodal segmentation. 98

Table 4.4 Precision results for Mnist and SVHN datasets. 99

Table 4.5 Performance for multimodal segmentation. Red bold indicates best performance for each metric. 100

Table 4.6 Performance for multimodal segmentation with different T values. Red bold indicates best performance for each metric. 101

Table 4.7 RMSE performance for multispectral data with semantic content. Note that RMSE performance is normalized with $\times 10^{-2}$. 102

Table 4.8 RMSE performance for multispectral data with semantic content and sparse lidar data. Note that RMSE performance is normalized with $\times 10^{-2}$. 102

LIST OF FIGURES

FIGURES

Figure 1.1 Chart depicts the sub-fields we have concentrated on in this thesis. Indeed, our motivation is to learn robust representations from data $\mathbf{x}$ by optimizing a function $f(.)$ with a set of learnable parameters θ. Here, X is an input space. Z and Y indicate latent representation spaces that are learned from data. Furthermore, c and t denote the indicators of data modality and task, respectively. In particular, we observe that solving a single task for one data modality as in unsupervised learning is not sufficient to handle all cases in the literature. Hence, we also consider multiple modalities and problem definitions to compute robust representations in this thesis. For a sub-field, different application domains with a variety of sensor types are selected. 3

Figure 2.1 The response of Sigmoid and ReLU functions. 20

Figure 3.1 Visualization of learnable parameters for conventional autoencoder structure. 36

Figure 3.2 Distribution characteristic of batch normalization layer. 38

Figure 3.3 Normalized similarity score distributions for same (blue) and coherent (red) materials. Results for inner product (first row) and SAD (second row) are illustrated. 40

Figure 3.4 Spectral signatures from Jasper (first row) and Urban (second row) data correspond to two different pure materials. 41

Figure 3.5 Four multispectral signatures that are corrupted with the concept of Denoising Autoencoder. 43

Figure 3.6 Flow of deep spectral convolution network for spectral unmixing. 44

Figure 3.7 Data distributions before (first column) and after (second column) applying DSCN++ method. Each color indicates the abundance maps of different pure materials (i.e., red, green, blue or black). Observe that our model maps the high-dimensional material signatures to a more separable space based on their abundances. 47

Figure 3.8 Color images of Synthetic, Urban, Samson, Jasper, Cuprite, University of Pavia and Mississippi Gulfport data, respectively 55

Figure 3.9 Abundance maps for Synthetic dataset. 56

Figure 3.10 Abundance maps for Urban dataset. First, second and third rows indicate ground truth annotations, predictions and pixelwise differences for each material. Pure material signatures for this scene are given in 3.13. 61

Figure 3.11 Abundance maps for Samson dataset. First, second and third rows indicate ground truth annotations, predictions and pixelwise differences for each material. Pure material signatures for this scene are given in 3.13. 62

Figure 3.12 Abundance maps for Jasper dataset. First, second and third rows indicate ground truth annotations, predictions and pixelwise differences for each material. Pure material signatures for this scene are given in 3.13. 62

Figure 3.13 Ground truth (blue) and estimated (red) endmember signatures for Urban (first row), Samson (second row) and Jasper (third row) datasets. 63

Figure 3.14 Abundance maps for Cuprite dataset. 63

Figure 3.15 Abundance maps for Pavia dataset. The outputs of three methods are illustrated as Endnet, SCM and SPICE. As noticed, our methods computes 8 materials instead of 7 compared to the baselines. Details can be found in the text. 64

Figure 3.16 Abundance maps for Mississippi Gulfport dataset. The outputs of three methods are illustrated as Endnet, SCM and SPICE. 65

Figure 3.17 Spectra of pixels from the Synthetic dataset. 73

Figure 4.1 Overall flow for multimodal data for biomedical data. 84

Figure 4.2 Response distributions of CT (blue) and MR (yellow) data for pre-structure (a) and post-structure. 88

Figure 4.3 Overall flow for multimodal data. 89

Figure 4.4 Visual data for Liver and Covid-19 datasets. (a) Liver CT,(b) Liver MR (c) Covid-19 CT 92

Figure 4.5 Samples and ground truth annotations for CT and MR data from different slices. 93

Figure 4.6 Visual data for MNIST and SVHN datasets. 94

Figure 4.7 RGB data, ground truth depth data and estimated depth data with multimodal optimization. 103

Figure 4.8 RGB data, ground truth depth data and estimated depth data with multimodal optimization and sparse data. 104

LIST OF ABBREVIATIONS

ABBREVIATIONS	
NN	Neural Networks
DCN	Deep Convolution Networks
DSCN	Deep Spectral Convolution Network
GAN	Generative Adversarial Network

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

In the last decades, unstructured data collected from different sensors and platforms becomes accessible to the public. An important concern is that the human perception system is limited to extract tractable patterns from data within feasible time instances. This concern multiplies when the complexity of input space covering multiple modalities and application domains is taken into account.

For this purpose, the availability of a large collection of data is altered in the ways how we learn, seek, and solve. To handle a large amount of data, smarter tools need to be incorporated in the quest to replace humans with more powerful computing units -i.e., computers and robots-. In this manner, these tools are used to compute the parameterized hypothesis classes, and large parameter sets are optimized to reach human-level intelligence. High-dimensional data is converted step-by-step with non-linear transformations into low-dimensional spaces. After all, this step enhances the learning capacity, known as representation learning. Indeed, this replacement is stimulated by the advent of general-purpose computing units in our daily lives in which fast and local hardware accesses can be done.

With all of the above in mind, data-driven solutions have recently been an essential and self-evident part of collectively distillations of patterns from data. Undoubtedly, data must be packed and annotated by our, the humans, expectations. In general, learning-based methods have become more critical due to their superiority for complex scenarios. The principal objective of these methods is to extract robust representations to work for seen and unseen cases with no or limited annotated data. Hence,

the generalization ability of these methods is decisive.

Despite the widespread use of learning-based methods, there is no perfect solution or model structure that answers all the questions raised from different tasks. For instance, data-driven solutions rely heavily on extensive and supervised annotations to unveil latent representations from data. Even if all of these steps are perfectly defined, sometimes the data itself may not be sufficient to compute robust features. The impacts of all these issues are discussed in *unsupervised learning* and *multimodal learning* in the literature.

In the scope of this thesis, unsupervised learning and multimodal learning are thoroughly analyzed to compute robust representations from data. Accurately, it is positioned for different sensor types and application domains to confirm the superiority of our contributions. In the end, we opt to find proper answers to the following questions:

- *How can we compute robust representations from data without supervision but make full use of domain-specific features?*

- *How can we learn robust representations from multimodal data that can be insufficient or impractical when a single learning model is used?*

- *How can we specialize the structure of a learning model that has a positive impact on the performance for different domains and sensor types?*

All of these questions will be answered throughout the thesis with precisely developed algorithms. Ultimately, these methods have structural and learning-based improvements for a variety of sensor types and problem definitions. Moreover, the theoretical foundations of these algorithms are discussed in detail. Note that we opt to compute the best representations from data under severe challenges that we have mentioned. Last but not least, we validate the superiority of our contributions in the experiments on well-studied datasets.

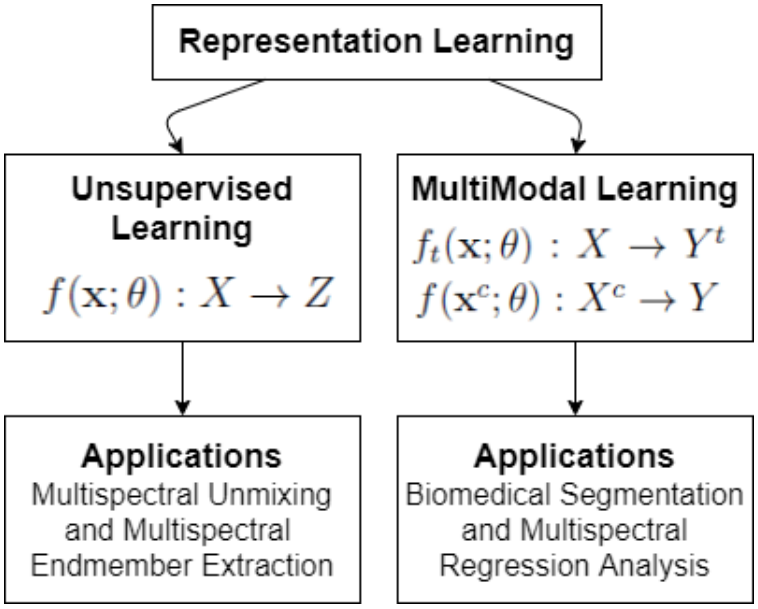

Figure 1.1: Chart depicts the sub-fields we have concentrated on in this thesis. Indeed, our motivation is to learn robust representations from data $\mathbf{x}$ by optimizing a function $f(.)$ with a set of learnable parameters θ. Here, X is an input space. Z and Y indicate latent representation spaces that are learned from data. Furthermore, c and t denote the indicators of data modality and task, respectively. In particular, we observe that solving a single task for one data modality as in unsupervised learning is not sufficient to handle all cases in the literature. Hence, we also consider multiple modalities and problem definitions to compute robust representations in this thesis. For a sub-field, different application domains with a variety of sensor types are selected.

1.2 Contributions for These Questions

As mentioned, the primary motivation of this thesis is to find proper answers to some of the essential questions that are recognized in the literature. These questions are concentrated on representation learning problem of course. For this purpose, we first provide the details about these questions and describe their challenges. Later, our contributions and novelties are reported, and the outline of the thesis is drawn.

In particular, Fig. 1.1 illustrates the sub-fields that we have concentrated in this thesis. Indeed, the main motivation is to learn robust representations from data by optimizing a function $f(.)$ with a set of learnable parameters θ. Unlike supervised (i.e., classi-

fication) tasks, we select two different fields: unsupervised learning (Chapter 3) and multimodal learning (Chapter 4). In the following parts, we will describe their challenges and our contributions.

1.2.1 Learning representations with no supervision

Even if the accessibility of unstructured data is increased like never before, the scarcity of supervision is still one of the drawbacks for data-driven solutions. The critical factor for solving this issue is that tractable patterns are computed from data with no supervision. For instance, if we correctly define the notion of cumulative rewards and penalties for computers to perform specific purposes, they naturally take actions to find optimum solutions as humans would. As noticed, the correct formulation of data is of the critical step to reach ideal solutions.

As expected, many studies are applied to various unsupervised learning applications in the literature. For the survey of this field, [1] explains the essential details. In particular, two approaches are exploited in the scope of this thesis: *auto-encoder model* [2] and *generative model* [3, 4]. Autoencoder models aim to learn robust representations roughly by resembling the original data. A distinct feature of this method is that most common distributions are decoded with parameter sets, while noisy assumptions are automatically removed from the solutions.

On the other hand, generative models align the distributions of actual and predicted data. For this purpose, it incorporates a discriminative model. This model inherently pushes the actual distributions far from the predicted ones. On the other hand, the original model fools the discriminative model by producing similar predictions. To this end, robust latent representations are estimated. Even if these methods have their advantages, there is no conclusion that one method is universally superior to the other. Note that we will discuss these advantages and disadvantages in the next chapters.

Mathematically, let $\mathbf{x}$ denotes a unimodal high-dimensional observed variable (e.g., image, video or text) and an empirical distribution is given as $p_d(\mathbf{x})$ over an input space X. Also, we define a latent variable $\mathbf{z} \in Z$ over a latent space Z. By using Bayes formula, the joint distribution can be factorized as $q(\mathbf{x}, \mathbf{z}) = p_d(\mathbf{x}) \cdot q(\mathbf{z}|\mathbf{x})$.

Here, latent variable $\mathbf{z}$ intuitively corresponds to latent representation extracted from data $\mathbf{x}$.

We can parameterize this latent variable $\mathbf{z}$ by θ and the joint distribution becomes $p_\theta(\mathbf{x}, \mathbf{z}) = p_d(\mathbf{x}) \cdot p_\theta(\mathbf{z}|\mathbf{x})$. Unsupervised learning tasks aim to convert input data to latent representation $\mathbf{z}$ with a set of parameters θ. For this purpose, these models maximize the mutual information between $q(\mathbf{x}, \mathbf{z})$ and $p_\theta(\mathbf{x}, \mathbf{z})$ and they align these distributions.

Notice that for $p_\theta(\mathbf{z}|\mathbf{x})$, true latent variable $\mathbf{z}$ is not completely known compared to supervised annotations. For this purpose, the actual capacity of these tasks is increased when the best objective functions are defined that help to minimize the distance between actual $q(\mathbf{z}|\mathbf{x})$ and parametric distributions $p_\theta(\mathbf{z}|\mathbf{x})$ (i.e., based on data characteristics and problem definitions) . At the end, a parametric hypothesis class is summarized as $f(\mathbf{x}; \theta) : X \rightarrow Z$ in an unsupervised manner.

In Chapter 3, hyperspectral data is selected as the main task for unsupervised learning. For the sack of clarity, hyperspectral data is denoted as multispectral data from now on even if there is dimensional differences. In particular, domain-specific characteristics and problem definitions are perfectly maintained in our methods to obtain $p_\theta(\mathbf{z}|\mathbf{x})$. Furthermore, two problem sets: *spectral unmixing* and *spectral endmember extraction* are determined. The overall aim of these problems is to unmix data for the estimation of constituent materials (i.e., endmember extraction) and their abundance maps (i.e., spectral unmixing) blindly. Note that the proposed methods can be adopted to different tasks as we indicated in the last chapter.

For this purpose, we present a novel method that formulates a unified auto-encoder structure. In this structure, spectral signatures and their abundances are optimized together. Sparsity, non-linearity and uncertainty are encapsulated in the solution by incorporating a class of NN functions. In addition, novel cost functions are proposed to achieve state-of-the-art performance. In particular, standard Euclidean cost function is replaced with spectral angle distance (SAD) to improve the performance. We also adopt the concept of Generative Adversarial Network (GAN) to spectral unmixing for the first time. Ultimately, this step captures the uncertainty exhibited from multispectral data. The rationale behind this contribution is the capacity of GANs to

model random noise by approximating the spectral variability in feature space.

1.2.2 Learning representations from multimodal tasks

In the literature, we observe that learning models cannot be reduced to a single modality. Therefore, we have also studied multimodal characteristics to improve the performance of a representation learning step. In the end, we adapt the previous formulation $f(\mathbf{x};\theta) : X \rightarrow Z$ to multimodal learning.

The notion of multimodal learning is to compute a collection of tasks for the same input data. On the other hand, the same objective function with multiple modalities is optimized for crossmodal learning. Even if these two tasks look different, multiple cost functions must be solved for both models. For additional backgrounds, [5] can be reviewed. For clarity, crossmodal and multimodal data are indicated as multimodal data from now on throughout the thesis.

No matter how independent the tasks are, Stein's paradox claims that there is an accurate solution (on average) when three or more parameter sets are estimated together than separate estimations of individual parameters. The concept of this paradox is demonstrated by the fact that a single observation can have somewhat noisy measurements in the estimation step. Hence, averaging operations for multiple tasks can yield a better estimate by suppressing the errors in the observations. Also, the generalization capacity of representations can be improved by exploiting common knowledge that is exhibited from training samples for related tasks.

This paradox can be accepted as the fundamental motivation behind multi-modal learning. As expected, this ultimately reduces the errors from multiple tasks and performs better due to its generalization capacity. Even if these tasks are independent, there are some common properties such as sensor devices, domain attributes and materials, etc. that positively influence the performance.

In this sense, consider an input sample $\mathbf{x} \in X$ and a collection of labels $\mathbf{y}^t \in Y^t$ for some attributes. Hence, the formulation in the previous section is updated. For multimodal task, a parametric hypothesis class per task is utilized as $f_t(\mathbf{x};\theta) : X \rightarrow Y^t$

where t denotes the indicator of tasks:

$$\mathcal{L}_{mlt} = \sum_{t=1}^{T} \alpha_t \mathcal{L}_t(\mathbf{x}, \mathbf{y}^t; \theta) \tag{1.1}$$

Here, $\mathcal{L}_t(.;.) : Y^t \times Y^t \rightarrow \mathbb{R}^+$ denotes the cost function and α_t values are either uniform or learnable coefficients to scale the contribution of tasks t to the overall cost function $\mathcal{L}_{mlt}$.

On the other hand, we consider a single parametric hypothesis class for crossmodal tasks while multiple input modalities are fed $f(\mathbf{x}^c; \theta) : X^c \rightarrow Y$ where c denotes the indicator of data modality. To this end, the cumulative cost function is revised as:

$$\mathcal{L}_{crss} = \sum_{c=1}^{C} \alpha_c \mathcal{L}(\mathbf{x}^c, \mathbf{y}; \theta) \tag{1.2}$$

α_c values can be unnecessary for this problem, but it helps to rescale the cost functions in case of imbalanced class problem [6]. Furthermore, observe that there is a single set of parameter θ for both learning schemes.

In Chapter 4, the task of multimodal is analyzed under the subjects of semantic segmentation and regression for biomedical and multispectral data. Precisely, CT and MR images are used to validate the proposed model for multimodal data. In particular, these two modalities have diverse visual characteristics, since the working principles are different due to acquisition times and inter-slice distances etc. For this purpose, a novel NN structure and cost function are introduced to improve the performance of multimodal semantic segmentation for biomedical data (i.e., CT and MR data). A distinct feature of our method is that no information about data sources is provided at train and inference phases. Hence, the solution can be realized in a single model for both modalities. This step enhances the usability of these models. Also, the performance is significantly improved. We also conduct additional experiments on a visual classification task that we observe similar results.

In addition, multi-spectral data is used to demonstrate the superiority of multi-modal learning by simultaneously optimizing the relative depth estimate and semantic segmentation problems. By combining multimodal optimization steps for semantic segmentation and relative depth estimation, we show that the convergence rate and per-

formance are improved compared to single-task optimization. The fundamental aspect that differs from the literature is that the actual capacity of multispectral data can be marked for inverse problems when a large amount of data is required for a stable solution. In the proposed method, this requirement significantly reduces by using a multimodal learning structure (i.e., supervised annotations). Also, this modification enhances the quality of solutions.

1.2.3 Restructuring learning models for different domains

As explained, sparsity, non-linearity, uncertainty, and other specific features are essential for unsupervised and multimodal approaches that must be maintained in the solution spaces. In problem formulations, these points must be taken into account by using a class of functions and parameter learning steps. Conventional structures sometimes cannot address all of these concerns and are not suitable for these needs. For instance, the motivation of batch normalization is to eliminate the covariate shift property in diverse data so that small changes between data distributions for similar classes are eliminated at train time. However, this is ineffective for multimodal learning since both distinct and shared patterns must maintain at the same time for practical solutions. With a minor yet effective adaptation, this layer becomes appropriate to this problem, and the state-of-the-art performance can be obtained.

In both Chapter 3 and 4, we present novel contributions to the reorganization of learning structures for the research mentioned above (i.e., unsupervised and multimodal learning). At the end, the characteristics related to data are encapsulated.

1.3 The Outline of the Thesis

Our motivation is to compute robust representations from data for a limited supervision and different modalities. Hence, the forthcoming chapters of this thesis are structured as follows. Please note that the most recent studies which are closely related to the tasks are summarized at the beginning of each chapter:

Chapter 2: Foundations: In this thesis, since neural networks (NN) have led a

series of breakthroughs over traditional machine learning methods, so the designed algorithms generously use NN concepts. Therefore we will present the theoretical and practical foundations of NN concepts from the beginning of *"what is a neuron?"* to the idea behind *generative models*.

Chapter 3: Spectral Unmixing and Endmember Extraction: The spatial resolution of multi-spectral sensors is limited compared to other optics operating in the visible spectrum domain. Hence, pixels are composed of mixture of material spectra. Even if this is a drawback, we can blindly learn essential patterns from data. In unsupervised manner, *spectral unmixing* and *spectral endmember extraction* will be computed from data with novel models. In addition, all theoretical and physical properties of data are discussed.

Chapter 4: MultiModal Learning: We tackle multimodal learning for liver and lung segmentation tasks. In particular, we will show that covariate shift property, which is crucial for a sufficient amount of statistical knowledge from individual modalities, is retained in the solution with a proper formation of NN functions and learning schemes. Furthermore, a multimodal optimization is adopted to address the need for large-scale data for an inverse problem. After all, stable representations are estimated for both domains.

Chapter 5: Discussion and Conclusion: We present the discussions of the outcomes of each chapter. Furthermore, we analyze the experimental results and finish with a short glimpse of possible solutions for the limitations of our models.

Last but not least, the outputs of this thesis are published as five different papers: one journal paper, two conference papers and two arxiv papers (under review), all of which is briefly summarized as follows:

Paper 1: *"Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing." IEEE Transactions on Geoscience and Remote Sensing 57.1 (2018): 482-496.*. This paper presents an end-to-end learning structure based on a neural network autoencoder that is enhanced and restructured by introducing additional layers and a projection metric. In particular, a novel loss function that is composed of a Kullback-Leibler divergence term with SAD similarity is exploited

for parameter optimization.

Paper 2: *"Deep spectral convolution network for hyperspectral unmixing." In 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 3313-3317. IEEE, 2018.* We propose a novel multi-spectral unmixing technique based on deep spectral convolution networks (DSCN) in which fully connected linear operations are replaced by spectral convolutions to extract local spectral features with a spectral normalization layer.

Paper 3: *"Hyperspectral Data to Relative Lidar Depth: An Inverse Problem for Remote Sensing." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 0-0. 2019.* We tackle an inverse problem to compute relative lidar depth from multi-spectral data. To solve the problem limitations, we incorporate semantic knowledge with supervised annotations to reduce the chance of parameter overfitting by exploiting multimodal learning. Moreover, the details of responses are enhanced with Laplacian pyramids and attention mechanisms.

Paper 4: *"Improved Deep Spectral Convolution Network For Hyperspectral Unmixing With Multinomial Mixture Kernel and Endmember Uncertainty." arXiv preprint arXiv:1808.01104 (2018).* This paper presents a novel framework for spectral unmixing that uses an improved deep spectral convolution network (DSCN++) and a multinomial mixture model to estimate abundance maps. Wasserstein Generative Adversarial Network (WGAN) is exploited for the first time to capture the uncertainty in the solution.

Paper 5: *"Cross-Domain Segmentation with Adversarial Loss and Covariate Shift for Biomedical Imaging." arXiv preprint arXiv:2006.04390 (2020).* A novel model is presented that learns robust representations from multimodal biomedical data. Furthermore, it achieves state-of-the-art performance when any of the modalities is supplied as input. Precisely, covariate shift property is retained with structural modification, and adversarial loss is exploited where sparse and rich representations are obtained.

CHAPTER 2

FOUNDATIONS

In this chapter, the theoretical and practical foundations of NN will be explained in detail. These foundations will start from the very primitive terms, and more complex concepts will be described later.

NN concept is simply a loose imitation of neurons in a human brain with a collection of computational units. It typically consists of many individual layers where each layer has a unique goal that helps to solve problems as humans could. However, the initial version of this concept has severe drawbacks, and it induces the NN domain's progress to remain very limited. With the emergence of new components that permit to stack deeper layers, this concept is resurged and leads a series of breakthroughs over traditional methods. In particular, these recent components provide promising results for sparsity, overfitting and vanishing gradient, which are discarded in conventional approaches.

The momentum of NN concepts is accelerated with the accessibility of data and computing devices that perform well for matrix multiplication. First, the history of artificial neural networks will be explained for a gentle introduction. NN components will be discussed later.

2.1 History of Artificial Neural Networks

For humans, it all starts with a neuron, and the complexity of neural structures (actually, it is consciousness) ultimately separates humans from other primitive species. The superiority is that this structure can be imitated in learning-based methods with

brilliant mathematical adaptations.

From the neuroscience perspective, a neuron is specialized to transmit knowledge to other neurons by generating electrical impulses called *Action Potentials*. Although there are different neuron types in our brain, the overall structural goal is to receive, integrate, and transfer signals from/to other neurons. Furthermore, there are three distinct parts for a neuron namely *Dendrites*, *Axons* and *Synapses*. Ultimately, these parts help neurons send and receive electrical signals by performing adaptive excitation operations with action potentials.

As noticed, these components can be used as computational units for learning-based models. Precisely, linear connections and weights between nodes correspond to synapses, and excitation operations are incorporated with a threshold-based activation operation.

Mathematically, let a neural model $f_i(.)$ for i$^{\text{th}}$ neuron over an input sample $\mathbf{x} \in \mathbb{R}^D$ with weights $\mathbf{w} \in \mathbb{R}^D$ and bias $\mathbf{b} \in \mathbb{R}^D$ terms such that the excitation operation $\sigma(.)$ is inserted to complete the mathematical adaptation of human biological brains as follows:

$$f_i(\mathbf{x}) = \sigma(\sum_{d=1}^{D} w_d \cdot x_d + b_d) \tag{2.1}$$

In history, the first artificial neural network inspired by human biological brains goes way back to the 1940s. It is only a computational model called *threshold logic* with electrical circuits [7]. Later, *hebbian learning* emphasizes that neurons can be associated if they are repeatedly active at the same time for similar data [8]. In 1958, the *perceptron* algorithm is invented that is the first application of binary classification [9]. Until the 1980s, the development of NN goes back and forth because they are hand-engineered models. The term of *backpropagation* steeply triggers the NN domain, and multi-layer networks can be trained automatically by updating learnable parameters on each layer [10].

As an extension, neuroscientists emphasize that the analogy of visual hierarchies called *deep hierarchies* is more useful to interpret data as humans would [11]. Hence, multi-layer networks (i.e., a hierarchical representation) can be suitable for learning-based models.

Even if novel components such as max-pooling, convolution etc. are introduced in this period, NN models strive to extract robust representations from data. The main reasons are the lack of computing resources and the weakness of layers for different cases (i.e., one of which is that parameters stuck to local minima).

NN concepts become active again in the early 2010s. An initial application is for recognizing higher-level concepts from unlabeled data on distributed computers [12]. The superiority of this method is based on the fact that it learns a large number of learnable parameters, as in the human brains, by stacking deeper hidden layers. Later, these observations guide the other methods to achieve state-of-the-art performance and surpass human prediction performance for classification tasks [13, 14, 15]. Deep neural networks inevitably power many aspects of modern society by influencing the commercial products that we use in our daily lives.

The aim of this chapter, as stated, is to cover all the essential terms in deep neural networks. Therefore, we divide this chapter into two as *Foundations of Deep Models* and *Deep Network Research*. In *Foundations of Deep Models* part, we will answer the questions of *what are the main components of modern neural networks?*, *how can learnable parameters be optimized?* and *what are the mathematical foundations for NN layers?*.

In *Deep Network Research*, we will describe the most recent terminologies that are slightly higher-level than individual layers such as *autoencoders*, *adversarial models* and their variants. Note that even if there are few other terms in the literature that we can expand further, we will mainly focus on the terms that are used extensively in this thesis.

2.2 Foundations of Deep Models

Sometimes the terminology of deep learning has been seen as a magical box that is not fully understood the underlying structures of layers are. Notably, the studies entitled *Explainable AI* [16] aim to compute reasonable explanations and visualizations about the latent representations to understand this terminology better. Nevertheless, it still outperforms when a sufficient amount of computation and data is fed to the models,

so that it is persistently applied to all computer vision and machine learning problems.

However, this learning model can be easily simplified to two main steps as *feed-forward* and *backward*. Briefly, a deep model commits a solution for an input in the forward step, and an error/reward occurs between predicted and desired outputs for each iteration. On the other hand, this estimated error/reward is propagated via layers, and learnable parameters are optimized accordingly in the backward step. These forward and backward steps are applied repeatedly until all parameters have reached equilibrium.

For this purpose, we will first explain more abstract concepts such as cost functions, gradient-based learning and learning models etc. Later, famous neural network layers used in state-of-the-art models will be described: dense, convolution, activation function, normalization etc.

2.2.1 Learning-Models

Generally speaking, a deep model aims to approximate some function $f^*(.)$ where it takes an input sample $\mathbf{x} \in \mathbb{R}^D$ and makes a prediction $\hat{\mathbf{y}} \in \mathbb{R}^K$ by computing a latent representation $\hat{\mathbf{z}} \in \mathbb{R}^M$.

Even if the desired outputs of a function can be calculated manually (it can take decades for complex problems), a learnable deep model can be trained with a new set of functions from data $f(., \theta)$ parameterized by θ (i.e., deep hierarchies). Indeed, the estimation of latent representations is a critical factor for high performance since the accurate projection of input data with problem definitions is essential.

If you have instructions on how to project data or a sufficient number of instances that truly approximates a function. In this case, learnable parameters can be trained by incorporating a sequence of linear/nonlinear transformations. This terminology is renowned as *supervised learning*. After all, a set of labeled data $\mathbf{y} \in \mathbb{R}^K$ is used to optimize learnable parameters θ. Note that there is no absolute confidence about latent space, and there are infinitely many solutions for single-tasks based on models, initialization, and problem definitions.

For *unsupervised learning*, if no instruction about data is available, then models automatically find common patterns and associations exhibited from data. As expected, these two terminologies have their advantages and disadvantages. In the following chapters, we will discuss them in detail.

2.2.2 Cost-Functions

Trainable parameters must be optimized with proper cost function(s), which vary for problem definitions and performance criteria.

The cost function concept is to map the estimated values of models to the actual desired space by linking them with an error value. This function quantifies how bad a model is for each iteration. Depending on the problem definitions, a large class of cost functions can be utilized. Furthermore, an error or a reward is either minimized or maximized that is calculated from some cost functions. The notation for the optimization of a parameter set θ with some cost function $\mathcal{L}$ is denoted as follows:

$$\min_{\theta} \mathcal{L} \tag{2.2}$$

In the scope of this thesis, two sets of cost functions are mainly utilized (except our contributions). *Mean Squared Error* (also known as *Euclidean Distance*) is a regression metric that takes the average of the squared difference between prediction $\hat{\mathbf{y}} \in \mathbb{R}^K$ and desired output $\mathbf{y} \in \mathbb{R}^K$ as follows:

$$\mathcal{L}_{mse} = \frac{1}{K} \sum_{k=1}^{K} (y_k - \hat{y}_k)^2 \tag{2.3}$$

Note that this class of cost functions generously penalizes the error values that are far from optimal annotations and adds small costs for the amounts that are close.

Furthermore, this cost function is reformulated from the probabilistic perceptive as follows:

$$\mathcal{L}_{kl} = D_{kl}(P||Q) = \sum_{\mathbf{x} \in X} P(x) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})} \tag{2.4}$$

Here P and Q indicate two probabilistic distributions and this cost function measures how one probability distribution diverges from other.

Negative Log-likelihood is a metric that calculates the entropy between two probability distributions. These two distributions explicitly correspond to the label distribution and prediction distribution:

$$\mathcal{L}_{ce} = H(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{K} \sum_{m=1}^{K} y_k \cdot \log \hat{y}_k \tag{2.5}$$

Notice that there is a negative sign at the front because the outputs are maximized by reducing the distance between two distributions.

In particular, softmax negative log-likelihood has a specific property that it constantly penalizes misclassified predictions. Unlike other cost functions as hinge-loss [17] etc, this cost is never zero. Hence, this induces a problem, and parameter overfitting is inevitable for this cost function.

Another interesting fact is that if we minimize KL-divergence, it is equivalent to the maximization of log-likelihood.

2.2.3 Gradient-Based Learning

Gradient vectors for parameters (i.e., the derivative of an error function in terms of parameters) are computed in the same way as any other machine learning model to optimize learnable parameters. Moreover, the magnitude and direction of the gradients vaguely indicate how the errors change as parameters are increased or decreased with small amounts. Gradient-based optimizers drive the cost functions to the opposite direction of gradients, where this implicitly corresponds to lower values in cost functions.

If we assume that all layers are differentiable, then errors/rewards at the output of a model can be effectively back-propagated to each layer by using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_l} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_L} \cdot \frac{\partial \mathbf{z}_L}{\partial \mathbf{z}_{L-1}} \cdots \frac{\partial \mathbf{z}_l}{\partial \mathbf{W}_l} \tag{2.6}$$

Here, $\mathbf{W}_l$ denotes the learnable parameter at layer l. Moreover, average gradient calculation from few samples offers several theoretical and practical benefits for parameter optimization instead of one or all samples. For instance, a gradient from one

sample can lead to the wrong direction due to noisy measurements. On the other hand, gradients estimated from all samples help to converge a better solution, but the computation requirement makes the solution impractical. For these reasons, a small subset of samples is randomly selected for each iteration, so that undesired gradients and high computational requirements are eliminated. This procedure is entitled as *Stochastic Gradient Descent*. Parameters surprisingly and quickly reach a minimum error in comparison to sophisticated optimization techniques [18, 19].

For each iteration, these gradients are used to update the parameters as follows:

$$\mathbf{W}^{t+1} = \mathbf{W}^{t} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{t}} \tag{2.7}$$

Here, η denotes the learning rate, and t index is for the iteration step. However, this update scheme (i.e., called as optimizer) gets easily trapped to local minima. As a solution, the momentum term accumulates gradient vectors from the past iteration steps and guides the parameter updates [20]. The influence of past gradients is still retained in the solutions, and oscillations of individual updates are eliminated by gradient smoothing. Notice that the momentum optimizer accelerates parameter search in the direction of minima.

$$\mathbf{V}^{t+1} = \eta \mathbf{V}^{t} - \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{t}} \tag{2.8}$$

$$\mathbf{W}^{t+1} = \mathbf{W}^{t} + \mathbf{V}^{t+1} \tag{2.9}$$

α is the coefficient of momentum, and it is usually accepted as 0.9. Other variants that share similar notions are also presented in the literature by mimicking second-order derivations [21].

2.2.4 Dense Layers

Dense layer is a simple linear operation that maps an input sample $\mathbf{x} \in \mathbb{R}^{D}$ to a target sample $\mathbf{y} \in \mathbb{R}^{K}$ by a learnable transformation $\mathbf{W} \in \mathbb{R}^{D \times K}$ (i.e., a matrix multiplication) as follows:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{2.10}$$

Here, $\mathbf{b} \in \mathbb{R}^K$ is also a learnable parameter called bias. After all, each sample is transformed into a global representation, as this layer connects all elements of input data and learnable parameters. Furthermore, this layer has both advantages and disadvantages. For instance, if we want to find overall interactions in data and embed locality to the representations, this is a suitable solution. In particular, classification tasks are recognized as the most common application area for this layer.

However, this is not practical to extract robust representations from local regions. In particular, a small data translation leads to entirely different outcomes. Furthermore, parameter size is ($\approx D \times K$) considerably large and somehow redundant. This drawback inevitably leads to parameter overfitting. For these reasons, the realization of deep networks with dense layers is almost impossible.

In parameter updates, its derivative is directly equal to input data because it is a linear operation:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{W}} = \mathbf{x} \tag{2.11}$$

2.2.5 Convolution Layers

Convolution layer is also a linear operation similar to dense layers. The main difference is that, as understood, a convolution operation is utilized. Moreover, representations are computed locally by parameter sharing:

$$y_i = \sum_{m=0}^{M} w_m \cdot x_{m+i} + b_m \tag{2.12}$$

Here, $\mathbf{w} \in \mathbb{R}^M$ is renowned as kernel and the dimension of target data $\mathbf{y}$ depends on both input data and kernel sizes (i.e., $\approx K = N - M + 1$).

In contrast to dense layers, repeated patches on data are precisely captured by convolution layers since shared parameters are used. For instance, if you want to detect human faces from input data, convolution operations probably yield high activation responses for all face positions. As noticed, this is not valid for the dense layer since locality is inserted to the representations. Indeed, this feature enhances the solution in which sparse connections are made. Furthermore, the total number of operations

is significantly reduced compared to dense layers. This step enables the models to exploit deeper layers as in the human perception system [15].

Furthermore, this layer is shift-invariant, and the output does not change even if the data is shifted. The derivation of convolution operation is formulated as:

$$\frac{\partial y}{\partial w_m} = \sum_{k=0}^{K} x_{k+m} \tag{2.13}$$

Note that both dense and convolution linear operations are not invariant to scaling and rotation changes. Additional steps must be used to have the robustness to these cases.

2.2.6 Activation Layers

Both convolution and dense layers are a linear operation that maps input data to a different space. In this form, output data does not convey too much knowledge due to the lack of sparsity and non-linearity. Hence, it must be transformed into rich representations with additional layers. As mentioned, the excitation operation in human brains intuitively determines whether input data is activated or not.

In the same manner, an activation layer $\sigma(.)$ works as a gate function. It transfers the input sample to the next layers by adding non-linearity characteristics. Note that this is an element-wise operation.

In the literature, several nonlinear activation functions are widely exploited in modern NN models. These functions must be used for different purposes and problem definitions.

Sigmoid function normalizes input data to 0 and 1 value bounds. Hence, smooth gradient vectors and activation responses are obtained at the output of this layer. However, these values can be easily saturated to its limits, and the error propagated from the next layers can be vanished. The response of this activation function and its derivative can be formulated as:

$$\mathbf{y} = \sigma(\mathbf{x}) = \frac{1}{1 + \exp^{-\mathbf{x}}} \tag{2.14}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \sigma(\mathbf{x}) \cdot (1 - \sigma(\mathbf{x})) \tag{2.15}$$

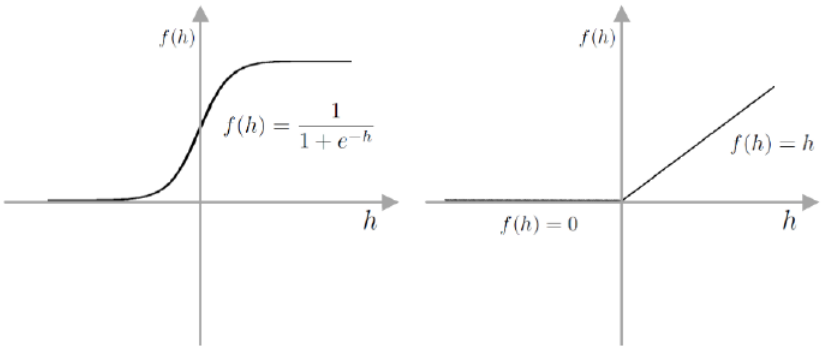

Figure 2.1: The response of Sigmoid and ReLU functions.

Rectified Linear Unit function is also a nonlinear activation that prunes the input values, which are less than 0. In particular, a quick convergence is expected for this function. However, the pruning operation can lead to the death of activation responses. This issue is solved by adding a learnable term a that determines the slope of negative activation responses [22, 23]. Similarly, the response of this activation function and its derivative can be formulated as follows:

$$\mathbf{y} = \max(\mathbf{x}, a\mathbf{x}) \tag{2.16}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{cases} 1, & \text{if } \mathbf{x} \geq 0 \\ a, & \text{if } \mathbf{x} < 0 \end{cases} \tag{2.17}$$

Fig. 2.1 illustrates the response characteristics of the Sigmoid and Rectified Linear Unit (ReLU) activation functions for various input values. As seen, Sigmoid function bounds the responses between 0 and 1 while there is no limit for the ReLU function.

Similarly, *Softmax* function normalizes the input data with the sum of all elements. This function is used in particular to transform inputs into a probabilistic space especially for classification tasks:

$$y_i = \frac{\exp^{x_i}}{\sum_{m=0}^{M} \exp^{x_m}}, \text{ where } i = 1..M \tag{2.18}$$

As noted, sigmoid and softmax activations are generally used for probabilistic normalization. However, a rectified linear unit is used for rich representations because sparse responses are obtained.

2.2.7 Regularization Layers

In simple terms, the purpose of regularization is to add a penalty term to the cost function that modifies the parameters so that parameter overfitting is resolved partially. The need for this term is multiplied when the total number of parameters is increased. Undoubtedly, the effect of the regularization term is not always reflected in the cost function for all hidden layers. For this purpose, Dropout layer is proposed [24, 25]. This layer adds randomization to each layer's parameters by dropping a subset of input samples in the training step. Hence, this layer treats the same data differently for each iteration and weakens the connections between layers that are established incorrectly for training data.

Mathematically, input data $f(\mathbf{x};\theta)$ is randomly sampled with a probabilistic distribution as $r \sim \text{Bernoulli}(p)$ and the output is equal to $\mathbf{z} = r \bigodot f(\mathbf{x};\theta)$. Here, $\bigodot$ indices an element-wise product and $0 < p \leq 1$ should be defined as a hyper-parameter.

However, this layer can lead to undesirable solutions if data is noisy. Therefore, this must be configured based on the problem definitions and property of data.

2.2.8 Normalization Layers

As long as you have a steep and unbound feature space, proper convolution and activation layers are not guaranteed for stable and deeper neural networks. A question from the readers may arise that some optimizers can already improve the parameter optimization step to a certain extent, as explained (i.e., momentum). However, this feature is not resolved, especially when the parameter size is large. Furthermore, the data distributions for the same objects can alter due to diverse visual characteristics. Hence, straight-forward models use many learnable parameters to capture all of these variations and fill the semantic gap. However, this is not practical.

For this purpose, normalization layer is presented to overcome these two issues. The idea is to normalize input data by using its mean and variance values:

$$y = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{2.19}$$

Here, γ and β are scale and shift terms that are learnable. In addition, the mean and variance characteristics of input data might differ based on normalization models. For instance, batch normalization, as understood, normalizes data with batch statistics [26]. However, instance normalization uses spatial statistics instead [27].

Note that the normalization of data with batch statistics ultimately introduces noise in the learning step since different batch features are used for each iteration. Hence, batch normalization layer also regularizes input data as dropout layer.

2.2.9 Pooling Layers

Besides the fact that convolution operations are shift-invariant, this feature is not applicable when convolution layers and dense layers are used together. In other words, the locality of latent representations adversely affects the performance.

For this purpose, pooling layers are introduced, and latent representations become invariant to small translations. In action, this layer downsamples the representation by using a patch window called the pooling window. Average and maximum operations are two operations that are typically utilized for downsampling.

Furthermore, this layer helps to summarize the representations for the next layers. One of these operations (i.e., average and maximum operations) is selected based on problem definitions. For instance, if we want to compute a sparse representation, max-pooling is more suitable. If, on the other hand, we combine the outputs of multiple models, average-pooling yields more stable results.

In parameter updates, the errors are only passed back from the active elements in the forward step.

2.3 Deep Network Research

Even if these explanations give an insight into the capacity, objectives, and drawbacks of individual neural network components, the combination of these layers makes the actual contribution to solve problems. Hence, advanced terms must be defined to

understand the most striking achievements in this area thoroughly.

As mentioned, these advances cover a wide variety of studies, from computer vision to remote sensing fields. Most notable research domains, like autoencoders, generative models, recurrent neural networks and reinforcement learning, lead the efforts for deep learning. As part of this thesis, we will focus on a small set of these terms related to our problems. For this purpose, we will describe concepts related to deep NN models such as autoencoder, generative model and skip-connections in the following sections.

Before these terms, we need to discuss an essential question that deep network research is focused on: *What makes the representations better?* There is no universal answer to this question in the literature. Based on problem definitions, this answer might change drastically. However, some specific properties affect the representation capacity as well.

Sparsity: Intelligent methods strive to obtain robust representations, for example, to distinguish cars from humans. Instead of keeping the same properties for different objects regularly, some of them must alter in latent space.

Non-linearity: For simplicity, linear models are usually used to depict the input data. However, the change in predictions is not proportional to input data for most cases. Hence, non-linearity must be preserved in the solutions.

Generalization: Ideal solution must also work for unseen data. Therefore, the models should obtain a generic solution for all data.

Deep hierarchies: As explained, the working principle of the human visual system is based on this assumption (i.e., deep hierarchies). Hence, representations must be in the form of hierarchies to learn latent patterns step-by-step.

2.3.1 AutoEncoder Model

Autoencoders aim to reconstruct input data by highlighting its latent patterns. In practice, this model consists of two parts as encoder and decoder. Encoder part takes an input sample $\mathbf{x}$ and transforms the data onto a latent space Z with a set of functions

as $\mathbf{z} = h(\mathbf{x}, \theta)$ where $\mathbf{z} \in Z$. As mentioned, there is no supervised knowledge available about latent representations. Hence, infinitely many solutions exist. On the other hand, decoder part reconstructs the original data by using latent representations as $\hat{\mathbf{x}} = g(\mathbf{z}, \theta)$. In the end, useful characteristics of data are emphasized in latent representations by resembling the training data as $\hat{\mathbf{x}} = g(h(\mathbf{x}, \theta), \theta)$.

In fact, this transformation can be useless unless some critical constraints are imposed. For instance, if the dimensionality of latent space is equal to input data, latent space captures almost the same knowledge (i.e., identity transformation) as in the original input space. The capacity of representations is inevitably deteriorated. For this purpose, the dimensionality of latent space is deliberately selected to be smaller than the input space to capture high-level patterns from data. Cost function $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}; \theta)$ must be optimized with learnable parameters θ by minimizing the dissimilarity between real and predicted samples.

Even if distinct representations can be computed with this base structure, sparse representations are still essential. This statement can be easily checked with a simple induction. Ultimately, the goal of autoencoders is to extract robust representations that are as powerful as the representations in supervised learning. Furthermore, studies show that the representations in classification tasks are usually sparse [15]. Hence, the representations of autoencoders must also have this property.

For this purpose, additional criteria (i.e., regularization terms), in addition to overall cost function can be exploited, which enforces sparsity to learnable parameters. Note that these criteria can also be imposed with NN layers where it is one of our contributions presented in the thesis:

$$\mathcal{L}_{sps} = \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}; \theta) + \Omega(\theta) + \Omega(\hat{\mathbf{x}}) + \Omega(\mathbf{z}) \tag{2.20}$$

Here, $\Omega(.)$ functions denote the regularization criteria that penalize learnable parameters, output or latent representation with the cost function $\mathcal{L}$. In fact, this might not be required to use all of these terms at the same time. Their contributions must be decided based on problem definitions.

2.3.2 Adversarial Model

Adversarial models utilize an additional NN model (i.e., called discriminative model) that classifies whether an input sample comes from real distribution or fake distribution. On the other hand, a NN model (i.e., called generative model) competes with this model to align the fake distribution with the real one.

Mathematically, a generative model $g(\mathbf{z}; \theta^g)$ makes a proposal $\hat{\mathbf{x}}$ from a latent representation $\mathbf{z}$ with trainable parameters θ^g. Similarly, a discriminative model optimizes a set of parameters θ^d to distinguish the real data $d(\mathbf{x}; \theta^d)$ and predicted one $d(g(\mathbf{z}; \theta^g); \theta_d)$ accurately. Later, the generative model updates its parameters θ^g to fool the discriminative model. This endless game between two individual models continues until all parameters have reached equilibrium. For this purpose, two sets of cost functions are defined as follows:

$$\mathcal{L}_{disc} = E[d(\mathbf{x}; \theta^d)] + E[1 - d(g(\mathbf{z}; \theta^g); \theta^d)] \tag{2.21}$$

$$\mathcal{L}_{gen} = E[d(g(\mathbf{z}; \theta^g); \theta^d)] \tag{2.22}$$

As stated, the discriminative model pushes the parameters far from generated samples. On the contrary, the generative model optimizes the parameters to produce samples similar to the real ones. Note that $E[.]$ indicates the expectation value of input data.

From the probabilistic perspective, the generative model defines a probability distribution p_g obtained from $\mathbf{z} \sim p_z$. Similarly, real data has a probability distribution p_r and by solving these cost functions, the model reaches an optimum solution when $p_g = p_r$. In this configuration, the optimum value for discriminative model $\mathcal{L}^*_{disc}$ is equal to $\frac{p_r}{p_r + p_g}$. When $p_g = p_r$, this value is set directly to 0.5.

Furthermore, Eq. 2.21 and 2.22 can be simplified with a variant of KL-divergence terms (i.e., Jensen–Shannon divergence) for an optimum discriminative solution as:

$$\mathcal{L}^*_{disc} + \mathcal{L}_{gen} = 2D_{JS}(p_r || p_g) - 2\log 2 \tag{2.23}$$

The Eq. 2.23 is mathematically valid for generative models, and it shows that some

critical drawbacks of KL-divergences remain in the solution. Hence, this structure suffers from significant drawbacks such as convergence, mode collapse and vanishing gradient [3, 4, 28].

For this purpose, other variants are proposed, such as Wasserstein model, to alleviate these problems [3, 4]. Note that these issues and the advantages of the Wasserstein model will be discussed in the following chapter.

2.3.3 Skip-Connections

From 2012 to the present day, deep learning models have dominated the main challenges in image classification. These methods have achieved state-of-the-art performance compared to hand-crafted models [15]. As mentioned, the virtue of these models is the advent of computing units and the introduction of useful layers. Hence, the concept of deep learning (i.e., deep stack kernels) improves to find rich and hierarchical representations. At some point, the deep layers are stuck into a particular number (i.e., at most, 16-19 layers can be used) because the output errors cannot be efficiently propagated back to the initial layers.

For this purpose, skip-connection is utilized in which the ability of error propagation for deep networks is strictly maintained. The concept introduces a unit connection of input data to the output layers as:

$$\mathbf{z}_{t+1} = \mathbf{z}_t + f(\mathbf{z}_t; \theta) \tag{2.24}$$

Indeed, this structure provides several advantages compared to conventional techniques. One of which is that this allows direct knowledge flow from initial layers to deeper layers. Hence, this feature enables the model to learn incremental representations to extend the representation capacity. Moreover, the errors at the output can be efficiently propagated since the derivation of the error function is transferred identically as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}_t} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_{t+1}}(1 + \frac{\partial f(\mathbf{z}_t; \theta)}{\partial \mathbf{z}_t}) \tag{2.25}$$

In addition, this structure is suitable for autoencoder models where skip-connections are utilized from encoder to decoder parts [29].

2.3.4 Attention Model

Attention models aim to focus on a subset of representations by using additional subnetworks. This concept enhances the sparsity of responses considerably.

Formally, a latent representation $\mathbf{z} \in \mathbb{R}^M$ is fed to a NN model $\varphi(., \theta)$ that computes element-wise scores $\mathbf{a} \in \mathbb{R}^M$ with a set of learnable parameters θ as follows:

$$\mathbf{a} = \varphi(\mathbf{z}, \theta) \tag{2.26}$$

$$\mathbf{z} = \mathbf{z} \bigodot \mathbf{a} \tag{2.27}$$

Here, $\bigodot$ is the element-wise product and the range of scores changes from 0 to 1. As noticed, 0 means no contribution is desired for the next layers while 1 indicates that the responses are fully retained.

In addition, the structure of attention model aims to extract shared knowledge from input data at the beginning of the model. Hence, input data is compressed by averaging operations either on spatial or channel responses. Spatial mean extracts knowledge from neighboring pixels by encoding where the model should attend. On the other hand, channel mean emphasize on what the model should attend. Depending on the problem definitions, spatial and/or channel attention can be utilized.

CHAPTER 3

SPECTRAL UNMIXING AND ENDMEMBER EXTRACTION

In this chapter, we aim to find a proper answer to the question that we ask in the beginning of this thesis: *How can we compute robust representations from data without supervision but make full use of domain-specific features?* Moreover, multispectral data is selected as an application domain due to its comparable results. Note that this same pipeline can be applied to biomedical data that we will explain in the conclusion chapter.

Unlike the output of visible-range sensors, data acquired from multi-channel sensors is a precious asset to interpret the environment for a variety of detection, identification, and monitoring applications [30, 31, 32]. Ultimately, these sensors give knowledge beyond the visible spectrum of humans. However, low spatial resolution is a severe limitation for these sensors and pure material signatures of a scene can be mixed in different fractions due to their spatial interactions.

After all, pixels consist of a mixture of material spectrum signatures. Constituents (endmembers) and their fractions (abundance maps) should be determined from data, and two techniques play a dominant role: *Endmember Extraction* and *Spectral Unmixing*. In endmember extraction, the aim is to distinguish pure spectral signatures (i.e., constituent materials) from data by exploiting common distribution statistics of signatures. Similarly, spectral unmixing is used to determine the fractions (i.e., abundances) that correspond to the percentage of materials contributed to the mixture of signatures.

The complexity of this problem is that pure materials and their fractions have to be estimated blindly [33, 34, 35, 36]. Since these two unknown parameters affect each

other, both parameter sets should be optimized together. Therefore, these cases make the problem one of the most challenging fields for unsupervised learning.

With a simplification, the mixture of materials can be formulated with a linear model [37]. Let a multispectral data point $\mathbf{x} \in \mathbb{R}^D$ such that it is the linear combination of endmember signatures $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_K] \in \mathbb{R}^{D \times K}$ with fractions $\mathbf{y} = [y_1, y_2, ..., y_K] \in \mathbb{R}^K$ as:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{e}_k y_k + \eta = \mathbf{E}\mathbf{y} + \eta, \text{ s.t. } y_k \geq 0, \sum_{k=1}^{K} y_k = 1 \tag{3.1}$$

Here, K denotes the number of endmembers. Moreover, D is the spectral bands of multispectral data. In addition, this formulation has an additional term η (assumed that it is a zero-mean Gaussian noise $\mathcal{N}(0, 1)$ to simulate possible noise sources such as sensor readout noise or illumination variability.

Even if this formulation provides sufficient knowledge to multispectral data models, some essential features are expected to be discarded. In particular, the atmospheric conditions (i.e., nonlinear interactions) and the variability of the intrinsic material signatures for the same classes lead to the underestimation of the solutions. Notice that we can preserve nonlinear characteristics by using learning-based NN models.

However, the variability of intrinsic material signatures (i.e., renowned as *Spectral Variability*) cannot be captured with this classic formulation. To strength the formulation [38], it must be redefined by addressing spectral uncertainty in the formulation:

$$\mathbf{x} = \sum_{k=1}^{K} (\mathbf{e}_k + \gamma_k) y_k + \eta = \mathbf{E}\mathbf{y} + \eta, \text{ s.t. } y_k \geq 0, \sum_{k=1}^{K} y_k = 1 \tag{3.2}$$

Here, the only difference is that it uses γ_k term, which adds a relaxation term for k^{th} material. This term imposes that even if a limited number of pure materials is used, there is an uncertainty that permits the signatures to alter for different samples under diverse illumination conditions.

In the scope of this thesis, we will present two novel and state-of-the-art methods that broadly use these two equations.

The first method is based on a novel two-staged neural network auto-encoder with

an end-to-end learning scheme that specializes to compute endmember signatures and fractional abundances from multispectral data. Note that this model is the first successful attempt that relies on a sparse auto-encoder network. It also outperforms conventional methods all in an unsupervised manner.

This architecture retains sparsity and non-linearity exhibited from data with a rectified activation function and a normalization layer. This adaptation allows us to set the bias terms to zero in the formulation, and the adverse effects of bias terms are eliminated in the solution. However, complete sparsity is not still provided and leads to overfitting because the networks tend to generate active latent representations for highly correlated materials regularly. Hence, hard-response selection and regularization techniques are also adopted. Furthermore, we replace the inner-matrix product in encoder part with spectral angle distance (SAD). Third, we introduce a set of penalty terms for objective function. They enforce closer angular similarity between the original and reconstructed data in addition to standard Euclidean (l2) reconstruction cost function. Lastly, endmember signatures and their fractional abundances are optimized together with a stochastic gradient-based solver. Hence, this offers advantages to handle large-scale data.

The second algorithm focuses on only spectral unmixing problem. First, dense linear operations are replaced with convolution layers to reduce the effects of high-dimensionality data. This modification enables the model to extract robust local representations. Also, it promotes a deeper architecture (i.e., deep hierarchies) with a stack of convolutions. Furthermore, a novel normalization layer (i.e., spectral normalization) is proposed, which improves the spectral selectivity of layers. More importantly, we introduce a multinomial mixture kernel based on a NN structure that estimates the abundance maps per-pixel by mimicking the Gaussian distributions mixture. In particular, the learnable parameters used for this kernel are optimized for the first time with Wasserstein GAN. This step leads to precise and stable solutions compared to other optimization techniques.

First, we will review the literature for spectral unmixing and endmember extraction. Later, the details of our methods are described in two different sections, as we have summarized. Experimental results obtained on several real and synthetic datasets will

be explained at the end.

3.1 Literature Review

Even if spectral unmixing and endmember extraction problems are nested and must be solved within a single model, the literature addresses these problems separately. Hence, this section is divided into two parts.

3.1.1 Endmember Extraction

As mentioned, the first methods for endmember extraction approach this problem as the data is modeled with linear models. Hence the studies generally concentrate on this area.

For this purpose, geometric models are widely used to determine the endmember signatures from multispectral data. Basically, data distributions are modeled as a simplex set. This idea intuitively embrace the fact that the vertices of a simplex set correspond to pure material signatures since all data points can be spanned with these sparse points.

One group of geometric models exploits the presence of pure samples in multispectral data. *Pixel Purity Index* (PPI), which is one of the first approaches under this assumption, repeatedly projects multispectral samples to a set of random vectors [39]. *N-FINDR* method randomly selects pixels, and a new signature is replaced with these pixels in case the volume of a simplex set is increased [40]. *Vertex Component Analysis* (VCA)-like method extends this assumption by imposing an orthogonality to endmember signatures [41]. Lastly, *Simplex Growing Algorithm* (SGA) similarly grows the simplex volume by leveraging the determinant of affinity of spectral signatures step-by-step [42]. However, these methods have severe drawbacks due to the mixture of materials at macroscopic and microscopic levels. Hence, these disadvantages can lead to inaccurate spectral estimates. In addition, the need for pure materials in real scenarios cannot hold for many data samples (i.e., particular for real data). However, due to their computational superiority, these methods can be recognized as the most

effective algorithms in linear models.

Another group for geometric models is to minimize the volume by adding a margin with additional constraints or kernel structures. *Minimum Volume Constrained-Nonnegative Matrix Factorization* (MVC-NMF) introduces an additional regularization term called a simplex volume term. It factorizes a matrix of samples into two non-negative matrices [43]. Hence, these matrices correspond to endmember signature and abundance maps, respectively. *Iterative Constrained Endmembers* (ICE) simplifies the solution compared to MVC-NMF by adapting a convex quadratic volume regularizer [44]. Furthermore, *Sparsity-Promoting ICE* (SPICE) extends ICE by incorporating sparsity-promoting priors [45]. However, the presence of outliers and noise can dramatically decrease performance since sub-optimal solutions are estimated. Hence, *Minimum volume simplex analysis* (MVSA) replaces the non-negative constraint with a hinge-type constraint that permits limited outliers in the solution [46]. Even if this assumption addresses some of the drawbacks, the complexity of these procedures (i.e., non-convex optimization) makes the approaches difficult to apply for real data.

In recent years, NN models dominate the endmember extraction since these models retain essential features in multispectral data. These features can be recognized as sparsity and nonlinearity. Hence, more robust representations can be computed compared to hand-crafted techniques and observations. As indicated, our proposed method is the first successful attempt based on a sparse auto-encoder network that achieves state-of-the-art performance. By using our method, deeper neural network layers are adopted [47, 48]. A stacked and nonnegative autoencoder is used to detect outliers with a noise-signal ratio [49].

3.1.2 Spectral Unmixing

Likewise, the first approaches are based on linear models due to their simplicity. In particular, sum-to-one and non-negative constraints are conditioned for least-square solutions [50]. The complexity of these models is reduced by setting all negative values to zero. Later, the rest is normalized so that the sum of all becomes one.

Even if the linear mixture models are simple and offer practical advantages, it is not sufficient to solve complex problems. Multiple scattering effects, microscopic-level material mixtures and water-absorbed environments can be recognized as the severe natural cases that cannot be handled by linear models. Hence, bilinear models are generally used to handle nonlinear scenarios in which bilinear interactions of linear frequencies are considered in the model. *Nascimento Bilinear Model* computes second-order interactions of two endmember signatures [51]. By sharing this notion, *Fan Bilinear Model* and *Generalized Bilinear Model* insert additional terms that scale the contribution of interactions [52, 53]. Even if the nonlinearity is retained in these models, one of the most important properties (i.e., spectral uncertainty) is still discarded.

As a solution, distribution-based models describe endmember signatures as they are sampled from some distributions. Hence, the effectiveness of spectral uncertainty is introduced to the models. In the literature, various types of distributions are exploited as *Normal Composition Model* (NCM) [54] and *Beta Composition Model* (BCM) [55]. Notice that these models approximate each fractional abundance to a unimodal distribution. However, this undermines the representation capacity since spectral variability is discarded. Furthermore, *Gaussian Mixture Model* (GMM) is adopted for this problem and implements a suitable solution, especially for real data [38]. However, the complexity of this method limits its applicability for large-scale data.

Recently, neural networks are used by additional cost and NN functions. In particular, Wasserstein GAN is integrated into our method to retain spectral uncertainty as in distribution-based models. The latter models adopt this idea with minor updates in their solutions [56].

3.2 Sparse Auto-Encoder Network for Endmember Extraction

As explained, NN autoencoders are composed of two successive parts to compute representations from input data. Encoder part converts input data into latent representations. Later, decoder part resembles the original data from these representations. After all, potential correlations in data are uncovered in nonlinear transformations.

Ultimately, these characteristics are captured by learnable parameters.

For this purpose, the motivation of our method is to transform the data into latent representations so that specific characteristics of data are retained as constituent material signatures and their fractions.

Mathematically, for a given sample $\mathbf{x} \in \mathbb{R}^D$, autoencoder transforms the data to a latent representation $\mathbf{y} \in \mathbb{R}^K$ with the inner matrix product of a learnable parameter set $\mathbf{W}^e \in \mathbb{R}^{D \times K}$ and $\mathbf{b}^e \in \mathbb{R}^K$. Then, the reconstruction of input sample $\hat{\mathbf{x}} \in \mathbb{R}^D$ is computed as follows:

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}^e \mathbf{x} + \mathbf{b}^e) \tag{3.3}$$

$$\hat{\mathbf{x}} = \sigma(\mathbf{W}^d \hat{\mathbf{y}} + \mathbf{b}^d) \tag{3.4}$$

Here, $\sigma(.)$ indicates the element-wise nonlinear activation function, and a logistic activation function is applied for nonlinearity. Similarly, the inner matrix product of a learnable parameter set $\mathbf{W}^d \in \mathbb{R}^{K \times D}$ and $\mathbf{b}^e \in \mathbb{R}^D$ is used to transform the latent representation back to the original input space. Furthermore, latent representations $\hat{\mathbf{y}}$ correspond to material abundance maps. As noticed, two individual parameter sets for encoder (i.e., $\mathbf{W}^e$, $\mathbf{b}^e$) and decoder (i.e., $\mathbf{W}^d$, $\mathbf{b}^d$) parts are utilized. Parameters are generally optimized by minimizing standard Euclidean (MSE) reconstruction error between the original and reconstructed samples as:

$$\mathcal{L} = \frac{1}{2} ||\mathbf{x} - \hat{\mathbf{x}}||_2 \tag{3.5}$$

However, this conventional model has severe drawbacks. Possible parameter outputs become ineffective for multispectral data if Eqs. 3.3, 3.4, 3.5 are used directly. An example of possible parameter outputs for encoder part is illustrated in Fig. 3.1. From this plot, bias terms behave as one of pure signatures that regularly affect latent representations. Furthermore, some of the physical constraints are ultimately discarded in this structure.

These drawbacks can be explained in threefold: 1) Logistic activation functions underestimate the results because they violate the sparsity of material mixture for multispectral data [57, 23, 58]. Moreover, some theoretical limitations hamper the parameter optimization. 2) Inner products are not suitable for multispectral data. For

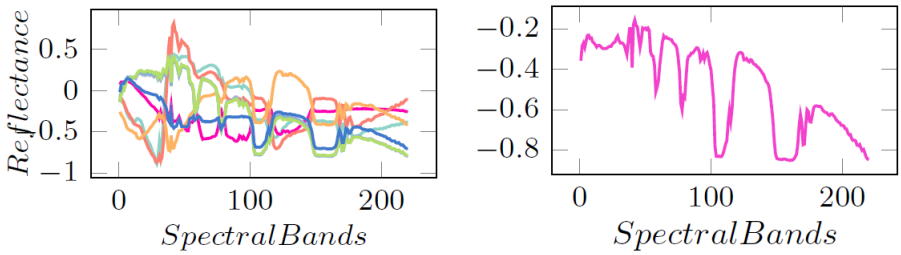

Figure 3.1: Visualization of learnable parameters for conventional autoencoder structure.

instance, if we consider the spectral variability, the inner matrix products inevitably result in diverse outputs for the same materials. 3) Similarly, standard Euclidean (l2) reconstruction discards the spectral variability exhibited from data. Moreover, there must be practical regularization terms that penalize the parameters for sparse responses. For all of these reasons, these limitations aggravate the conventional methods to obtain state-of-the-art performance.

For this purpose, the main objective of our method is to propose an autoencoder structure for endmember extraction on multispectral data. Specifically, we want to maintain the estimation power of autoencoders on latent representations. To improve the performance of autoencoders, we make critical modifications in the NN structure and optimization step.

First, bias term is removed from dense layers due to its adverse effects. However, this removal devastates the sparsity of responses so that additional layers, such as normalization layers, must be employed. Second, we replace the inner matrix product with spectral angle distance to compute fractional abundances. Lastly, we propose a novel cost function that maintains multi-spectral characteristics and regularizes the model for sparse responses.

For the sake of simplicity, we describe these modifications in different subsections.

3.2.1 Zero-biased Filters

Bias term is one of the critical learnable parameters in NN structures. Ultimately, this parameter determines which kernels are active and continue to contribute to the next layers [59]. However, bias term for multispectral data contradicts the idea that none of the materials can be constantly expected in all pixels. Since this term constantly influences the composition of pixels, it can be defined as one of the pure signatures at the end. Surely, this notion is not physically true. Hence, this term is removed from autoencoder structure as formulated in Eqs. 3.3, 3.4. Moreover, with this adaptation, the formulation retains the simplex set assumption, which constitutes the basis of conventional methods, as explained in [41, 60, 61].

Furthermore, activation function for decoder part in Eq. 3.4 is also discarded for effective error propagation step. To this end, $\mathbf{W}^d$ identifies as pure endmember signatures $\mathbf{E}$, which also provides a simplicity for the representations. Lastly, we utilize two separate parameter sets for encoder (i.e., $\mathbf{W}^e \in \mathbb{R}^{D\times K}$) and decoder (i.e., $\mathbf{W}^d \in \mathbb{R}^{K\times D}$) parts same as the conventional methods.

3.2.2 Sparsity and Nonlinearity

In section 2, sparsity and nonlinearity are two of the critical characteristics that must be preserved for robust representations. Similarly, these are critical features that must be strictly supplied for multispectral data [62, 63, 64, 65]. However, logistic function is not suitable for sparsity, even if it produces nonlinear responses.

For this purpose, rectified linear activation is replaced with logistic function in our model. In particular, ReLU limits the negative responses so that this increases the sparsity of latent representations. Since the responses are not logistically scaled, this layer helps for the rapid saturation of the learning step for parameters. Our structure is also tested with the variants of ReLU that scale the negative responses with additional terms. However, their performance is worse because the negative responses affect the relations of signatures (i.e., this violates the greater than zero constraint in Eqs. 3.1, 3.2).

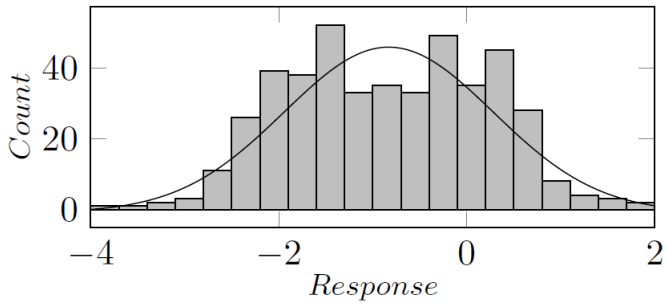

Figure 3.2: Distribution characteristic of batch normalization layer.

However, the removal and replacement of some critical components (i.e., bias and logistic functions) reduce the sparsity of kernel in the conventional structures. Hence learnable parameters become ill-posed [58] (i.e., small changes in space lead to oscillation). Hence, a normalization layer is introduced before ReLU activation function for encoder part. In particular, this layer imposes selectivity for the next activation function. Specifically, suppose that shift term is discarded in Eq. 2.19, it allows roughly 50% of activation responses to pass to the next layers. Hence, it automatically eliminates the outliers from latent representations and increases the sparsity at the end. Fig. 3.2 illustrates the output characteristics of a normalization layer. Note that batch statistics are used for normalization step. For the sake of simplicity, dense layer, batch normalization and activation function are denoted as $\mathbf{z}^p = f(\mathbf{x}; \theta)$ in the thesis.

This structure still leads to parameter overfitting for the dimensionality of data (i.e., dense layers). Hence, additional steps must be utilized for reliable abundance maps and endmember extraction. For this purpose, we use a two-step regularization approach. First, Dropout layer is integrated into the outputs of activation function. We empirically observe that for the scenes with strongly correlated materials, p value must be close to 0.5. Otherwise, p value is set to 1.0 as a default value. Second, a subset of top activations is hardly selected to increase the sparsity of decoder part [66] (i.e., $\mathbf{W}^d$). At the output of encoder part, l1 normalization is applied to ensure the sum-to-one constraint as follows:

$$\hat{\mathbf{y}} = \frac{\mathbf{z}^s}{||\mathbf{z}^s||_1 + \epsilon} \tag{3.6}$$

A stochastic gradient optimizer is used for parameter updates. Hence, all operations must be differentiable for proper error propagations. Eq. 3.6 is also differentiable except the value at zero. By using the chain rule, the partial derivative of l1 normalization layer is computed as follows:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}^s} = \frac{1}{\partial \mathbf{z}^s}(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} - \hat{\mathbf{y}} \sum_{k=1}^{K} \frac{\partial \mathcal{L}}{\partial z_k} sign(z_k^s)) \tag{3.7}$$

Here $\mathcal{L}$ indicates the cost function and $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}}$ defines the propagated error from upper layer up to the output of encoder part. $sign(.)$ denotes the sign operation that returns the element-wise results.

3.2.3 Robust Latent Representations

One of the essential features of NNs is to transform input data into latent representations with nonlinear transformations. After all, these layers contain latent correlations and distinct data characteristics.

Except for minor cases [67], a linear inner matrix product (i.e., dense layer) substantially outperforms almost all learning-based applications in the invisible domain. However, there is a variety of operations for multispectral fields that are superior for several tasks. Hence, we test the variants of inner product operations for our problem. In particular, robustness and separability of latent representations are analyzed in detail for these operations.

For this purpose, we design a simple test setup that evaluates the operations by comparing their similarity scores for the same and coherent signatures. Fig. 3.3 illustrates the score distributions of normalized inner product and spectral angle distance (SAD). Observe that inner product operation constantly yields high scores even for different materials. However, the outputs of spectral angler distance (SAD) are easily separable for the coherent ones. Similar observations are also reported in the literature [68, 69, 70, 71, 72].

We replace the inner matrix product with spectral angle distance to obtain robust and separable latent representations under these circumstances. Note that the use of angler operation instead of inner product also provides robustness for spectral variability.

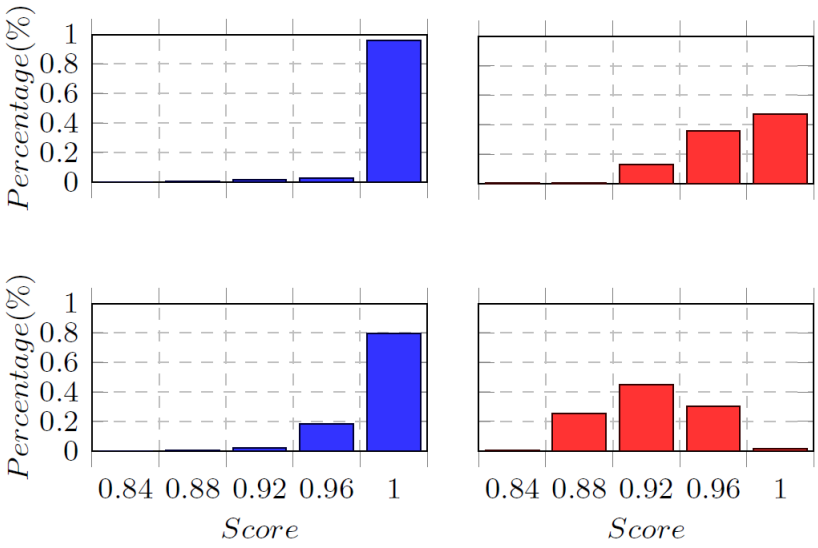

Figure 3.3: Normalized similarity score distributions for same (blue) and coherent (red) materials. Results for inner product (first row) and SAD (second row) are illustrated.

Spectral angle distance measures the spectral angle between two spectral signatures, and close to zero scores indicate high correlations between signatures. An argument for spectral angle distance is reported for its weakness to nonlinear cases [69, 70, 71]. Indeed, our method guarantees nonlinearity with activation functions.

Mathematically, spectral angle distance computes similarity scores as follows:

$$sc^{\text{SAD}}(\mathbf{x}_i, \mathbf{x}_j) = \cos^{-1}\left(\phi(\mathbf{x}_i, \mathbf{x}_j)\right) \tag{3.8}$$

Here, $\phi(.,.)$ estimates the angle of two samples $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{x}_j \in \mathbb{R}^D$:

$$\phi(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \mathbf{x}_j}{||\mathbf{x}_i||_2 ||\mathbf{x}_j||_2} \tag{3.9}$$

Notice that Eq. 3.8 computes a score that varies between $[0, \pi]$. However, scores must be normalized to $[0, 1]$ where 1 indicates the highest similarity score for decoder part. Hence, $sc^{\text{SAD}}(.,.)$ is projected to this range with:

$$sc^{\text{NSAD}}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{sc_{\text{SAD}}(\mathbf{x}_i, \mathbf{x}_j)}{\pi} \tag{3.10}$$

For encoder part, we employ the normalized spectral angler distance (NSAD) as $sc^{\text{NSAD}}(\mathbf{x}, \mathbf{W}^e)$ to transform the input data to latent representations than dense layer

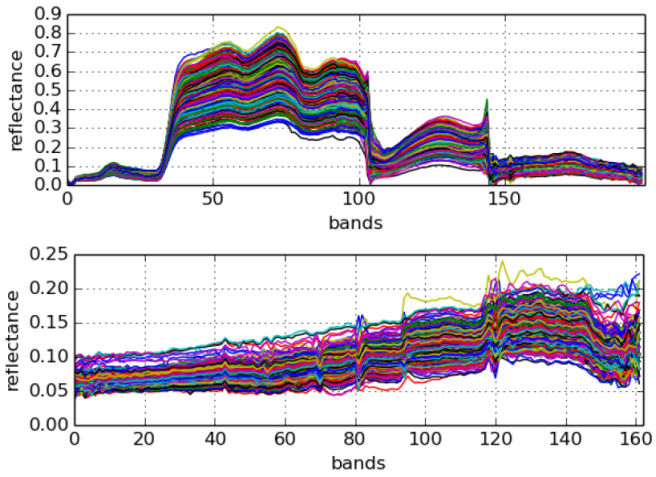

Figure 3.4: Spectral signatures from Jasper (first row) and Urban (second row) data correspond to two different pure materials.

as $\mathbf{W}^e\mathbf{x}$. NSAD is also differentiable, and the overall error is propagated with the chain rules by the gradient of k$^{\text{th}}$ row $\mathbf{w}_k$ of $\mathbf{W}^e$ where $\mathbf{w}_k \in \mathbb{R}^{1\times D}$ respect to the overall cost function $\mathcal{L}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_k} = \frac{\partial \mathcal{L}}{\partial sc_k^{\text{NSAD}}} \frac{\partial sc_k^{\text{NSAD}}}{\partial sc_k^{\text{SAD}}} \frac{\partial sc_k^{\text{SAD}}}{\partial \phi_k} \frac{\partial \phi_k}{\partial \mathbf{w}_k} \tag{3.11}$$

Here, $sc^{\text{NSAD}} = sc^{\text{NSAD}}(\mathbf{x}, \mathbf{w}_k)$, $sc^{\text{SAD}} = sc^{\text{SAD}}(\mathbf{x}, \mathbf{w}_k)$ and $\phi = \phi(\mathbf{x}, \mathbf{w}_k)$ are the simplified versions of original formulations. Partial derivatives of these formulations are written as:

$$\frac{\partial sc_k^{\text{NSAD}}}{\partial sc_k^{\text{SAD}}} = \frac{-1}{\pi} \tag{3.12}$$

$$\frac{\partial sc_k^{\text{SAD}}}{\partial \phi_k} = \frac{-1}{\sqrt{1-\phi_k^2}} \tag{3.13}$$

$$\frac{\partial \phi_k}{\partial \mathbf{w}_k} = \frac{\mathbf{x}}{||\mathbf{w}_k||_2||\mathbf{x}||_2} - \frac{\mathbf{w}_k * (\mathbf{w}_k\mathbf{x})}{||\mathbf{w}_k||_2^3||\mathbf{x}||_2} \tag{3.14}$$

Until now, we have presented our contributions to the NN structure. Next, we focus on the optimization step with additional terms.

3.2.4 Parameter Learning

As explained, spectral variability affects the mixture of materials, especially for real data. Fig. 3.4 illustrates the pure material samples from real data. Hence, the standard Euclidean norm cannot be sufficient to address severe illumination changes and material variability. On the other hand, spectral angler distance partially overcomes these limitations even if spectral signatures are varied for materials.

Furthermore, additional terms for the overall function must be added to impose the sparsity and smoothness for learnable parameters. After all, the overall cost function is formulated as follows:

$$\mathcal{L}_{re} = \frac{\lambda_0}{2}||\mathbf{x} - \hat{\mathbf{x}}||_2 + \lambda_1 H(1, sc^{\text{NSAD}}(\mathbf{x}, \hat{\mathbf{x}})) + \lambda_2||\mathbf{z}^p||_1 + \lambda_3||\mathbf{W}^e||_2 + \lambda_4||\mathbf{W}^d||_2 \tag{3.15}$$

where $sc^{\text{NSAD}}(.,.)$ denotes the normalized spectral angle distance that measures the distance between original and reconstructed data.

Notice that there is a large number of hyper-parameters that tunes the contributions of these cost terms. In particular, λ_0 and λ_1 control the influence of Euclidean and Negative Log-Likelihood cost terms. Empirically, we observe that Euclidean cost cannot be directly set to zero, so that λ_0 and λ_1 are accepted as 0.01 and 10 in the experiments.

Furthermore, a sparsity term is inserted [73], and its contribution is scaled with λ_2. This term penalizes latent representations that are constantly active for all input data. Hence, sparsity is ultimately promoted. Similarly, the default value of λ_2 is set to 0.1, which is relatively high. Note that this value can be increased or decreased due to the mixture level of materials. Lastly, smoothness terms are used for encoder and decoder parts. λ_3 and λ_4 are set to 10^{-5}.

As explained, a stochastic gradient-based Adam optimizer is utilized [21]. Learning rate and iteration number are fixed to 0.001 and 400K for all datasets. Batch-size is set to 64 to limit the computation load. VCA-like methods [41, 40, 74] are used for the initialization of learnable parameters in both encoder and decoder parts (i.e., $\mathbf{W}^e$ and $\mathbf{W}^d$) to take advantage of pre-trained parameters. This initialization is practical

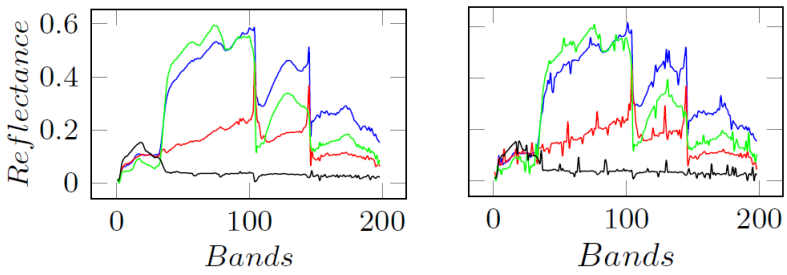

Figure 3.5: Four multispectral signatures that are corrupted with the concept of Denoising Autoencoder.

for more stable solutions and speeds up the training process.

3.2.5 Extra Step for Overfitting and Generalization

We defined additional layers, such as Dropout, to prevent overfitting due to the input dimensions in the previous steps. Moreover, another concept (i.e., called *Denoising Autoencoder* [2]) is introduced to ensure both overfitting and generalization capacity for the model. The purpose is to corrupt the input data with an additive Gaussian Noise for each iteration, and the reconstruction costs are estimated from the original data. This assumption enables the model to generalize the solution for nearby samples in the input space. This scheme explicitly leads the model to exploit possible deformations, such as illumination changes, sensor noise, in advance.

For this purpose, input data is disturbed with Gaussian noise as $(\tilde{\mathbf{x}}|\mathbf{x} \sim \mathcal{N}(x|0, \sigma^2))$. In fact, this scheme can impede the behavior of models for noisy data [75]. Hence, σ value needs to adapt to the scenes. To get a glimpse of the possible outputs for this step, Fig. 3.5 visualizes both the original input and the corrupted version.

3.2.6 Fractional Abundance Estimation

As stated, abundance maps and endmember signatures are optimized together in our method. However, we can estimate these values in two different modes: 1) We can immediately use the outputs of encoder part in our model. 2) We can use a separate

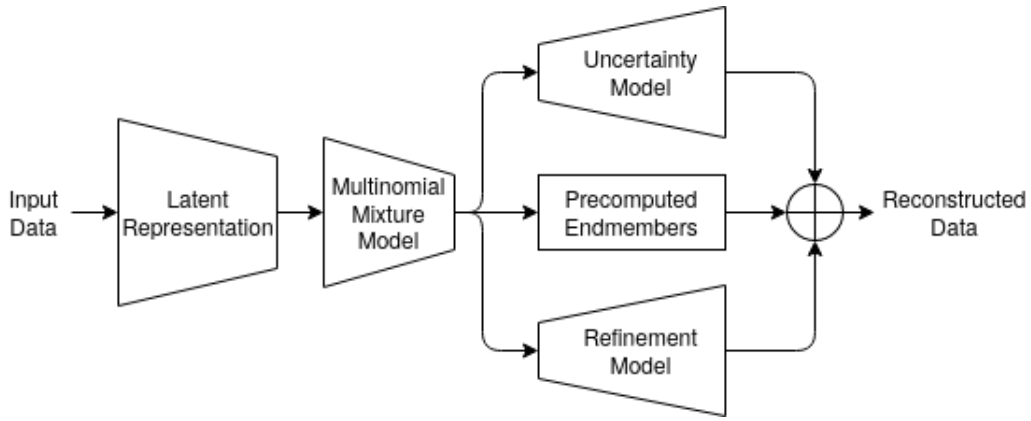

Figure 3.6: Flow of deep spectral convolution network for spectral unmixing.

model by exploiting the estimated endmember signatures.

In the experiments, we observe that the abundance maps estimated by our method are not accurate, even if the parameters are optimized in an end-to-end manner, and endmember extraction step achieves state-of-the-art performance. One of the main reasons is the batch normalization [27]. For this purpose, we exploit another model (i.e., *Simplex Projection Unmixing* (SPU) [74, 50]). Actually, this model is not trainable with our method so that the proposed method is first trained and then abundance maps are estimated. Moreover, SPU model maintains the non-negativity and sum-to-one constraints in the solutions.

3.3 Deep Spectral Convolution Network for Multi-Spectral Unmixing

The dimensionality of multispectral data is one of the disadvantages, especially for unsupervised learning. The main reason is that irregularities in input spaces can lead to holes in the partitioning step. This problem inevitably leads to overfitting for parameters. A practical solution in the literature is to use supervised annotations by filling these holes with semantic knowledge [76, 13]. Another solution aims to divide the data into several parts, and common representations are estimated to reduce the number of learnable parameters in the models [77].

In fact, supervised annotations are not exploited for spectral unmixing (i.e., it is impractical). Therefore, the total number of learnable parameters must be revised to

simplify the problem while improving performance.

For this purpose, dense layers with affine transformations are replaced with 1D convolution layers. Since fewer parameters are optimized, the stacks of convolutions are exploited to boost the representation capacity with deep hierarchies. However, the chance of kernel instability is critical so that the NN structure should be reformed with novel cost functions.

In fact, our structure is almost the same as in the sparse auto-encoder. First, the input data is converted into abundance maps in encoder part $\mathbf{y} = Enc(\mathbf{x}; \theta^e)$. For our method, this part is composed of two steps. First, high-dimensional input data is hierarchically summarize to low-dimensional space, and a latent representation is computed as $\mathbf{z} = H(\mathbf{x}; \theta^e)$ where $\mathbf{z} \in \mathbb{R}^M$. Here, the size of latent representations $\mathbf{z}$ needs not necessarily match with the number of fractional abundances K. Hence, M denotes the dimension of latent representations (i.e., empirically, it is set to 10). Later, abundance maps are estimated by using these latent representations with a multinomial function $\hat{\mathbf{y}} = G(\mathbf{z}; \theta^p)$. Since our method targets to find abundance maps only, decoder part consists of a single dense layer, which corresponds to the matrix multiplication (i.e., $\mathbf{E}\hat{\mathbf{y}}$) for the reconstruction of the original data. The flow of the proposed method is illusrated in Fig. 3.6.

3.3.1 Stack of Spectral Convolutions

The replacement of dense layer with 1D convolutions provides an advantage to stack consecutive non-linear kernels in multiple layers. However, the conventional model must be redefined based on this structure. Precisely, parametric ReLU is used as an activation function to improve the knowledge flow between layers. The formulation of this function is already defined in Eq. 2.16. As a simple reminder, negative input responses are scaled with a learnable parameter so that the gradients for negative values are also exploited in parameter updates. Hence, efficient error propagation allows to form a deeper structure, and rich latent representations are obtained at the end. Notice that nonlinearity is still valid for this function.

The second modification for this structure is that a novel normalization layer is uti-

Table 3.1: Structure of DSCN++. Conv1D, SN and BN denote 1D convolution, spectral normalization and batch normalization respectively.

Index	**Index of Inputs**	**Operation(s)**	**Output Shape**
(1)	-	Input Spectra	D
(2)	(1)	Conv1D(10, 21, 1)	D $\times$ 10
(3)	(2)	LReLU + AvgP(5) + BN	D/5 $\times$ 10
(4)	(3)	Conv1D(10, 3, 1)	D/5 $\times$ 10
(5)	(3)	Conv1D(10, 5, 1)	D/5 $\times$ 10
(6)	(3)	Conv1D(10, 7, 1)	D/5 $\times$ 10
(7)	(4),(5),(6)	Concat	D/5 $\times$ 30
(8)	(7)	LReLU + AvgP(2) + SN	D/10 $\times$ 30
(9)	(8)	Conv1D(5, 3, 1)	D/10 $\times$ 10
(10)	(9)	LReLU + AvgP(2) + SN	D/20 $\times$ 10
(11)	(10)	Linear(M) + LReLU	M

lized. Batch normalization is typically selected as a baseline due to its normalization and regularization features. In particular, batch normalization introduces some noise at train time, which also regularizes latent representations. However, we find that parameter updates are deteriorated when data is noisy. As expected, multispectral data exhibit noisy measurements as it is one of the sensor features. Also, an additional noise term is used to formulate this term in the solution for linear and nonlinear models. Hence, we present a spectral normalization layer. The only difference is that spectral statistics are used instead of batch statistics (i.e., mean and variance values are calculated from data characteristics).

Furthermore, we modify the order of convolution and normalization layers. This rearrangement enhances the selectivity of representations by feeding more statistical knowledge to the convolution layers. Notably, the detailed analysis of this operation will be analyzed in the next chapter.

Third, convolution operations only extract knowledge from small windows. However, this can ruin the fact that the size of the informative part of data cannot fit to constant

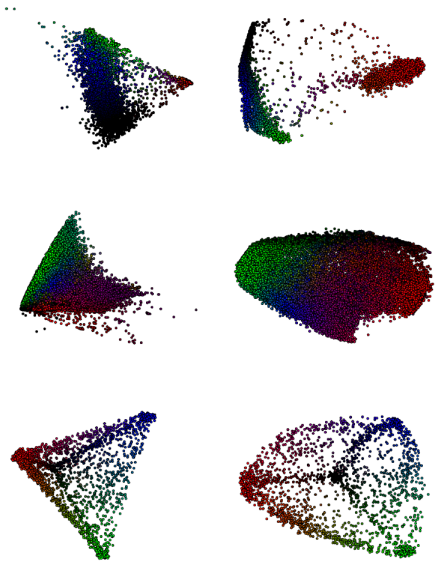

Figure 3.7: Data distributions before (first column) and after (second column) applying DSCN++ method. Each color indicates the abundance maps of different pure materials (i.e., red, green, blue or black). Observe that our model maps the high-dimensional material signatures to a more separable space based on their abundances.

windows. Hence, we use *Inception module* [14] that has multiple kernels of different sizes. Also, these kernels are performed on the same input data simultaneously. Hence, rich representations are estimated. Complete structure with input/output relations for all layers is visualized in Table 3.1.

Observe that this encoder part transforms high-dimensional data into low-dimensional ones by improving the representation capacity. For this purpose, we expect to transform unstructured input space to more separable space to obtain precise abundance maps. To demonstrate the superiority of our method, Fig. 3.7 illustrates the input data distributions and latent representation estimated by our method. Note that colors are discriminative with their real abundance values. From these results, it is clear that our method blindly achieves more separable space for multispectral data.

3.3.2 Multinomial Mixture Kernel For Abundance Maps

In the previous part, we estimate a high-level latent representation for multispectral data. Now, abundance maps are computed all in an unsupervised manner. For this purpose, a NN multinomial mixture model is utilized. As explained, distribution-based methods reproduce signatures as probabilistic distributions. Primary methods employ a unimodal distribution as:

$$p(\mathbf{y}|\mathbf{z}) \sim \mathcal{N}(\mathbf{z}|\mu, \mathbf{\Sigma}) \tag{3.16}$$

Abundance maps are intuitively estimated from $\mathbf{z}$ by using a single Gaussian distribution where μ and $\mathbf{\Sigma}$ indicate the mean and covariance matrix. In fact, this formulation underestimates the solution when spectral variability is taken into account. Therefore, Eq. 3.16 is restored with *Gaussian Mixture Model (GMM)* by computing the distributions from statistical perspective:

$$p(\mathbf{y}|\mathbf{z}) \sim \sum_{n \in \kappa} p(n) p(\mathbf{y}|\mathbf{z}, n) \tag{3.17}$$

Here, n is a random variable in sample space κ. Furthermore, we extend these probabilistic values as $p(n) \sim \alpha_n$ and $p(\mathbf{y}|\mathbf{z}, n) \sim \mathcal{N}(\mathbf{z}|\mu_n, \mathbf{\Sigma}_n)$. Hence, this model interprets the abundance maps as multinomial distributions with the mixture of multivariate Gaussian distributions:

$$p(\mathbf{y}|\mathbf{z}) \sim G(\mathbf{z}; \theta^p) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(\mathbf{z}|\mu_n, \mathbf{\Sigma}_n) \tag{3.18}$$

where N is the total number of mixture components by constraining the value by $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n = 1$.

$$\mathcal{N}(\mathbf{z}|\mu_n, \mathbf{\Sigma}_n) = \frac{1}{(2\pi)^{\frac{N}{2}} |\mathbf{\Sigma}_n|^{\frac{1}{2}}} \exp\left(\frac{-1}{2}(\mathbf{z} - \mu_n)^T \mathbf{\Sigma}_n^{-1} (\mathbf{z} - \mu_n)\right) \tag{3.19}$$

The concept of multinomial distributions is adopted for spectral unmixing because abundance maps are overly represented so that spectral distortions can be handled appropriately. Hence, for effective representations, N should be selected as $N >> K$. α_n is the weight term that determines the contribution of n^{th} of multimodal component to abundance values.

For this purpose, we implement this framework with a NN model in our method. We prefer to use a NN model since the overall system can be trained in an end-to-end manner. In fact, NN parameters are optimized with a new type of cost function for the first time in the literature. The details of this cost function will be discussed in the following subsection.

If we extend Eq. 3.18 to divide this term into two learnable parts, the equation is expressed as:

$$G(\mathbf{z};\theta^{p}) = \sum_{n=1}^{N} \beta_n g_n(\mathbf{z}) \tag{3.20}$$

where,

$$\beta_n = \frac{\alpha_n}{(2\pi)^{\frac{N}{2}} |\mathbf{\Sigma}_n|^{\frac{1}{2}}} \tag{3.21}$$

$$g_n(z) = \exp\left(\frac{-1}{2}(\mathbf{z}-\mu_n)^T \mathbf{\Sigma}_n^{-1}(\mathbf{z}-\mu_n)\right) \tag{3.22}$$

As noticed, at least two sets of parameters must be optimized for this configuration. Hence, two separate single-layer subnetworks are used for this purpose as $v_n(\mathbf{z};\theta^{p1}) = g_n(\mathbf{z})$ and $c_n(\mathbf{z};\theta^{p2}) = \beta_n$ where $n = 1, ..., N$.

In addition, the quadratic form of $g_n(\mathbf{z})$ is renounced in our formation, and multivariate Mahalanobis distance is normalized with a sigmoid function to calculate probabilistic relations. To this end, the normal distribution resembles with a cumulative distribution function as follows:

$$v_n(\mathbf{z};\theta^{p1}) = \frac{1}{1+\exp(\mathbf{w}_n^{p1}\mathbf{z})} \tag{3.23}$$

Similarly, β_n term is calculated with a NN model and sum-to-one constraint is retained with softmax activation:

$$c_n(\mathbf{z};\theta^{p2}) = \frac{\exp(\mathbf{w}_n^{p2}\mathbf{z})}{\sum_{n=1}^{N}\exp(\mathbf{w}_n^{p2}\mathbf{z})} \tag{3.24}$$

In the end, all subparts of encoder can be trained in an end-to-end manner.

In fact, the idea to imitate multimodal mixture models with NN networks is not original. Precisely, other attempts are already made in the literature. For instance,

one method employes multiple subnetworks for each multimodal component in this sense [78]. However, the total number of subnetworks is large. This feature ultimately introduces extra complexity and parameters for the optimization step. For another method, mean and covariance kernels are directly implemented in the model than the parameterization with linear NN layers [79]. In this case, we empirically observe that the performance becomes low even if more stable results are obtained. As explained, all operations are differentiable in our formulation, but it is not applicable for this method.

Even if these structural modifications formulate the problem perfectly, it still requires a specific cost function to have the spectral variability. In the next subsection, we will present this novel cost function and discuss the advantages of this cost function, especially for the multinomial mixture kernel parameters.

3.3.3 Parameter Updates for Multinomial Mixture Kernel

Similarly, Eq. 3.15 is used as the overall cost function to optimize learnable parameters in our method. As discussed, the property of spectral variability cannot be fully covered for abundance maps, even if spectral angle distance (SAD) is robust for this property. Hence, we exploit adversarial model, especially for the parameters of multinomial mixture kernel. First, we provide theoretical and mathematical explanations for this cost function.

As a note, *Expectation Maximization* (EM) is the most popular technique to estimate GMM parameters [80]. Furthermore, this technique guarantees a stable solution by optimizing negative log-likelihood function. Note that we also use this function in our method for parameter updates. However, the studies show that it suffers from obtaining an ideal solution for several reasons, such as the random initialization of parameters and the discontinuity. KL-divergence similarly has these drawbacks since it is equivalent to the maximization of log-likelihood. Its weakness can be summarized in twofold: First, since GMM is continuous and locally Lipschitz [28], KL-divergence cannot fulfill these properties. Also, KL-divergence is sensitive to parameter initialization due to the weakness of local minimum. For further details, we recommend to read [81, 28, 82, 83, 84]. In addition, the studies show that Euclidean cost function

yields overly smooth results that hardens to reach accurate solutions [85].

These drawbacks undermine the model to obtain stable and state-of-the-art performance for mixture kernels. For this purpose, adversarial model is utilized in our method.

As noticed, adversarial models (GAN) have demonstrated the ability to learn any data distribution for various machine learning and computer vision tasks. Ultimately, it aligns known and unknown distributions by transforming input space step-by-step. However, for similar reasons, the original GAN model remains incapable of random parameter initialization and discontinuity since the formulation can be simplified to a variant of KL-divergence, as stated in Eq. 2.23.

Hence, *Wasserstein GAN* is employed that provides several benefits compared to the original version. Specifically, Wasserstein distance is continuous and differentiable almost everywhere. Also, since it has a smooth cost function, the initialization is not a serious problem compared to KL-divergence.

Mathematically, Wasserstein distance $W(p, q)$ measures the distance between two distributions p and q where they generate variables as $\mathbf{x}^p$ and $\mathbf{x}^q$. Kantorovich-Rubinstein duality minimizes the distance as:

$$\max_{\|f\|_L \leq 1} \big(\mathbb{E}_{\mathbf{x}^p}[f(\mathbf{x}^p)] - \mathbb{E}_{\mathbf{x}^q}[f(\mathbf{x}^q)] \big) \tag{3.25}$$

Here $f(.)$ is a 1-Lipschitz function. Inspired by the structure of *Variational AutoEncoder* (VAE) [86], this duality can be adopted to our formulation in Eq. 3.18 as:

$$\max_{\|f\|_L \leq 1} \left(\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] - \mathbb{E}_{\hat{\mathbf{x}}} \left[f\Big(\sum_{k=1}^{K} (\mathbf{e}_k + \gamma_k) \sum_{n=1}^{N} \beta_n g_n(\mathbf{z}) \Big) \right] \right) \tag{3.26}$$

Here γ_k is omitted for the simplicity and β_n value is accepted as not random. Note that this simplification is resolved with extra steps. In addition, the generated sample from multinomial components is denoted as $\hat{\mathbf{x}}$. With these observations, the formulation is simplified as:

$$\max_{\|f\|_L \leq 1} \left(\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] - \sum_{k=1}^{K} \mathbf{e}_k \sum_{n=1}^{N} \beta_n \mathbb{E}_{\hat{\mathbf{x}}}[f(g_n(\mathbf{z}))] \right) \tag{3.27}$$

Table 3.2: Structure of discriminative network.

Index	Index of Inputs	Operation(s)	Output Shape
(1)	-	Input Spectra	D
(2)	(1)	Conv1D(5, 21, 5)	D/5 $\times$ 5
(3)	(2)	BN + PReLU	D/5 $\times$ 5
(4)	(3)	Conv1D(10, 5, 2)	D/10 $\times$ 10
(5)	(4)	BN + PReLU	D/10 $\times$ 10
(6)	(5)	Conv1D(20, 5, 2)	D/20 $\times$ 20
(7)	(6)	BN + PReLU	D/20 $\times$ 20
(8)	(7)	Linear(5)	D/20 $\times$ 5

Observe that Wasserstein GAN is used for parameter updates. Furthermore, there is an additional gradient penalty term in Wasserstein GAN for stable solutions:

$$\mathcal{L}_{disc} = \mathbb{E}[d(\mathbf{x};\theta^d)] - \mathbb{E}[d(\hat{\mathbf{x}};\theta^d)] + \lambda_{pq}\mathbb{E}[(\|\nabla_{\tilde{\mathbf{x}}} d(\tilde{\mathbf{x}};\theta^d)\|_2 - 1)^2]. \tag{3.28}$$

Here, $d(.,.)$ is the discriminative module and it is a 1-Lipschitz function. Furthermore, $\tilde{\mathbf{x}}$ is uniformly sampled data from real $\mathbf{x}$ and reconstructed $\hat{\mathbf{x}}$ data. λ_{pq} scales the contribution of penalty term that is set to 10.

Last but not least, the idea of *PatchGAN* [87] is adopted in our method. Hence, local patches are fed to the discriminative model than full data, and local characteristics (i.e., local spectral variability) are practically captured. The structure for discriminative network is also presented in Table 3.2.

3.3.4 Extra Steps for Spectral Variability

We describe the overall framework for our method. As noticed, the aim is to find abundance maps from precomputed endmember signatures accurately. Therefore, decoder part (i.e., $\mathbf{W}^d$) is not updated at train time.

However, by merely expanding Eq. 3.2, Two sets of equations are unveiled. One of which is the projection of endmember signatures with abundance maps $\mathbf{e}_k y_k$ and

another $\gamma_k y_k$ defines the impact of uncertainty term to each abundance value. Empirically, we observe that the use of these two extra modules can eventually help to improve performance. Therefore, additional uncertainty and refinement NN models are presented at decoder part.

For this purpose, we introduce a NN model $u(\hat{\mathbf{y}}, \eta; \theta^u)$ that takes the output of multinomial mixture kernel and random noise as input. This model computes an additive full-size spectral signature at the end. It is mainly composed of a stack of dense layers that incrementally enlarges the dimension of input data (i.e., $K + L \rightarrow 20 \rightarrow D$). Here, K and L denote the number of abundance maps and the size of random noise. Observe that there is a random noise η to mimic the spectral variability. TanH activation function is used at the output of this module to scale the responses.

Second, we utilize a refinement NN model $u(\hat{y}; \theta^r)$ that enables small updates to precomputed endmember signatures. Hence, instead of drastic updates on these signatures, we impose small refinements for more stable solutions. Similarly, TanH activation is utilized. Note that this model also computes a full-size additive spectral signature.

The final output of decoder part is reformulated as:

$$\mathbf{x} = \mathbf{E}\mathbf{y} + \lambda_u u(\mathbf{y}, \eta; \theta^u) + \lambda_r u(\mathbf{y}; \theta^r) \tag{3.29}$$

Here, λ_r and λ_u rescale the contributions of refinement and uncertainty NN models. Empirically, these values are set to 0.05 and 0.1, respectively.

3.3.5 Implementation Details

Our method has four main models: spectral convolutions (called DSCN), a multinomial mixture model, refinement models and a discriminative network. Note that all of these modules are differentiable, and an end-to-end learning scheme can be exploited for parameter updates. However, each module is updated with a separate cost function. For the sake of clarity, these modules and their specific cost functions are explained in detail.

Only the reconstruction cost function is used for spectral convolution, and its pa-

rameters θ_e are optimized. Similarly, the parameters of discriminative module θ^d are trained with the adversarial cost as:

$$\min_{\theta^e} \mathcal{L}_{re} \tag{3.30}$$

$$\min_{\theta^d} \mathcal{L}_{disc} \tag{3.31}$$

On the other hand, cost functions of refinement θ_r and uncertainty models θ_u are scaled with coefficients because their goal is to make small changes on the pretrained endmember signature. Empirically, the values of these coefficients are set to 0.001 as:

$$\min_{\theta^r} 0.001\mathcal{L}_{re} \tag{3.32}$$

$$\min_{\theta^u} 0.001\mathcal{L}_{gen} \tag{3.33}$$

Observe that the uncertainty model uses adversarial cost since Wasserstein distance has several theoretical benefits to the conventional cost functions for stable and accurate results. We have already explained the advantages throughout the thesis. Lastly, the mixture model is divided into two sets of parameters in Eqs. 3.34, 3.35 as θ^{p1} and θ^{p2}. The main reason is that the parameters for normal distributions must be optimized with adversarial cost, as proved in Eq. 3.27. Similarly, this learning step helps to have spectral variability in the solutions:

$$\min_{\theta^{p1}} \mathcal{L}_{re} \tag{3.34}$$

$$\min_{\theta^{p2}} 0.1\mathcal{L}_{gen} + 0.01\mathcal{L}_{re} \tag{3.35}$$

In addition, Adam stochastic optimizer is used by setting β_1 as 0.7, similar to the sparse autoencoder [21]. Mini-batch size is fixed to 64. Learning rate and the number of training iterations are set to 0.002 and 10K, respectively. As noticed, training iterations are small since abundance maps are merely calculated from precomputed endmember signatures.

Figure 3.8: Color images of Synthetic, Urban, Samson, Jasper, Cuprite, University of Pavia and Mississippi Gulfport data, respectively

3.4 Experiments

Experiments are conducted on various well-studied real and synthetic datasets. Furthermore, we report quantitative and qualitative results by comparing them with baseline methods. Note that ground truth annotations and pixel-wise abundance maps are available for almost all datasets. Otherwise, we will inform the readers at the beginning of the data sections.

First, we will describe the details of these datasets and evaluation metrics used in the experiments. Baseline methods will be briefly summarized later. After all, the experimental results will be provided for individual datasets.

3.4.1 Multispectral Datasets

We evaluate our methods on both synthetic and real data. Color images are illustrated in Fig. 3.8. In fact, we first need to discuss that it can be hard to simulate synthetic data for extreme cases. In particular, nonlinearity and spectral variability cannot be created correctly (i.e., even if Hapke Model or Bilinear Mixture Model (BMM) is used). Therefore, our experiments are mostly performed on real data. Here, we provide all information for these datasets.

In addition, abundance map estimation for the ground truth is impossible for real data. Therefore, supervised unmixing problem is resolved. For the detail explanations, we recommend to read [88].

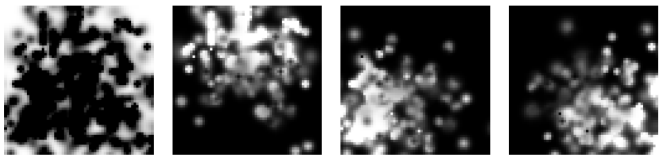

Figure 3.9: Abundance maps for Synthetic dataset.

3.4.1.1 Synthetic Data

Four pure materials are used to simulate the data from ASTER spectral library [38]: Limestone ($\#1$), Basalt ($\#2$), Concrete ($\#3$) and Asphalt ($\#4$). Spectral dimension is 200 and spectral range varies from 0.4μm to 14μm. Spatial resolution of data is 60×60. This data is constructed by the fact that Limestone is used as a background material. The rest (i.e., basalt, concrete and asphalt) is randomly placed to this scene as Gaussian-shaped distributions. Abundance maps are shown in Fig. 3.9.

3.4.1.2 Urban Data

There are four materials [89, 90]: Asphalt ($\#1$), Grass ($\#2$), Tree ($\#3$), Roof ($\#4$). Spectral dimension is 210 and spectral range varies from 0.01μm to 25μm. Spatial resolution of data is 307×307. Several channels ($1-4$, 76, 87, $101-111$, $136-153$ and $198-210$) are removed due to water-vapor absorption and atmospheric effects.

3.4.1.3 Jasper Data

The scene has four pure materials [91]: Tree ($\#1$), Water ($\#2$), Soil ($\#3$) and Road ($\#4$). Spectral dimension is 224 and spectral range varies from 0.38μm to 25μm. Spatial resolution of data is 100×100. Due to the atmospheric effects and water-vapor absorption, some of the channels ($1-3$, $108-112$, $154-166$ and $220-224$) are discarded.

3.4.1.4 Samson Data

Data is generated by the SAMSON sensor, and there are three materials [90]: Soil ($\#1$), Tree ($\#2$) and Water ($\#3$). Spectral dimension is 156 and spectral range varies from 0.4μm to 0.9μm. Spatial resolution of data is 95×95.

3.4.1.5 Cuprite Data

AVIRIS sensor is used, and there are twelve pure materials [89, 90, 92]: Alunite ($\#1$), Andradite ($\#2$), Buddingtonite ($\#3$), Dumortierite ($\#4$), Kaolinite$_1$ ($\#5$), Kaolinite$_2$ ($\#6$), Muscovite ($\#7$), Montmorillonite ($\#8$), Nontronite ($\#9$), Pyrope ($\#10$), Sphene ($\#11$) and Chalcedony ($\#12$). Spectral dimension is 188 and spectral range varies from 0.4μm to 2.5μm. Spatial resolution of data is 250×190. Some of the channels ($1-2$, $104-113$ and $148-167$) are discarded due to the atmospheric effects and water-vapor absorption.

3.4.1.6 University of Pavia Data

The ROSIS sensor over Pavia-Italy records data, and there are seven materials [75]: Asphalt-Bitumen ($\#1$), Meadows ($\#2$), Trees ($\#3$), Metal Sheets ($\#4$), Bare Soil ($\#5$), Gravel-Bricks ($\#6$) and Shadow ($\#7$). Spectral dimension is 103 and spectral range varies from 0.43μm to 0.86μm. Spatial resolution of data is 610×340. In the original data, there are eight materials. However, bricks-gravel and asphalt-bitumen yield nearly identical spectra. Hence, we consider these materials as one for all methods.

3.4.1.7 Mississippi Gulfport Data

The field of University of Southern Mississippi-Gulfpark Campus is used, and there are seven materials [93]: Trees ($\#1$), Grass Pure ($\#2$), Grass Ground ($\#3$), Dirt $\&$ Sand ($\#4$), Road ($\#5$), Shadow($\#6$) and Building ($\#7$). Spectral dimension is 64 and spectral range varies from 0.36μm to 1.04μm. Similarly, it contains eleven materials

in the original data. However, we eliminate sidewalk, yellow curb, cloth panels, and water classes due to class imbalance problem.

3.4.2 Metrics

Two metrics are deployed to evaluate endmember extraction and spectral unmixing methods: Spectral Angle Distance (SAD) and Root Mean Square Error (RMSE). Precisely, SAD is used to compare predicted endmember signatures with ground truth signatures:

$$\text{SAD}(\mathbf{x}, \hat{\mathbf{x}}) = \cos^{-1}\left(\frac{\mathbf{x}\,\hat{\mathbf{x}}}{\|\mathbf{x}\|_2\|\hat{\mathbf{x}}\|_2}\right) \tag{3.36}$$

On the other hand, RMSE is used to measure the difference between predicted abundance values and ground truth as follows:

$$\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{K}\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}. \tag{3.37}$$

Note that for both metrics, lower is better. For reliability of assessments, tests are repeated 20 times for all methods, so mean and standard deviation are reported.

3.4.3 Baselines

Our methods, sparse autoencoder network for endmember extraction (Endnet) and deep spectral convolution network for multispectral unmixing (DSCN++), are compared by open-source methods. Note that we reproduce all the results for open-source methods. For the sake of simplicity, we group these baseline methods accordingly, whether they are compared with Endnet or DSCN++ method.

For Endnet method, the following baselines are selected:

- Vertex Component Analysis (VCA) [41] (code is available on `http://www.lx.it.pt/bioucas/code.htm`)
- Minimum Volume Simplex Analysis (MVSA) [46] (code is available on `http://www.lx.it.pt/bioucas/code.htm`)

- Sparsity Promoting Iterated Constrained Endmembers (SPICE) [45] (code is available on `https://github.com/GatorSense/SPICE`)
- Spatial Compositional Model (SCM) [94] (code is available on `https://github.com/zhouyuanzxcv/Hyperspectral`)
- Distance-MaxD (DMaxD) [34] (code is available on `https://sites.google.com/site/robheylenresearch/code`)
- $l_{1|2}$-NMF [90] (code is not available, the results are provided from original work)
- DgS-NMF [90] (code is not available, the results are provided from original work)

For DSCN++ method, the following baselines are also selected:

- Linear Mixture Model (LMM) [95] (code is available on `https://sites.google.com/site/robheylenresearch/code`)
- Generalized Bilinear Model (GBM) [52] (code is available on `https://sites.google.com/site/robheylenresearch/code`)
- Post-Nonlinear Mixing Model(PPNM) [96] (code is available on `https://sites.google.com/site/robheylenresearch/code`)
- Multilinear Mixing Model (MLM) [95] (code is available on `https://sites.google.com/site/robheylenresearch/code`)
- Normal Compositional Model (NCM) [97] (code is available on `https://github.com/napovargas/BayesNCM`)

3.4.4 Sparse Autoencoder Network for Endmember Extraction (Endnet)

3.4.4.1 Parameter Setting

Default parameter settings are retained for almost all baseline methods. Only the parameters of SCM algorithms are tuned for each dataset. Multilinear Mixing Model

Table 3.3: Average computation time for methods on three datasets.

Computation Time (Sec)			
	Urban	Samson	Jasper
VCA-MLM	≈ 3600	≈ 349	≈ 352
SCM	≈ 1021	≈ 187	≈ 472
EndNet-DMaxD-SPU (ours)	≈ 914	≈ 789	≈ 855

(MLM) [98] is used for MVSA, VCA and DMaxD algorithms, which provides the best performance among other non-linear methods to estimate abundance maps (Note that we also tested these algorithms with Fully-Constrained Least Square (FCLS), Generalized Bilinear Model (GBM) [99], Postnonlinear Mixing Model (PPNM) [100] and Simplex Projection Unmixing (SPU). MLM yielded the best performance). In addition, SPICE and SCM compute their abundance map results.

For SCM method, different hyperparameter sets are used for individual datasets. For instance, $\beta_1 = 10$, $\beta_2 = 5$ and $\rho = 0.01$ are determined for Samson and Jasper datasets while these values are set to $\beta_1 = 10$, $\beta_2 = 10$ and $\rho = 0.01$ for Urban and Cuprite datasets. In addition, $\beta_1 = 10$, $\beta_2 = 10$ and $\rho = 0.01$ are tuned for University of Pavia dataset. Lastly, $\beta_1 = 10$, $\beta_2 = 5$ and $\rho = 0.01$ are used for Mississippi Gulfport dataset.

For EndNet method, two hyper-parameters must be tuned by the users: p value defined in Dropout layer and λ_2 in Eq. 3.15 value for denoising autoencoder. Default values are set to $p = 1.0$ and $\lambda_2 = 0.1$. Note that these values can be redefined based on the characteristics of the scenes. In the experiments, EndNet-VCA and EndNet-DMaxD denote the initialization algorithms for our learnable parameters as VCA and DMaxD, respectively.

Note that experiment results for each dataset are reported in the individual section.

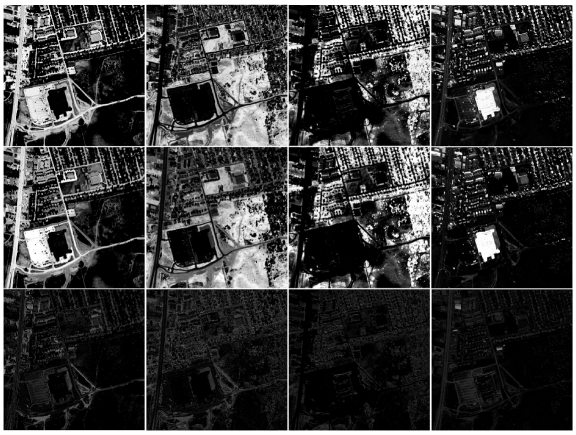

Figure 3.10: Abundance maps for Urban dataset. First, second and third rows indicate ground truth annotations, predictions and pixelwise differences for each material. Pure material signatures for this scene are given in 3.13.

3.4.4.2 Computational Complexity

The performance for methods is also evaluated in terms of computational and memory demands. As explained, the proposed method uses a stochastic-based optimizer that effectively scales these two terms and improves the applicability [101]. Furthermore, we empirically compare our method with baselines in terms of computation times in Tab. 3.3.

3.4.4.3 Tests on Datasets

Urban Data: As stated, we report both quantitative and qualitative results for Urban dataset in Tab. 3.4. From these results, we can undoubtedly indicate that our method achieves state-of-the-art performance for SAD and RMSE metrics. It approximately improves the baseline methods for 4.2% and 1.2%, respectively. Note that the second best method (i.e., DgS-NMF) leverages spatial priors in their estimation. This observation is critical since no additional knowledge is exploited in our method.

We also visualize the estimated abundance maps in Fig. 3.10. Absolute differences between the estimation and ground truth annotations are also computed to emphasize the errors on data. Observe that estimation errors are usually concentrated on the ob-

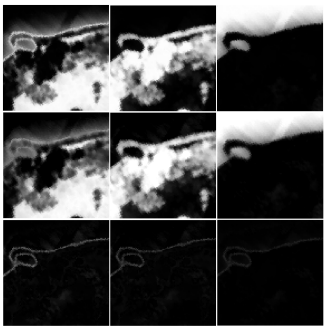

Figure 3.11: Abundance maps for Samson dataset. First, second and third rows indicate ground truth annotations, predictions and pixelwise differences for each material. Pure material signatures for this scene are given in 3.13.

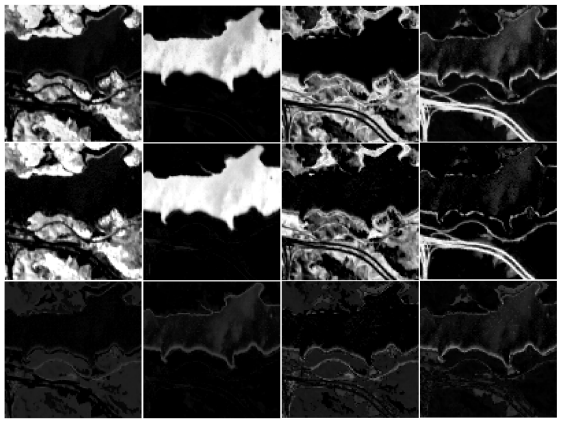

Figure 3.12: Abundance maps for Jasper dataset. First, second and third rows indicate ground truth annotations, predictions and pixelwise differences for each material. Pure material signatures for this scene are given in 3.13.

ject boundaries, where material mixtures increase. This result shows that our method is successful even for microscopic level mixture to some extent.

Samson Data: For Samson Data, the results are given in Tab. 3.5 and Fig. 3.11. Similarly, the proposed method outperforms all baselines by improving SAD and RMSE metrics approximately 2% and 3%. From the qualitative results, only the intersection of water-ground fields has absolute difference errors.

Jasper Data: For this data, we redefine the value of λ_2 as 0.4 since soil ($\#3$) and road ($\#4$) are highly mixture materials. Hence, the sparsity of material abundances is increased for this scene. Table 3.6 reports SAD and RMSE performance for all

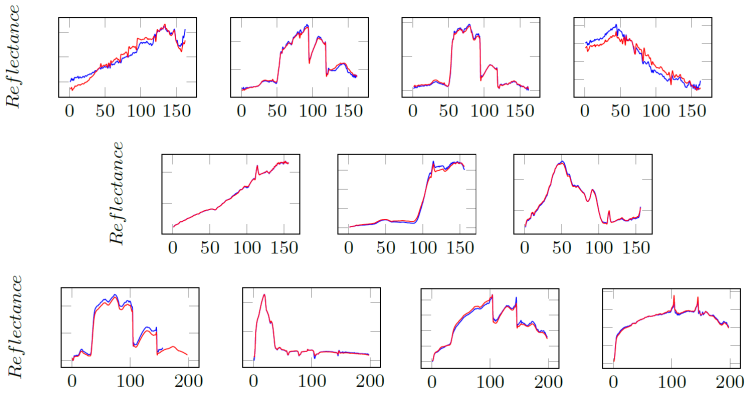

Figure 3.13: Ground truth (blue) and estimated (red) endmember signatures for Urban (first row), Samson (second row) and Jasper (third row) datasets.

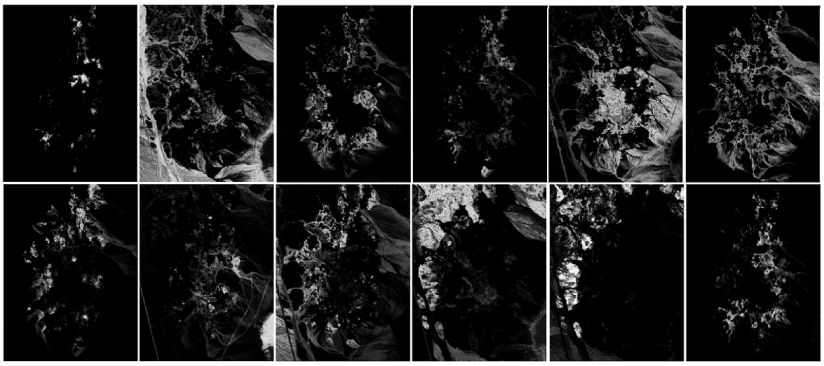

Figure 3.14: Abundance maps for Cuprite dataset.

methods. Even if our method obtains the best performance, improvements are not prominent compared to other datasets. Similarly, the errors are concentrated on the boundaries of different materials, as illustrated in Fig. 3.12. A critical observation is that our method with VCA initialization drastically drops down for this data. The reason is that DMaxD algorithm detects more uncorrelated spectral signatures than VCA. Hence, it maximizes the distance for estimated endmember signatures. Our method converges to more accurate solutions.

For Urban, Samson and Jasper dataset, ground truth and estimated endmember signatures are illustrated in Fig. 3.13.

Cuprite Data: Experimental results are reported in Tab. 3.7. Note that this data has only endmember signatures as ground-truth, and no abundance map value is available.

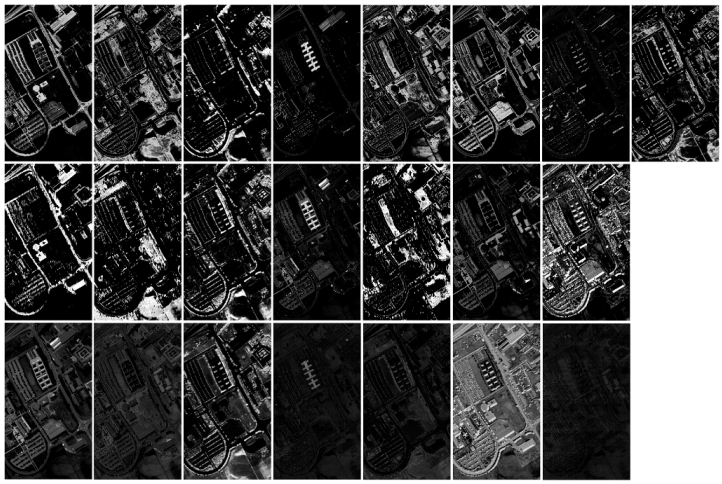

Figure 3.15: Abundance maps for Pavia dataset. The outputs of three methods are illustrated as Endnet, SCM and SPICE. As noticed, our methods computes 8 materials instead of 7 compared to the baselines. Details can be found in the text.

Hence, SAD performance is only provided. Since materials are highly correlated, we adjust the sparsity term λ_2 and p to 0.001 and 0.8, respectively. From these results, our method obtains the lowest SAD results on average. Fig. 3.14 visualizes the estimated abundance maps. We compare our results with [92] and validate that the results are consistent.

University of Pavia Data: This dataset (similar to Mississippi Gulfport data) does not have ground truth annotations for both endmember signatures and abundance maps. Hence, we compute an average spectra of pixels for each material. Later, we accept these averaged signatures as pure endmember signatures. Note that similar steps are done for this dataset in the literature [94].

Tab. 3.8 and Fig. 3.15 show the experimental results for this dataset. In addition, we redefine λ_2 and p to 0.01 and 0.8 respectively. Our method achieves the best scores for SAD results, and 3% improvement is introduced. Observe that the output of the three methods are illustrated as qualitative results. Moreover, our method computes 8 materials instead of 7. When we analyze the data in detail, we observe that there is a different material (i.e., grass) that is discarded. Hence, we compute an extra material for our method, Endnet. Note that the use of 8 material makes worse the performance for baseline methods.

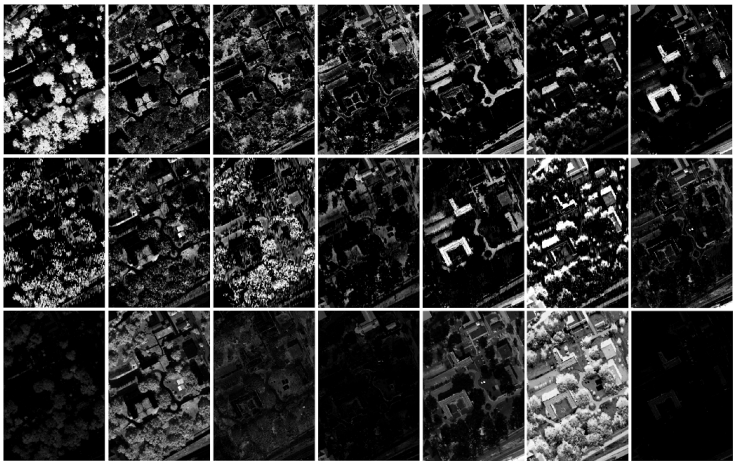

Figure 3.16: Abundance maps for Mississippi Gulfport dataset. The outputs of three methods are illustrated as Endnet, SCM and SPICE.

Mississippi Gulfport Data: For this dataset, λ_2 and p values are tuned as 0.1 and 0.8. In Table 3.9, our method yields the best overall SAD score. Similarly, the abundance maps of three methods are illustrated in Fig. 3.16. Estimation quality of our results is significantly better, especially for Tree ($\#1$), Shadow ($\#6$) and Building ($\#7$).

For the baselines, SCM cannot separate road and building materials since these material signatures are quite similar.

3.4.5 Deep Spectral Convolution Network for Multi-Spectral Unmixing (DSCN++)

3.4.5.1 Parameter Setting

Standard parameter settings are also preserved for the baseline methods. Furthermore, SCM configurations are transferred from Endnet. For endmember extraction step, we utilize three methods as DMaxD, SCM and Endnet. This feature shows that our method can be independently applied to other methods for spectral unmixing.

Furthermore, we provide quantitative performance for individual datasets. For clarity, we test each step individually and report the results for these steps in the experiments. Precisely, the performance is reported with or without each step, and their impacts are carefully analyzed. For instance, DSCN indicates the usage of a stack of convolutions

Table 3.4: SAD and RMSE results on Urban dataset. Mean and standard deviation are reported. Best results are illustrated in bold.

Endm.	Spectral Angle Distance (SAD) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	21.56 ±3.1	13.16 ±0.0	19.04 ±0.1	45.04 ± >9	9.38 ±0.0	6.06 ±0.2	**5.86 ± 0.1**	6.72 ±0.2	6.88 ±0.2
#2	39.04 ±5.1	>99 ±0.0	>99 ±1.3	>99 ±>9	10.55 ±0.0	20.05 ±0.4	13.69 ±0.4	4.04 ±0.1	**3.92 ±0.3**
#3	26.77 ±7.5	7.43 ±0.0	37.83 ±9.4	30.89 ±>9	14.03 ±0.0	3.71 ±0.0	4.12 ±0.0	3.68 ±0.2	**3.53 ±0.1**
#4	82.39 ±0.1	21.74 ±0.0	18.41 ±1.9	66.37 ±>9	41.86 ±0.0	14.22 ±0.2	10.54 ±0.3	3.99 ±0.7	**3.35 ±0.5**
Avg.	41.77 ±4.5	37.88 ±0.0	43.81 ±3.1	60.57 ±>9	18.79 ±0.0	11.01 ±0.2	8.55 ±0.2	4.60 ±0.3	**4.42 ±0.3**
	Root Mean Square Error (RMSE) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	31.31 ±7.2	27.08 ±0.0	27.69 ±0.3	29.43 ±3.4	32.79 ±0.0	14.77 ±0.1	13.18 ±0.1	**10.24 ±0.1**	10.41 ±0.2
#2	41.98 ±5.6	48.99 ±0.0	31.07 ±0.0	35.32 ±4.5	36.25 ±0.0	16.16 ±0.2	12.95 ±0.0	13.01 ±0.3	**12.24 ±0.3**
#3	27.93 ±3.7	28.07 ±0.0	27.27 ±0.4	26.33 ±1.9	32.61 ±0.0	12.65 ±0.2	9.57 ±0.1	8.69±0.3	**8.35 ±0.3**
#4	21.91 ±2.1	21.59 ±0.0	21.13 ±0.4	21.93 ±6.9	32.86 ±0.0	6.90 ±0.1	6.27 ±0.0	6.13 ±0.2	**5.92 ±0.1**
Avg.	30.79 ±4.7	31.43 ±0.0	26.79 ±0.4	28.25 ±4.2	33.59 ±0.0	12.62 ±0.1	10.49 ±0.1	9.51 ±0.2	**9.23 ±0.2**

Table 3.5: SAD and RMSE results on Samson dataset. Mean and standard deviation are reported. Best results are illustrated in bold.

Endm.	Spectral Angle Distance (SAD) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	22.33 ±2.7	4.04 ±0.0	4.19 ±0.0	45.37 ±>9	1.52 ±0.0	6.21 ±7.3	5.64 ±7.4	1.32 ±0.2	**1.29 ±0.1**
#2	4.90 ±0.2	**2.19 ±0.0**	5.61 ±0.0	12.19 ±>9	3.79 ±0.0	5.23 ±0.3	4.80 ±0.3	4.72 ±0.2	4.69 ±0.1
#3	12.29 ±0.0	13.04 ±0.0	18.15 ±0.0	9.24 ±>9	20.21 ±0.0	11.97 ±2.1	4.7 ±0.3	3.36 ±0.2	**2.95 ±0.3**
Avg.	13.17 ±1.0	6.42 ±0.0	9.31 ±0.0	22.26 ±>9	8.51 ±0.0	7.80 ±3.2	5.05 ±2.7	3.13 ±0.2	**2.98 ±0.2**
	Root Mean Square Error (RMSE) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	16.92 ±6.3	15.08 ±0.0	19.66 ±0.0	27.76 ±>9	18.65 ±0.0	8.58 ±3.3	7.77 ±3.8	5.86 ±0.0	**5.72 ±0.0**
#2	16.64 ±3.3	22.08 ±0.0	23.81 ±0.0	28.16 ±2.8	21.65 ±0.0	7.44 ±3.7	7.74 ±3.6	4.15 ±0.1	**3.84 ±0.1**
#3	25.42 ±0.6	26.41 ±0.0	31.78 ±0.0	35.57 ±3.4	34.61 ±0.0	5.55 ±0.9	2.70 ±0.9	**2.07 ±0.0**	2.11 ±0.0
Avg.	19.66 ±3.2	21.19 ±0.0	25.08 ±0.0	30.49 ±5.4	24.97 ±0.0	7.19 ±2.4	6.07 ±2.8	4.01 ±0.0	**3.88 ±0.0**

Table 3.6: SAD and RMSE results on Jasper Ridge dataset. Mean and standard deviation are reported. Best results are illustrated in bold.

Endm.	Spectral Angle Distance (SAD) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	24.34 $\pm$5.3	15.58 $\pm$0.0	18.41 $\pm$1.2	40.95 $\pm>$9	7.34 $\pm$0.0	15.10 $\pm$0.3	**4.66 $\pm$0.2**	8.23 $\pm$3.1	4.99 $\pm$0.4
#2	25.21 $\pm$0.4	25.39 $\pm$0.0	$>$99 $\pm$1.0	59.83 $\pm>$9	7.80 $\pm$0.0	4.60 $\pm$0.0	4.60 $\pm$0.0	**3.62 $\pm$0.4**	4.23 $\pm$0.9
#3	34.81 $\pm>$9	13.35 $\pm$0.0	26.27 $\pm$0.7	28.03 $\pm>$9	5.61 $\pm$0.0	6.16 $\pm$0.5	5.66 $\pm$0.2	6.23 $\pm$0.7	**4.47 $\pm$0.3**
#4	50.56 $\pm$9.3	10.69 $\pm$0.0	7.01 $\pm$0.8	86.19 $\pm>$9	3.36 $\pm$0.0	9.81 $\pm$0.1	6.73 $\pm$0.1	67.06 $\pm$0.5	**1.96 $\pm$0.2**
Avg.	33.73 $\pm$6.2	16.25 $\pm$0.0	37.92 $\pm$0.9	53.75 $\pm>$9	6.02 $\pm$0.0	7.19 $\pm$2.4	5.41 $\pm$0.1	21.29 $\pm$1.2	**3.91 $\pm$0.5**
	Root Mean Square Error (RMSE) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	11.99 $\pm$5.2	15.14 $\pm$0.0	20.80 $\pm$0.4	22.05 $\pm$6.9	10.31 $\pm$0.0	16.16 $\pm$0.5	11.66 $\pm$0.2	8.19 $\pm$2.6	**8.24 $\pm$0.4**
#2	11.59 $\pm$0.9	21.57 $\pm$0.0	24.03 $\pm$0.7	27.08 $\pm>$9	14.51 $\pm$0.0	5.57 $\pm$0.0	**4.13 $\pm$0.0**	26.18 $\pm$0.5	6.17 $\pm$0.3
#3	12.54 $\pm$6.9	13.89 $\pm$0.0	20.30 $\pm$1.8	21.35 $\pm$5.8	9.70 $\pm$0.0	17.02 $\pm$0.4	11.13 $\pm$0.3	19.91 $\pm$0.9	**8.98 $\pm$0.2**
#4	14.46 $\pm$3.2	15.76 $\pm$0.0	12.87 $\pm$0.2	27.44 $\pm>$9	7.87 $\pm$0.0	6.73 $\pm$0.2	**5.68 $\pm$0.1**	30.65 $\pm$0.3	8.55 $\pm$0.1
Avg.	12.65 $\pm$4.1	16.59 $\pm$0.0	19.50 $\pm$0.8	24.48 $\pm$8.1	10.60 $\pm$0.0	11.37 $\pm$0.2	8.15 $\pm$0.2	21.22 $\pm$1.1	**7.96 $\pm$0.3**

Table 3.7: SAD results on Cuprite dataset. Mean and standard deviation are reported. Best results are illustrated in bold.

Endm.	Spectral Angle Distance (SAD) ($\times 10^{-2}$)								
	VCA	DMaxD	MVSA	SPICE	SCM	$l_{1\|2}$-NMF	DgS-NMF	EndNet-VCA	EndNet-DMaxD
#1	12.89 ±5.3	**8.58 ±0.0**	23.60 ±5.7	14.71 ±1.6	24.43 ±4.6	16.22 ±2.0	12.48 ±1.8	13.60 ±2.9	14.18 ±2.5
#2	8.21 ±1.9	**7.15 ±0.0**	>99 ±>9	9.32 ±1.6	7.19 ±0.2	10.15 ±3.0	7.59 ±1.2	7.65 ±0.1	7.34 ±0.2
#3	**9.04 ±2.0**	11.58 ±0.0	>99 ±>9	10.26 ±0.3	13.25 ±0.3	12.50 ±7.5	10.81 ±3.2	11.44 ±0.8	11.66 ±1.1
#4	9.51 ±3.0	8.01 ±0.0	38.36 ±>9	11.91 ±1.3	12.22 ±0.9	13.07 ±5.3	11.01 ±2.1	8.38 ±0.6	**7.81 ±0.7**
#5	8.88 ±1.5	**8.47 ±0.0**	25.44 ±6.2	13.09 ±2.7	9.20 ±1.3	9.42 ±1.8	9.02 ±2.9	12.06 ±0.9	13.16 ±1.1
#6	7.08 ±5.2	8.52 ±0.0	>99 ±>9	9.60 ±0.5	6.80 ±0.4	9.82 ±2.7	7.42 ±1.2	6.05 ±0.3	**5.69 ±0.2**
#7	17.13 ±4.0	**10.26 ±0.0**	41.73 ±>9	9.97 ±0.5	17.15 ±2.5	29.86 ±7.4	20.51 ±6.0	15.93 ±1.9	15.45 ±1.1
#8	6.22 ±0.4	6.19 ±0.0	81.84 ±>9	8.60 ±0.5	6.28 ±0.2	10.27 ±4.7	7.27 ±1.2	5.51 ±0.3	**5.75 ±0.1**
#9	7.95 ±1.0	**7.36 ±0.0**	65.95 ±>9	11.45 ±1.6	7.61 ±0.1	12.80 ±4.1	8.88 ±1.7	8.36 ±0.7	7.78 ±0.1
#10	11.11 ±4.5	13.08 ±0.0	88.65 ±>9	6.99 ±1.9	5.89 ±0.1	8.12 ±1.9	8.82 ±5.0	**5.87 ±0.5**	6.12 ±0.4
#11	8.40 ±5.9	28.56 ±0.0	52.02 ±>9	8.99 ±2.4	**7.11 ±1.7**	11.03 ±3.3	8.15±2.1	9.03 ±1.4	8.38 ±1.2
#12	10.07 ±4.5	**8.23 ±0.0**	9.10 ±1.35	8.63 ±0.9	9.29 ±0.2	13.72 ±6.0	13.62 ±6.3	10.56 ±1.6	11.14 ±0.9
Avg.	9.71 ±2.9	10.54 ±0.0	71.92 ±>9	10.30 ±1.3	10.54 ±1.0	13.07 ±4.1	10.46 ±2.9	**9.53 ±1.0**	**9.54 ±0.8**

Table 3.8: SAD results on University of Pavia dataset. Mean and standard deviation are reported. Best results are illustrated in bold.

Endm.	Spectral Angle Distance (SAD) ($\times 10^{-2}$)					
	VCA	DMaxD	SPICE	SCM	EndNet-VCA	EndNet-DMaxD
#1	51.02 $\pm$8.9	7.50 $\pm$0.0	57.98 $\pm>$9	2.93 $\pm$0.0	**2.31 $\pm$1.1**	2.38 $\pm$0.6
#2	39.43 $\pm>$9	92.98 $\pm$0.0	66.33 $\pm>$9	**1.85 $\pm$0.0**	4.19 $\pm$1.1	6.07 $\pm$1.8
#3	28.85 $\pm>$9	$>$99 $\pm$0.0	12.50 $\pm$5.3	4.70 $\pm$0.0	**1.62 $\pm$0.1**	2.03 $\pm$0.8
#4	42.47 $\pm$4.6	30.60 $\pm$0.0	44.45 $\pm>$9	11.67 $\pm$0.0	10.11 $\pm$1.2	**8.57 $\pm$1.7**
#5	85.19 $\pm>$9	70.95 $\pm$0.0	$>$99 $\pm>$9	7.69 $\pm$0.0	**5.73 $\pm$0.6**	7.04 $\pm$0.8
#6	42.65 $\pm$8.8	56.93 $\pm$0.0	93.00 $\pm>$9	27.65 $\pm$0.0	4.62 $\pm$2.6	**1.59 $\pm$0.5**
#7	53.04 $\pm$3.7	53.06 $\pm$0.0	94.72 $\pm>$9	**5.29 $\pm$0.0**	55.80 $\pm$0.3	13.40 $\pm$1.9
Avg.	48.96 $\pm>$9	61.33 $\pm$0.0	67.79 $\pm>$9	8.82 $\pm$0.0	12.07 $\pm$1.0	**5.87 $\pm$1.2**

Table 3.9: SAD results on Mississippi Gulfport dataset. Mean and standard deviation are reported. Best results are illustrated in bold.

Endm.	Spectral Angle Distance (SAD) ($\times 10^{-2}$)					
	VCA	DMaxD	SPICE	SCM	EndNet-VCA	EndNet-DMaxD
#1	17.04 $\pm$>9	12.71 $\pm$0.0	>99 $\pm$>9	15.31 $\pm$0.0	3.47 $\pm$1.0	**2.75 $\pm$0.2**
#2	45.73 $\pm$>9	12.21 $\pm$0.0	>99 $\pm$>9	2.07 $\pm$0.0	6.32 $\pm$5.2	**1.28 $\pm$0.1**
#3	25.90 $\pm$1.8	26.12 $\pm$0.0	>99 $\pm$>9	**1.93 $\pm$0.0**	9.99 $\pm$8.4	3.97 $\pm$0.4
#4	42.09 $\pm$>9	34.12 $\pm$0.0	>99 $\pm$>9	**4.78 $\pm$0.0**	13.55 $\pm$>9	8.31 $\pm$0.7
#5	60.43 $\pm$9.7	24.13 $\pm$0.0	88.03 $\pm$>9	29.92 $\pm$0.0	23.48 $\pm$>9	**2.72 $\pm$0.7**
#6	39.10 $\pm$9.0	47.06 $\pm$0.0	>99 $\pm$>9	21.03 $\pm$0.0	13.69 $\pm$>9	**11.78 $\pm$1.0**
#7	37.42 $\pm$4.7	9.28 $\pm$0.0	46.60 $\pm$>9	**6.40 $\pm$0.0**	8.66 $\pm$2.5	11.56 $\pm$0.4
Avg.	38.25 $\pm$>9	23.67 $\pm$0.0	>99 $\pm$>9	11.65 $\pm$0.0	11.31 $\pm$7.9	**6.06 $\pm$0.5**

Table 3.10: RMSE performance on synthetic dataset for different configurations. Best results are indicated as Bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
$N = 4$	8.32 $\pm$1.4	4.47 $\pm$0.7	9.20 $\pm$1.4
$N = 8$	8.15 $\pm$1.4	4.36 $\pm$0.9	9.14 $\pm$1.0
$N = 16$	8.18 $\pm$1.0	3.97 $\pm$0.5	9.06 $\pm$1.1
$N = 24$	7.55 $\pm$1.1	4.42 $\pm$0.6	9.51 $\pm$1.2
w\o DSCN	10.78 $\pm$1.2	10.96 $\pm$1.5	13.37 $\pm$1.9
w\o EU	5.43 $\pm$0.3	3.81 $\pm$0.9	**6.43** $\pm$0.4
w\o WGAN	**5.23** $\pm$0.4	**3.22** $\pm$1.2	8.88 $\pm$0.3

instead of dense layers. EU also stands for the usage of refinement and uncertainty models with NN networks. Lastly, WGAN denotes the usage of Wasserstein GAN in the parameter optimization step.

Similarly, the computational complexity for our method is relatively small since the total number of learnable parameters is decreased compared to Endnet.

3.4.5.2 Test on Datasets

Synthetic Data: Even if synthetic data is conventional to evaluate the methods, it is not suitable to have all adverse cases exhibited on real data. As indicated, it is hard to simulate nonlinearity and spectral variability on data. For spectral variability, in particular, only additive noise is inserted that cannot reflect the actual physical conditions. Fig. 3.17 illustrates several samples formed from two pure materials. Observe that this leads to irrelevant signatures. For this purpose, we utilize Euclidean distance with SAD negative log-likelihood for parameter optimization. Note that we exploit Euclidean cost function only for synthetic dataset.

Experimental results are reported in Tab. 3.10. In particular, we first test all individual steps and parameter configurations. An important observation is that if we do not use EU and WGAN steps, the model achieves the best performance. This result makes

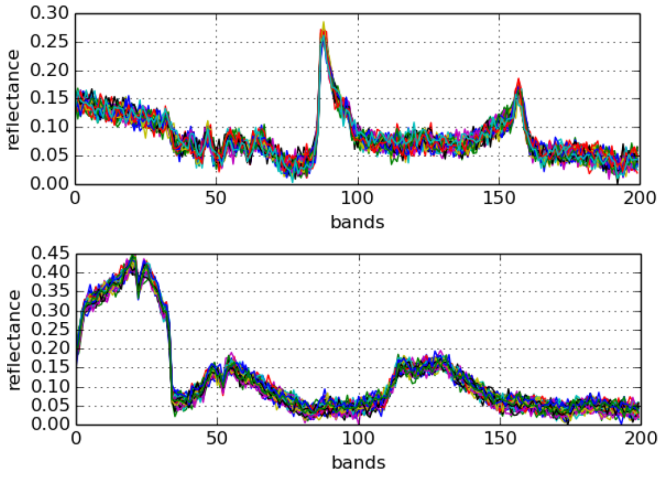

Figure 3.17: Spectra of pixels from the Synthetic dataset.

Table 3.11: RMSE performance on synthetic dataset for baselines. Best results are indicated as Bold-Red and second best results are shown as Bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
LMM	**5.17**	3.36	17.62
GBM	5.62	**3.26**	17.39
PPNM	7.66	4.01	9.34
MLM	6.52	3.85	**6.71**
NCM	**4.09**	3.85	11.17
DSCN++	5.23	**3.22**	**6.43**

Table 3.12: RMSE performance on Urban dataset for different configurations. Best results are indicated as Bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
$N = 4$	15.96 $\pm$2.6	13.26 $\pm$0.6	8.27 $\pm$0.4
$N = 8$	**15.47** $\pm$1.9	13.29 $\pm$0.8	8.05 $\pm$0.5
$N = 16$	16.32 $\pm$3.2	13.24 $\pm$0.5	8.16 $\pm$0.4
$N = 24$	16.56 $\pm$2.1	**13.23** $\pm$0.8	**8.04** $\pm$0.3
w\o DSCN	22.12 $\pm$2.6	16.63 $\pm$4.6	11.42 $\pm$0.8
w\o EU	34.14 $\pm$0.3	15.91 $\pm$0.3	8.38 $\pm$0.4
w\o WGAN	31.43 $\pm$1.6	15.76 $\pm$1.0	8.52 $\pm$0.5

sense since spectral variability is simulated as a random additive noise, and WGAN cost eventually learns this noise characteristics instead of spectral variability. Hence, this drastically reduces the performance and undermines the model. However, we will show the actual capacity of these steps on real data.

We also compare our method with baselines. In Tab. 3.11, these results show that the proposed method performs better even if the full capacity of model is not used.

Urban Data: To determine the best parameter configurations, Tab. 3.12 shows the experimental results. It can be seen from the results that N value (i.e., number of multinomial mixture components) must be larger compared to K value (i.e., number of endmember signatures). Even if this observation is made, there is no specific value for N for different endmember extraction methods. Observe that this value is influenced by the rate of mixture level of materials. Also, the capacity of endmember extraction methods affect this performance. All three steps that are proposed in our method ultimately improve the performance. For instance, significant drops are observed when DSCN is not removed. This result makes sense because high-dimensional representation ultimately leads to holes, as explained. WGAN also helps to converge more stable solutions. In particular, endmember extraction methods with low-capacity like DMaxD, this step becomes more critical to reach ideal solutions.

Table 3.13: RMSE performance on Urban dataset for baselines. Best results are indicated as Bold-Red and second best results are shown as Bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
LMM	31.04	28.00	27.59
GBM	31.04	27.00	27.08
PPNM	36.54	28.17	15.06
MLM	31.43	**16.34**	**9.48**
NCM	**31.04**	28.00	27.51
DSCN++	**15.47**	**13.23**	**8.04**

Similarly, we compare the performance of our method with the baselines. The results are shown in Tab. 3.13. As observed, performance for all three endmember extraction methods is significantly improved. In particular, EU and WGAN steps outperform all baselines, especially for DMaxD method, which ultimately eliminates to be stuck into local minima for learnable parameters. This step is also explained and suggests our previous claims.

Jasper Data: The performance of our method with different configurations is reported in Tab. 3.14. Observe that the results are almost similar to Urban data. The only difference is that performance is slightly better when DSCN is not used for SCM method. For baselines, our method obtains state-of-the-art performance, as shown in Tab. 3.15.

Note that we also compare our results with the estimates of Endnet. These results outperform Endnet on all datasets.

Table 3.14: RMSE performance on Jasper dataset for different configurations. Best results are indicated as Bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
$N = 4$	**12.03** $\pm$0.7	18.10 $\pm$0.9	6.38 $\pm$0.7
$N = 8$	12.29 $\pm$0.9	18.33 $\pm$1.1	6.61 $\pm$0.7
$N = 16$	12.12 $\pm$0.7	**17.92** $\pm$0.8	**6.31** $\pm$0.7
$N = 24$	12.27 $\pm$0.6	18.03 $\pm$1.1	6.62 $\pm$1.1
w\o DSCN	12.47 $\pm$0.6	17.49 $\pm$1.2	6.87 $\pm$2.2
w\o EU	14.78 $\pm$0.4	20.26 $\pm$0.4	8.36 $\pm$0.5
w\o WGAN	15.63 $\pm$0.3	19.03 $\pm$0.4	10.15 $\pm$0.5

Table 3.15: RMSE performance on the Jasper dataset for baselines. Best results are indicated as Bold-Red and second best results are shown as Bold.

$\times 10^{-2}$	DMaxD	SCM	EndNet
LMM	15.31	19.78	22.60
GBM	15.98	19.95	21.91
PPNM	**12.99**	**19.78**	14.72
MLM	16.59	20.73	**8.13**
NCM	15.29	19.56	22.42
DSCN++	**12.03**	**17.92**	**6.31**

CHAPTER 4

MULTIMODAL LEARNING

In this chapter, we give the details about the question: *How can we learn robust representations from multimodal data that can be insufficient or impractical when a single learning model is used?* For this purpose, biomedical, multispectral and visible data are used.

In general, the data that we use for knowledge extraction consists of different modalities with diverse data characteristics. In addition, this data is interpreted by multiple attributes to depict the latent representations accurately. As explained, these two problems are renowned as *multimodal learning* in the literature.

In particular, current NN models reach superior performance and outperform the human perception systems due to this deep structure. For these problems (i.e., multimodal learning), however, these methods can be insufficient to combine all the solutions as one model. Hence, separate NN models are required by many learnable parameters to capture all these variations. This weakness becomes essential, especially when we want to reach human-level intelligence.

Admittedly, using a single NN model than multiple NN models offers several functional advantages too. These benefits can be explained in twofold: First, separate models for multiple modalities reduce the usability of solutions. Specifically, there should be supervision to express the source indices to the models at every turn.

Second, it reduces the chance of parameter overfitting if model parameters are shared. Precisely, the convergence rate of parameters is increased for the domains, especially for insufficient labeled data.

In fact, multimodal learning problems have already been discussed in the literature, especially for specific fields. Hence, we address these issues and propose universal methods that can be used to learn robust representations once for all.

In particular, we first tackle multimodal semantic segmentation for biomedical data. As an initial step, we discuss why the current state-of-the-art methods collapse for this problem by supporting with theoretical and empirical explanations. Later, we make modifications in NN models to keep the covariate shift property. Furthermore, an additional cost function is used, which is based on adversarial model. In the experiments, we validate that sparse and rich representations can be obtained. The superiority of our method is that no label information related to data source is required at train and inference phases.

Second, we address an inverse problem by proposing a multimodal NN model to estimate the relative depth estimation from multispectral data. Furthermore, an additional task is solved with relative depth estimation (i.e., semantic segmentation) with shared parameters. Hence, we reduce the chance of parameter overfitting with better convergence. The main contribution is that multispectral data is used for the first time with multimodal learning, and the actual capacity of data is emphasized.

For our methods, extensive experiments are conducted for various datasets. Like the previous chapter, we will first describe the studies in the literature for multimodal learning domains. Later, the novel features of the proposed methods are explained in two separate sections. In the end, experimental results are reported with detailed discussions.

4.1 Literature Review

In the section, we will review the literature primarily for multimodal learning domains. Furthermore, since semantic segmentation is used in both problems, the literature review for this domain is also described.

4.1.1 Semantic Segmentation

Segmentation of visual data is one of the most difficult problems in computer vision and machine learning fields. The goal is to classify all individual pixels on visual data based on their semantic content.

Semi-automated methods come into prominence in earlier approaches due to their simplicity. The purpose is to use human operators and weakly trained learning-based models together to annotate data automatically. Human operators actively interact with data to fill in the missing semantic knowledge for the learning-based models in this process.

Generally, these semantic knowledge is computed from handcrafted features by taking advantage of color and geometric patterns [102]. However, the method performance depends heavily on the precision of humans (i.e., instant labels). As expected, these failures quickly lead to misinterpreted results. Furthermore, this is still time-consuming and reduces the diagnoses spends of patients, especially for biomedical field.

The initial version of fully-automated methods surpasses semi-automated approaches by a large margin in terms of accuracy and efficiency [103]. These methods require a large amount of labeled data to realize more comprehensive solutions. Handcrafted features are also used with robust classification techniques.

However, handcrafted features deteriorate the performance even if the total amount of labeled data increases. The main reason is that weak observations for data are obtained via human interpretations. As expected, the era of deep learning eliminates these drawbacks by collectively distilling patterns from data in an end-to-end manner [29, 104].

For this purpose, a large number of NN models have been proposed in the literature: FCN-8 [105], U-Net [29] and deepMedic [104]. These methods have their own novel NN structures that concentrate on different purposes. Specifically, one of which aims to boost the convergence rate for learnable parameters, while the other focuses on accurate performance. For instance, U-Net exploits skip-connections to propagate

the errors to the initial layers effectively. On the other hand, FCN-8 fuses latent representations in a single layer to increase the representation capacity. However, this step raises the computation load.

Additional improvements are also presented to these core structures. In particular, the concept of dilated convolutions [106] is employed instead of discrete convolutions. In this structure, the receptive field of kernels is adaptively enlarged. Hence, the need for hierarchical representations for the property of translation-invariant becomes unnecessary. Indeed, this method is exceptionally effective for semantic segmentation tasks, even if it also increases the hardware requirements.

Furthermore, post-processing steps are utilized in the recent models. Accurately, conditional random fields detect low-level spatial relations with neighboring features to straighten prediction results [107]. One of the drawbacks is that this model cannot be trained with base NN structures since it is not differentiable. For this purpose, adversarial model is incorporated to enhance the predictions at the boundaries of segmentations [108, 109]. Different cost functions, such as dice and focal cost functions, are also utilized [110, 111] for better segmentation results.

Last but not least, similar structures are broadly adopted for biomedical data with minor updates [112, 113, 114].

4.1.2 CrossModal Learning

An inherent feature of crossmodal data is that no additional knowledge is needed. Precisely, registration knowledge for different modalities or sources of data cannot be used as priors for complete solutions. Even if this research field has a very impactful direction, few studies are rarely discussed in the literature [115].

As expected, a simple solution for this problem is to use multiple dedicated models for individual modalities to compute robust representations [116]. In particular, the authors experiment with four dual-stream models (i.e., joint encoder-decoder parts, separated encoder part, separated decoder part and separated encoder-decoder parts). The outputs of this method prove that the model with separate encoder-decoder parts yields the best performance. This conclusion shows that the conventional structure is

not practical for unpaired crossmodal data.

Furthermore, clustering-based methods are used in which data is assorted hierarchically into several subspaces based on their geometric and appearance features [117]. Later, these subspaces are used to estimate segmentation results. Notice that this method offers robustness for crossmodal learning by keeping domain-specific statistics in clusters. However, clustering-step is not differentiable.

Recent studies show that normalization layers that are specialized for modalities boost the performance [118, 119]. Precisely, specific-normalization layers are utilized for individual modalities so that each layer has its own parameter sets. However, supervision is required for the selection of modalities, and this hinders the feasibility of solutions.

Similarly, domain adaptation methods are also used for this problem [120, 121, 122]. This idea is to compute cycle-transformation between modalities to adopt modality features to others. However, supervision is still essential for these methods.

4.1.3 MultiModal Learning

A critical issue for this problem is that multiple objective functions must be optimized concurrently to eliminate the bias towards tasks. The studies are roughly divided into two parts: One of which is that the models treat to the problem as multi-cost optimization where different tasks conflict with each other. Hence, the proper optimization step must be utilized to eliminate these cases. Second, the benefits of multimodal learning are emphasized on different tasks. Of course, more related tasks are selected for this purpose. Note that our method matches the second part since we aim to show that the real performance of multispectral data is valid for inverse problems when multiple-cost functions are optimized for better parameter convergence. In [123], the authors explain the importance of multimodal data fusion.

In [124], the authors exhaustively test all possible NN structure (i.e., from full-shared networks to independent-networks) for coherent two tasks. The outcomes show that soft-shared networks (i.e., a shared backbone with independent task-specific networks) yields the best performance.

In particular, different tasks need to be balanced for stable parameters that can work across all tasks. For this purpose, [125] adaptively modifies the weights of tasks with gradient norms. Furthermore, [126] defines the term of weight uncertainty to learn the best adaptive parameter scaling. Similarly, [127] uses a gradient-based model which is impractical for high-dimensional data so that an upper bound is defined for practical optimization step.

The authors [128] propose a pyramid model where additional tasks are at each pyramid scale. Furthermore, [129] incrementally transfers knowledge between tasks for all layers. However, a large number of parameters are updated, which reduces the feasibility of the models. Moreover, an attention model is utilized to compute task-specific representations from a single global representation [130]. Indeed, the studies show that the use of multimodal learning eventually improves performance for almost all applications.

4.2 MultiModal Learning For Biomedical Data

In recent decades, learning-based solutions have been used to replace human interventions. In particular, these methods have become critical for biomedical data to reduce the time for diagnoses and the misinterpretation rates due to human factors.

These methods are used in a variety of biomedical applications. However, semantic segmentation plays a vital role among these applications since clinicians often exploit these methods to automatically estimate the segments of body organs. However, current segmentation algorithms can operate on a single data modality even if clinicians prefer to interpret multimodal data for better diagnosis.

The main reason for this case is that learning methods (i.e., deep neural networks in particular) are sensitive to diverse visual changes [131]. Hence, there is a drawback to the convergence of ideal solutions, and performance for multimodal data is expected to decrease.

In this thesis, our goal is to semantically segment biomedical data from different modalities (i.e., CT or MR data) by utilizing a single NN model. Note that the types of

modalities are not provided at both training and inference phases. Hence, the model offers a universal solution that works on all modalities. Ultimately, this improves the usability of the solutions. Besides a practical solution, this learning scheme has important advantages, especially for the convergence of parameters. Indeed, complete models must maintain shared and distinct features related data to achieve state-of-the-art performance.

For this purpose, we use a modified deep convolution network (DCN), which has the configuration of hard-parameter sharing for all modalities. The first contribution is that we analyze the conventional DCN methods by theoretical and experimental considerations and recognize the weakness of these methods. Later, we reorder the normalization and convolution layers to maintain covariate shift property, which is underestimated in the conventional techniques. Even if this modification vaguely retains this property, it unexpectedly result in poor performance. Hence, we add a learning term to the objective function with adversarial model. This term regularizes the NN model to cluster the input data based on their modal-based similarities. After all, the selectivity of representation is truly increased. Our method ensures that learnable kernels receive more reliable statistical knowledge from all modalities. In our model, modality information of input data is not provided as prior knowledge to the model. This feature is essential since our method achieves state-of-the-art performance even if training and inference procedures are still the same as the unimodal model.

4.2.1 Proposed Method

As explained, our primary goal is to deploy a novel method that can be trained with all available data from multiple modalities. No prior information is provided for data to the model in both train and inference phases. It the end, a single parameter set is optimized.

If we assume that input data $\mathbf{x}^c \in X^c$ per input modality space is transformed to a multi-class label space Y such that there is a single hypothesis class that is applicable for all modalities as $f(\mathbf{x}^c; \theta) : X^c \rightarrow Y$. Here, c denotes the data modalities. Observe that a shared parameter set θ is optimized. As previously stated, our task for

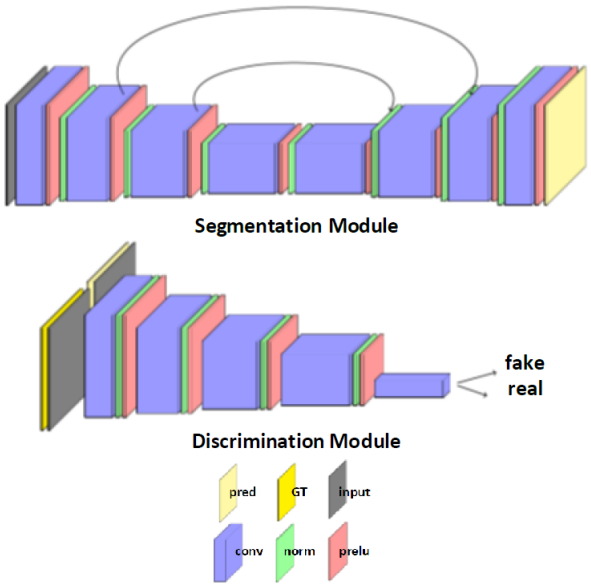

Figure 4.1: Overall flow for multimodal data for biomedical data.

multimodal data is semantic segmentation, so that softmax negative log-likelihood cost function (i.e., Eq. 2.5) is utilized for parameter updates.

After all, the model strives to reach an equilibrium for multimodal data with the same class id t as:

$$f(\mathbf{x}^0;\theta) = f(\mathbf{x}^1;\theta) = ... = f(\mathbf{x}^C;\theta) = y^t \tag{4.1}$$

For this purpose, we utilize a DCN model that reserves many learnable parameters. The structure of the model is similar to autoencoder, where it consists of encoder-network parts. For instance, encoder part takes input data and computes latent representations step-by-step by arranging the dimensions and representation capacity simultaneously. In other words, the spatial resolution of data is compressed while representation capacity is increased in deep hierarchies. Later, these latent representations are aggregated to produce pixel-level classification results. Skip-connection is exploited in this structure to enhance the flow from encoder layers to decoder layers. The overall flow is illustrated in Fig. 4.2.

However, this conventional method is insufficient to achieve superior performance for multimodal data. Hence, we first redefine the DCN structure. Later, an additional cost function is introduced to have domain-specific statistics without additional parameters

or supervised annotations.

4.2.2 DCN Structure

As noticed, the DCN structures extensively use convolution layers to compute local representations from data. These layers are supported with additional layers to boost the performance in terms of accuracy and efficiency. Strictly speaking, normalization and activation layers are deployed at the convolution operation outputs for nearly all applications. Ultimately, this configuration regularizes the convolution kernels to decrease the covariate shift due to diverse data distributions. Of course, sparsity and nonlinearity are also maintained in the solutions.

However, the objective of normalization layers (i.e., elimination of covariate shift) degrades the multimodal data by only allowing common distributions at the end. This structure concludes that even if modality of data changes, response distributions are still the same or similar. Hence, this layer solely emphasizes the common patterns exhibited from data. Ultimately, this idea is harmful for multimodal data because distinct patterns must also be retained with common ones for robust representations.

First, we mathematically analyze that how the normalization layers eliminate the covariate shift from models. At a single layer, convolution operation and batch normalization can be simplified with $f_{conv}(\mathbf{z}) = \mathbf{W}\mathbf{z}$ and $f_{bn}(\mathbf{z}) = \dfrac{\mathbf{z} - \mu(\mathbf{z})}{\sigma(\mathbf{z})}$. Here, $\mu(.)$ and $\sigma(.)$ correspond to batch mean and batch variance of $\mathbf{z}$, respectively. When these layers are consecutively applied as follows:

$$f_{post}(\mathbf{z}) = \frac{\mathbf{W}\mathbf{z} - \mu(\mathbf{W}\mathbf{z})}{\sigma(\mathbf{W}\mathbf{z})} \tag{4.2}$$

Observe that the mean and variance operations are $\mu(\mathbf{W}\mathbf{z}) = E[\mathbf{W}\mathbf{z}]$ and $\sigma(\mathbf{W}\mathbf{z}) = E[(\mathbf{W}\mathbf{z} - E[\mathbf{W}\mathbf{z}])^2$. As noticed, the statistics depend profoundly on learnable parameters $\mathbf{W}$ since it is applied before the normalization layer. Input data is transformed into a different feature space (i.e., distribution) that holds solely common representations for all modalities. Eventually, this structure induces a drawback for multimodal learning.

For this purpose, we reorder the convolution and normalization layers. This mod-

Table 4.1: Percentage of responses forwarding to next layers.

Methods	Modality of Data	
	CT	MR
Pre-BN-DCN w/o CAN	0.596	0.627
Post-BN-DCN w/o CAN	0.584	0.592
Post-BN-DCN	0.491	0.480
Pre-BN-DCN (ours)	0.277	0.411

ification undermines the effect of convolution layers towards common and distinct features can be computed from data. Hence, the formulation is updated as follows:

$$f_{pre}(\mathbf{z}) = \frac{\mathbf{W}\mathbf{z} - \mathbf{W}\mu(\mathbf{z})}{\mathbf{W}\sigma(\mathbf{z})} \tag{4.3}$$

Observe that mean and variance operations $\mu(\mathbf{z}) = E[\mathbf{z}]$ and $\sigma(\mathbf{z}) = E[(\mathbf{z} - E[\mathbf{z}])^2$ depend only on input data.

A concern may arise from the readers that when we use these operations in multiple layers, can it be equal to the same output characteristics as in the post-structure (i.e., $f_{post}(.)$). Note that a nonlinear activation function is utilized after these operations, so linear relations disappear. In other words, this concern is eliminated with the sparsity property of activation functions.

However, this modification unexpectedly yields poor performance. The main reason is that we still fit a unimodal distribution for each normalization layer. Even if this structure takes more statistical knowledge of modalities into account, the outputs are still underestimated.

Hence, we exploit a different cost term in the objective function, which intuitively enhances the sparsity and modality knowledge in cluster-based methods. The details of this observation will be explained in the next subsection.

4.2.3 Parameter Learning

Even if the NN structure modification allows more statistical knowledge for multimodal data, the distributions are still unimodal and degrade the sparsity for the

responses. To empirically validate this observation, we conduct an additional experiment, and the outcomes are reported in Tab. 4.1. The results basically show the portion of responses that are active in the next layers for different configurations. Observe that the pre-structure decreases the sparsity of responses compared to post-structure, which is dangerous. The reason for this is that it harms the combination of normalization and activation operations, as explained in the sparse autoencoder chapter.

For this purpose, we introduce an adversarial cost function. As a short reminder, the initial applications of adversarial models for semantic segmentation are to boost NN outputs, especially for the boundaries of segments, as we reviewed. Here, however, we use this algorithm to cluster the latent space and group the knowledge from individual modalities. Ultimately, this regularizes the representations and pushes them to multi-modal cases than unimodal ones. Similarly, the sparsity experiments are repeated for this final model as well. In Tab. 4.1, pre-structure with adversarial cost yields the best sparsity among others. The reason is that the use of both covariate shift property and adversarial model improve the sparsity because more statistical knowledge is used in the models at the end.

Formally, true annotations and model predictions (i.e., all pixel-wise) are combined with input data before they feed them to the discriminative model. Input data $\mathbf{x}$ and true annotation $\mathbf{y}$ are concatenated by channel-wise as $\mathbf{v}$. In addition, input data $\mathbf{x}$ and model prediction $\hat{\mathbf{y}}$ are conditioned $\hat{\mathbf{v}}$ in the same manner. Note that no modality information is used in this step. After all, the discriminative model distinguishes these representations by solving a Wasserstein loss:

$$\mathcal{L}_{disc} = \mathbb{E}[d(\mathbf{v};\theta^d)] - \mathbb{E}[d(\hat{\mathbf{v}};\theta^d)] + \lambda_{pq}\mathbb{E}[(\|\nabla_{\tilde{\mathbf{v}}} d(\tilde{\mathbf{v}};\theta^d)\|_2 - 1)^2]. \tag{4.4}$$

$$\mathcal{L}_{gen} = E[d(\hat{\mathbf{v}};\theta^d)] \tag{4.5}$$

Notice that it is still a 1-Lipschitz function. Furthermore, $\tilde{\mathbf{x}}$ is uniformly sampled data from real $\mathbf{v}$ and prediction $\hat{\mathbf{v}}$ data. λ_{pq} is set to 10.

For this purpose, visual patterns that belong to the same classes are forced to have similar representations. In addition, the generalization capacity of our model is improved,

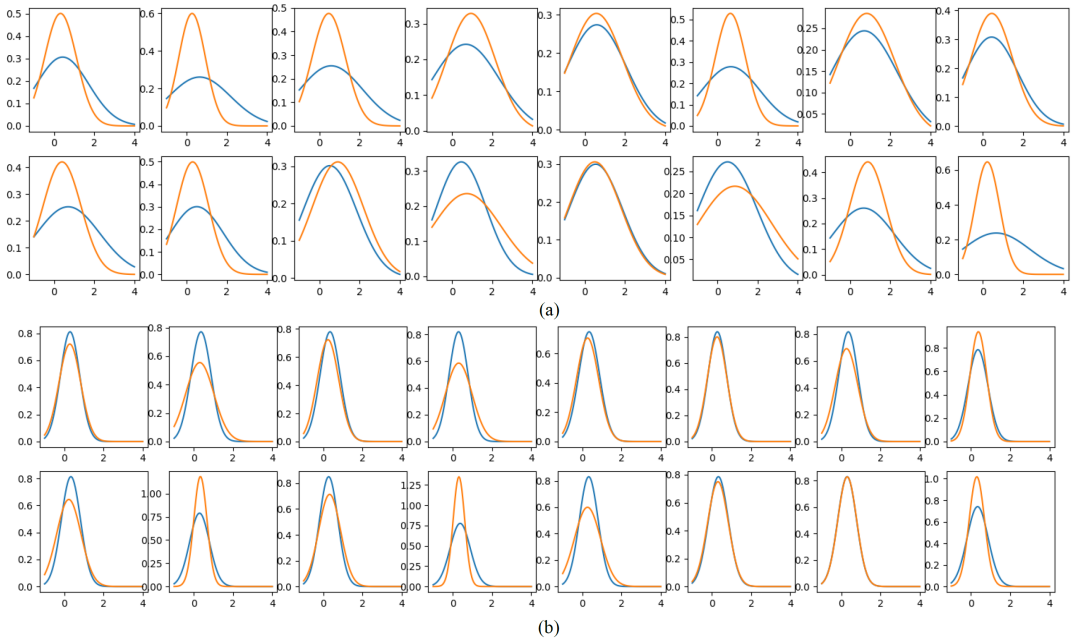

Figure 4.2: Response distributions of CT (blue) and MR (yellow) data for pre-structure (a) and post-structure.

as explained in [132]. The overall cost function L corresponds to the combination of L_{ce} and L_{gen} by rescaling with coefficients as follows:

$$\mathcal{L} = \mathcal{L}_{ce} + 0.001 \times \mathcal{L}_{gen} \tag{4.6}$$

Adam optimizer is used to train learnable parameters in the model. Learning rate and number of iterations are set to 0.001 and $100K$, respectively. Parametric ReLU is selected for activation function.

To show the superiority of our method, we visualize the response distributions for CT and MR data sampled from various NN layers. In Fig. 4.2, almost identical distribution characteristics are observed for post-structure. This result proves that the post-structure solely maintains common patterns exhibited from data. On the other hand, for the proposed structure (i.e., pre-structure), CT and MR data behave differently while identical distributions are also retained.

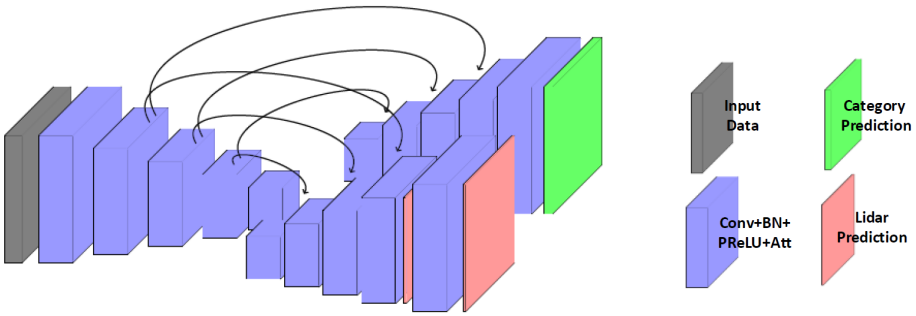

Figure 4.3: Overall flow for multimodal data.

4.2.4 Data Preparation and Post-Processing

Until now, input data $\mathbf{x}$ is accepted as it corresponds to a slice of CT or MR data. However, our method can also be adopted to estimate semantic segmentation results from multi-sliced data. Hence, sequential knowledge about data can be exploited in this manner.

Formally, multi-sliced data corresponds to channel-wise concatenation of T next and T previous adjacent slices, and the final channel size is set to $2 \times T + 1$. Here, T is a hyper-parameter, and we experimentally determine the practical value for multimodal data (i.e., CT and MR data). The main issue is that since multimodal data is unpaired, increasing slice numbers can adversely affect the performance.

Moreover, we utilize a pre-processing step that rescales the contrast of individual pixels. Note that contrast changes due to variations in sessions and uncalibrated imaging devices are typical for these modalities.

4.3 Multi-Modal Learning For MultiSpectral Data

As a short review, a multispectral sensor collects measurements from a wide range of electromagnetic spectrum, and richer content knowledge can be extracted from the data compared to the visible range sensors. However, several theoretical and practical limitations impede the applicability of the sensors. One of these is the acquisition of data is expensive, so that the scarcity of data is inevitable. Furthermore, the measurements are sensitive to noise, spectral variability and low-spatial resolution. Lastly, the

dimensionality of data can be problematic, especially for modern machine learning techniques.

All of these concern is multiplied when latent representations are computed with regression-based models. For regression problems, in particular, a continuous space needs to be optimized with learnable parameters, which has infinitely many possible solutions at the end. Hence, these problems surely consume a large amount of data for stable solutions. This requirement grows when the dimension of input data is increased.

In this sense, multimodal learning provides some practical benefits over single-task optimization by solving multiple cost functions to obtain more stable representations.

For this purpose, we combine two well-studied tasks for multispectral data, as relative depth estimation and semantic segmentation, to alleviate the necessity of large-scale data for the regression task. As explained, this domain is well-studied. The main contribution of our method is to show that multispectral data can be more reliable for inverse problems since rich content knowledge is easily extracted from data. As mentioned, however, these limitations do not permit to indicate the actual performance of this sensor. Hence, we solve multi-cost functions, as described in which state-of-the-art performance is clearly obtained for multispectral data.

4.3.1 Proposed Method

Formally, input data $\mathbf{x}$ is converted to pixel-wise depth estimation $\mathbf{y}^s$ and semantic segmentation $\mathbf{y}^d$ by using a NN method. As indicated, the NN structure has two parts. First, the input data is passed through encoder part $\mathbf{z} = h(\mathbf{x}; \theta)$ to compute common latent representations $\mathbf{z}$. Note that this part has shared parameters and is generally known as backbone. Later, these representations are fed to two task-specific NN models $\mathbf{y}^d = g(z; \theta^d)$ and $\mathbf{y}^s = g(z; \theta^s)$ with separate parameter sets as θ^d and θ^s. Similarly, skip-connections are utilized to establish direct connections from encoder part to task-specific NN models. The overall flow is illustrated in Fig. 4.3.

Several improvements are also proposed to this NN structure. First, we use the concept of Laplacian pyramid. Specifically, the task-specific model for depth estimation

outputs the predictions at two different scales that doubles the resolution. Hence, the error is propagated from multiple outputs instead of the top layer only. Furthermore, these predictions are conditioned to the next layers at each scale, and incremental representations can be learned from data. Also, attention model is utilized for the backbone of our NN model. Note that a similar notion is later used in [130]. Spatial mean is exploited in the attention models.

4.3.2 Parameter Learning

We use three individual cost terms for optimization: Euclidean cost function to minimize the difference between the estimated and ground-truth for depth estimation, softmax log-likelihood for semantic segmentation and an l1 regularization term as follows:

$$\mathcal{L} = \sum_{c=1}^{C} \mathcal{L}_{mse}^{c} + 0.1 \times \mathcal{L}_{ce} + 0.01 \times ||\mathbf{y}^{d}||_{1} \tag{4.7}$$

Here, C indicates the scale number of Laplacian pyramid. As noticed, the influence of softmax log-likelihood is scaled by 0.1. As noticed, an accurate depth estimation is our primary motivation for parameter updates. Hence, this cost function with a particular configuration is used to refine the shared parameters for regression task. Furthermore, l1 regularization is applied to the outputs of depth estimation model. As noted, we use lidar data as ground-truth annotations. These annotations are sparse (i.e., the most values of pixels are zero). Hence, it ultimately penalizes the pixels that are constantly active. Intuitively, this term can be accepted as a balancing term in imbalance class problem.

4.4 Experiments

In this section, we will first describe the datasets that we utilized to test multimodal problems. Moreover, evaluation metrics are explained before giving the experimental details. Later, we will present the experimental results and compare the performance of our methods with baselines.

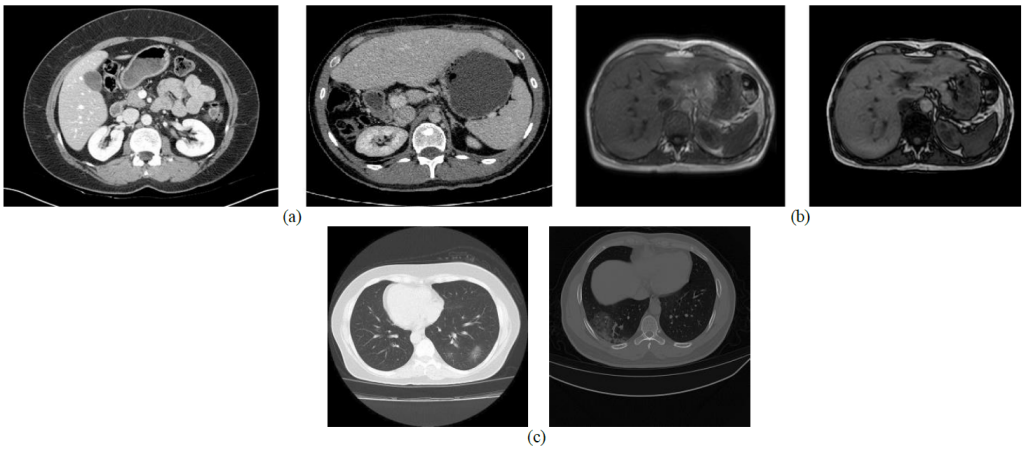

Figure 4.4: Visual data for Liver and Covid-19 datasets. (a) Liver CT,(b) Liver MR (c) Covid-19 CT

4.4.1 Biomedical Data

As explained, we use biomedical data to evaluate the performance of our method for multimodal problem. For this purpose, we decide to use liver and Covid-19 datasets. Covid-19 data consists of 20 labeled CT lung scans, and radiologists annotate it as Lung and Infected parts. 5 of these scans are reserved for the test phase, and the rest is used at the train time. Some visual samples for this dataset are illustrated in Fig. 4.4. Observe that this data has severe visual variations due to scanner types. Hence, this introduces useful data to validate our method for multimodal problem.

The second dataset is CT and MR liver data that is unpaired as expected. Both data types contain upper abdomen image series of 20 patients (40 image sequences in total). Similarly, 8 of these scans are utilized in the test, while 12 are used for the training step. A critical characteristic of data is that all patients are potential donors, so no focal lesions (i.e., tumors) are labeled. For CT and MR data, it is normalized as the patient orientation. Some visual samples for this dataset are shown in Fig. 4.4. Furthermore, some samples and ground truth annotations from different slices 4.5.

Note that CT and MR data have diverse characteristics since two separate working principles have for two of these modalities. As expected, these diverse characteristics are ultimately eliminated by conventional methods so that data-specific characteristics are discarded. Indeed, each offers its own advantages and disadvantages that make

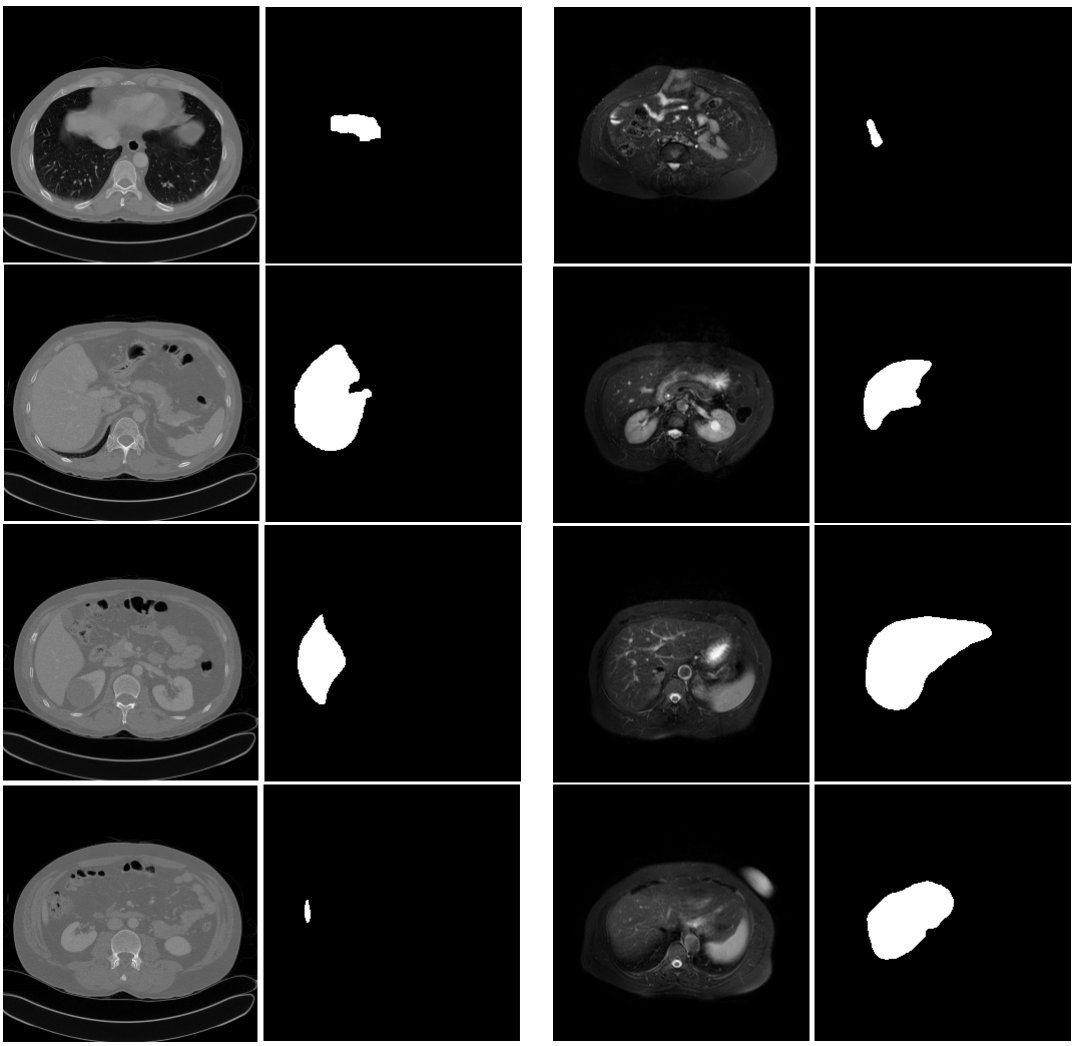

Figure 4.5: Samples and ground truth annotations for CT and MR data from different slices.

Figure 4.6: Visual data for MNIST and SVHN datasets.

patient diagnosis easier or more difficult. For instance, a single CT scan is typically recorded less than a minute. However, MR data takes up to 20 minutes. The acquisition time significantly affects the performance of these modalities. CT data requires additional contrast-enhancement for soft tissues. Also, MR data is sensitive to patient movements, and fluctuations occur for the same sequence. Indeed, operational costs for MR data are substantial as well. Furthermore, inter-slice distance is different for CT and MR data. Precisely, MR data has a greater inter-slice distance, so sudden changes are observed.

4.4.2 Visible-Range Data

We also test our method on a visual classification task with the same configuration in biomedical data. Hence, we use Mnist and SVHN datasets composed of digit numbers (i.e., 10 classes). Some visual samples for these datasets are given in Fig. 4.6. Moreover, we significantly reduce the number of samples to understand the effect of multimodal data more effectively. Hence, the first 10K data is used from each dataset. Similar to biomedical data, all learnable parameters are shared.

4.4.3 Multispectral Data

For multimodal data, we use 2018 IEEE GRSS Data Fusion Challenge Dataset [133]. This data is composed of multi-sensor data as multispectral, lidar and RGB types. In

particular, multispectral data covers the spectral range from 0.38μm to 1.05μm with a spectral dimension of 48. The ground resolution is 1m for multispectral data. On the other hand, the ground resolutions for lidar and RGB are better, and they are 50cm and 2cm, respectively. Hence, lidar and RGB data are scaled to the resolution of multispectral data in our experiments. For the sake of simplicity, the data is split into 14 non-overlapping parts. The resolution of each part is 512×512. For this data, data augmentation is generously exploited at train time.

4.4.4 Evaluation Metric

For biomedical data, we utilize various metrics to compare 2d and 3d objects in terms of visual similarities. Five score metrics are calculated for CT and MR data as follows. Note that these metrics have frequently been used for the evaluation of methods in biomedical segmentation competitions [134]:

- Volumetric Overlap (VO)
- Relative Volume Difference (RVD)
- Average Symmetric Surface Distance (ASSD)
- RMS Symmetric Surface Distance (RMSD)
- Maximum Symmetric Surface Distance (MSSD)

For Covid-19 data, Pixel Precision (PR) and Volumetric overlap (VO) are utilized. These metrics will be briefly summarized in this section. First, Root MeanSquare Error (RMSE) metric is used for multimodal data:

$$\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{K}\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}. \tag{4.8}$$

Volumetric Overlap (VO): It merely measures the portion of the total number of pixels that interacts with ground truth segmentation. Later, this metric is scaled with 100. In the worse case, it produces zero if there is no overlap:

$$\text{VO} = \frac{V_{seg} \cap V_{ref}}{V_{seg} \cup V_{ref}} \times 100 \tag{4.9}$$

Relative Volume Difference (RVD): Likewise, this metric measures the absolute difference between ground truth and predicted results. When accurate segmentation is performed, the score value is zero. Hence, the range of metrics is converted by using the inverse ratio of this metric:

$$\text{RVD} = \left| \frac{V_{seg} - V_{ref}}{V_{ref}} \right| \times 100 \tag{4.10}$$

Average Symmetric Surface Distance (ASSD): This metric is used to measure the similarities of two objects in terms of their shapes. Precisely, two 3d objects are aligned so that their centers are overlapped. Later, euclidean distance to boundaries is calculated. The lowest distance is accepted as a symmetric surface distance for each pixel. Later, the mean of symmetric surface distances is calculated. This value gives zero if they are the same object.

RMS Symmetric Surface Distance (RMSD): Similarly, Root Mean Square (RMS) of symmetric surface distances is calculated for two objects.

Maximum symmetric surface distance (MSSD): This metric represents the largest difference between two objects. This metric is critical since it is useful for the evaluation applications of surgical precision.

4.4.5 MultiModal For Biomedical Data

In our experiments, we compare the performance of our method with several baseline algorithms in the literature. Precisely, two DCN structures are selected as Cond-DCN [116] and Att-DCN [135]. Cond-DCN uses an additional NN model classifier that predicts the modality id of data. Then, these classification results are conditioned to the semantic segmentation model. The drawback of this method is that the modality id should be provided at train time, which is not practical for complete solutions.

Furthermore, Att-DCN adopts the concept of attention model for the skip-connection layers. Hence, a subset of responses are used for semantic segmentation task.

Test on Liver Data: Experimental results are shown in Tab. 4.5. In addition, the last column is reserved for the average scores of CT and MR performance for all metrics. Here, T (i.e., slice-number) is set to 1. The results show that our method outperforms

Table 4.2: Performance for individual modalities and a multimodal method.

Methods	Modality of Data	
	CT	MR
DCN for CT	75.81	-
DCN for MR	-	79.47
Pre-BN-DCN (ours)	**84.20**	**82.81**

all baselines with a large margin. In particular, the performance of Cond-DCN is limited, that uses an additional NN model to predict the modality of data. One of the main reasons is that this structure needs to learn additional parameters. Due to the limited amount of data, this inevitably leads to parameter overfitting.

When we compare the proposed structure (pre-structure) with the conventional one (post-structure), we find that it performs slightly worse if no adversarial model is used. Precisely, lack of sparsity is one of the main reasons. For this purpose, we incorporate adversarial model that ultimately regularizes the overall method by clustering latent representations based on their modalities. Note that no information for data modality is used at this time. In particular, the improvement for pre-structure is better than post-structure. Using data-specific statistics has a positive impact on performance at the end.

Furthermore, we analyze the effect of normalization layers in detail. For this purpose, we use two types of normalization models: batch normalization (BN-DCN) or instance normalization (IN-DCN). Note that there is no critical difference in performance for either model. This result shows that our solution does not overfit to a particular model.

In fact, the most critical observation is *Does multimodal learning improve the performance compared to individual models*? To answer this question, we conduct a further experiment by using separately trained models for both modalities. In Tab. 4.2, our multimodal method significantly improves the performance compared to these baselines. As expected, the use of all available data regardless of their modalities ultimately leads to more stable and robust representations.

As an ablation study, we also test the impact of slice-number to the performance. The

Table 4.3: Precision and IOU results on Covid-19 dataset for multimodal segmentation.

Methods	Parts (PR / IoU)		
	Lung	Covid	BG
Post-BN-DCN	**96.47** / 38.89	30.38 / 17.91	89.74 / -
Pre-BN-DCN (ours)	95.48 / **41.13**	**40.39** / **22.16**	**92.84** / -

results are shown in Tab. 4.6. The last column gives the mean scores of CT and MR performance for all metrics. From these results, $T = 0$ for batch normalization and $T = 1$ for instance normalization achieve the best performance. As indicated, MR data is sensitive to noise due to the acquisition time. Hence, instance normalization is more appropriate for this type of normalization (i.e., we have already discussed that batch normalization introduces errors for each iteration). On the other hand, for CT data, batch normalization is superior. Furthermore, the use of multi-slice data as input harms the performance. The main reason is that inter-slice distance is different for CT and MR data.

Test on Covid-19 Data: In our experiments, we only report the performance of post and pre-structures for this dataset. The results are illustrated in Tab. 4.3. Observe that pre-structure predominantly achieves the best performance for all lung parts. Even if the data consists of CT type, there is a high variation on data because different sensor models are used to collect data. In addition, the rates of false predictions are high for post-structure. Ultimately, our method (i.e., pre-structure and adversarial model) improve the robustness of latent representations.

4.4.6 MultiModal For Visible-Range Data

Experimental results for visible-range data are illustrated in Tab. 4.4. From these results, the same observations can be made as in biomedical data. Note that, single and dual notations indicate that a single model is used for each dataset or all datasets are used with a single model. Precisely, the use of adversarial loss with normalization structure yields compatible performance compared to individual models. Note that the number of parameters for multimodal data is reduced as well by two in the end.

Table 4.4: Precision results for Mnist and SVHN datasets.

	Pre-DCN-Single	Pre-DCN-Dual	Pre-DCN-Dual-CAN
Mnist	99.24	99.06	99.22
SVHN	90.16	89.93	90.73
	Post-DCN-Single	Post-DCN-Dual	Post-DCN-Dual-CAN
Mnist	99.16	99.03	99.18
SVHN	90.36	90.23	90.57

4.4.7 Multi-Modal For MultiSpectral Data

In our method, the main goal is to validate that the use of multispectral data for inverse problems is currently impractical because the limited data size aggravates to learn optimal parameters from a continuous space. Since this is a critical drawback, multispectral data can provide richer knowledge about scenes compared to RGB data. For this purpose, we produce similar experimental configurations for all configurations to understand the impacts of RGB and multispectral data to depth estimation.

First, lidar depth estimation is computed for three data models: RGB, multispectral and multispectral-semantic. Here, multispectral-semantic indicates that multimodal cost functions are optimized together for both inverse and semantic segmentation tasks. Furthermore, the impact of Laplacian layer and attention function is also evaluated for these models. Experimental results are reported in Tab. 4.7. From the results, the use of multispectral data performs slightly better results compared to RGB data. Ultimately, multispectral data has more meaningful clues about scenes in terms of spectral details. Furthermore, the use of multimodal cost functions improves the performance compared to other data models. As discussed, noise-prone characteristics and difficulties to access a large amount of data adversely affect the performance of multispectral data. With this multimodal scheme, these limitations are mitigated. Visual samples are illustrated in Fig. 4.7

Second, we evaluate the impact of sparse lidar samples fed along with input data. This configuration shrinks the solution space and reduces the possible ambiguity (i.e., the

Table 4.5: Performance for multimodal segmentation. Red bold indicates best performance for each metric.

Methods	Performance Metrics (CT / MR)					
	VO	RVD	ASSD	RMSD	MSSD	Overall
Cond-DCN [115]	86.42/81.26	80.86/26.69	87.02/91.46	82.88/91.04	22.79/76.21	71.99/73.33 – 72.66
Att-DCN [134]	89.35/83.32	84.26/40.60	89.42/92.64	86.60/92.96	60.09/86.07	81.94/79.12 – 80.53
Pre-BN-DCN w/o CAN	89.46/84.29	85.98/23.35	88.34/95.82	85.17/94.87	62.13/87.04	82.22/77.07 – 79.64
Post-BN-DCN w/o CAN	91.06/87.66	79.22/41.43	88.25/96.62	83.62/95.67	52.58/88.73	79.01/82.02 – 80.51
Post-BN-DCN	**91.37**/**87.46**	82.70/46.97	88.91/**96.63**	84.10/**95.74**	54.37/**89.49**	80.29/**83.26** – 81.75
Pre-IN-DCN (ours)	90.93/84.77	88.35/45.02	91.33/93.00	89.08/93.46	**65.45**/88.51	**85.03**/80.95 – 82.99
Pre-BN-DCN (ours)	91.30/84.68	**89.72**/**56.87**	**92.03**/93.09	**89.92**/93.25	58.01/86.17	84.20/82.81 – **83.50**

Table 4.6: Performance for multimodal segmentation with different T values. Red bold indicates best performance for each metric.

Methods	Performance Metrics (CT / MR)					
	VO	RVD	ASSD	RMSD	MSSD	Overall
IN-DCN (T=0)	89.56/**86.15**	77.04/65.11	89.07/**93.28**	85.33/**93.67**	48.10/87.47	77.82/**85.13** – 81.47
IN-DCN (T=1)	90.93/84.77	88.35/45.02	91.33/93.00	89.08/93.46	**65.45**/**88.51**	**85.03**/80.95 – 82.99
IN-DCN (T=2)	88.69/83.94	78.40/50.09	89.29/93.03	85.90/93.17	49.27/86.28	78.31/81.30 – 79.80
IN-DCN (T=3)	89.69/81.59	88.05/45.52	90.84/92.43	88.06/92.25	54.28/83.73	82.22/79.11 – 80.66
BN-DCN (T=0)	**91.63**/85.03	87.92/**66.74**	91.68/92.75	89.41/92.81	61.44/86.24	84.42/84.70 – **84.56**
BN-DCN (T=1)	91.30/84.68	**89.72**/56.87	**92.03**/93.09	**89.92**/93.25	58.01/86.17	84.20/82.81 – 83.50
BN-DCN (T=2)	90.53/83.58	86.45/50.19	91.34/93.02	88.83/93.30	57.32/86.55	82.89/81.33 – 82.11
BN-DCN (T=3)	89.29/81.61	85.39/35.83	90.62/92.52	87.75/92.62	54.65/85.79	81.54/77.67 – 79.60

Table 4.7: RMSE performance for multispectral data with semantic content. Note that RMSE performance is normalized with $\times 10^{-2}$.

	RGB	MultSpec	MultSpec-Smnt
Att-Lap w/o	0.46 ± 0.27	1.38 ± 0.15	**1.34** ± 0.13
Att-Lap	1.41 ± 0.26	1.35 ± 0.16	**1.29** ± 0.11

Table 4.8: RMSE performance for multispectral data with semantic content and sparse lidar data. Note that RMSE performance is normalized with $\times 10^{-2}$.

Data(%)	RGB	MultSpec	MultSpec-Smnt
0.1	0.31 ± 0.18	0.30 ± 0.11	**0.27** ± 0.08
0.07	0.38 ± 0.21	0.33 ± 0.15	**0.30** ± 0.12
0.05	0.57 ± 0.38	0.46 ± 0.19	**0.44** ± 0.17
0.01	1.21 ± 0.34	1.11 ± 0.21	**1.08** ± 0.19

nature of inverse problems) by providing a small set of exemplar samples with real data. In Tab. 4.8, sparse lidar data drastically improves performance for all models as expected. Observe that the use of a multimodal solution has better results compared to other models. Visual samples are illustrated in Fig. 4.8

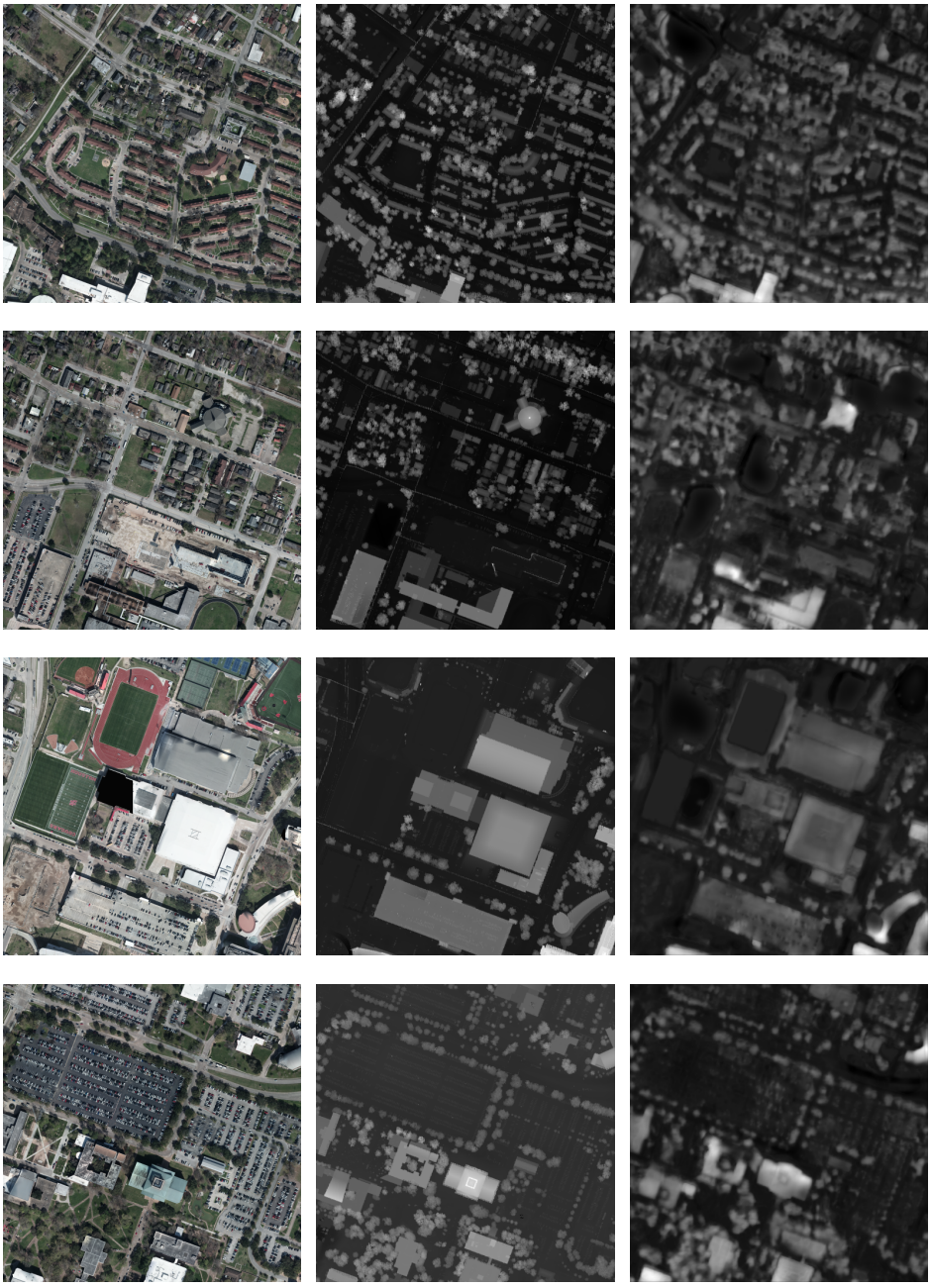

Figure 4.7: RGB data, ground truth depth data and estimated depth data with multi-modal optimization.

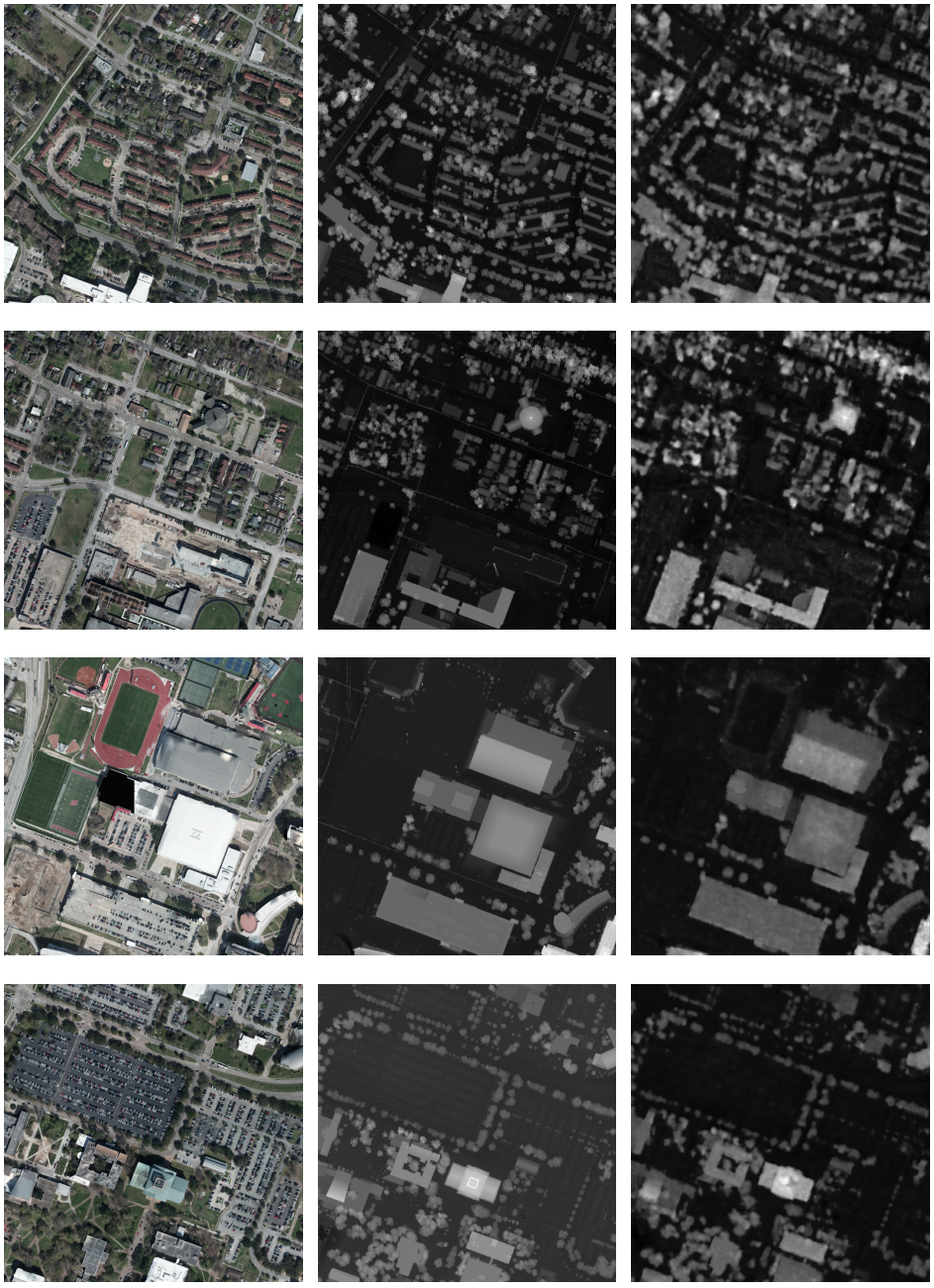

Figure 4.8: RGB data, ground truth depth data and estimated depth data with multi-modal optimization and sparse data.

CHAPTER 5

CONCLUSION

5.1 Conclusions and Limitations

At the beginning of this thesis, we ask several essential questions to ourselves about the quest to replace humans with smart tools. In particular, these questions aim to answer two particular domains: unsupervised learning and multimodal data.

Our first direction is to show that robust representations can be computed from data without supervision if we can make full use of domain-specific features by formulating them in the structure of learning models. For this purpose, we blindly extract two unknown knowledge from multispectral data: constituent materials and their fractional abundances. In particular, we propose a novel NN model that is the first successful attempt with NN for this problem. This very promising direction allows the models to retain sparsity, nonlinearity and spectral variability in the solutions. These features are critical for multispectral data, as explained in the thesis.

Furthermore, the robustness of abundance map estimation is improved with adversarial model. Indeed, the adversarial models effectively parametrize the uncertainty exhibited from the data recognized as spectral variability. This model is also identified as the first model that employs adversarial learning for spectral unmixing in a similar sense. Experimental results validate that both methods outperform the baseline methods in the literature by a large margin. We believe that the findings of our methods, especially for the NN structure, will guide the studies in this domain.

The main limitation of these methods is that while we are able to compute accurate constituent materials and their fractional abundance maps from data, our methods act

as two-individual steps that are solved individually. The solution to combine these steps is not apparent. Hence, we need to define an additional learning procedure to eliminate the possible cases in advance, such as parameter overshooting and overfitting. For our DSCN method, we precisely use a stack of 1d convolutions to transform data into a richer representation space. However, the endmember signatures are predefined so that the transformation operations are somehow semi-supervised. Therefore, this drawback needs to be handle with further steps for accurate solutions.

The second direction is to address multimodal data for various tasks. In particular, a single-objective function for multiple modalities and multiple-objective functions for single data are solved at the end. These sets ultimately provide some theoretical and practical benefits. Undoubtedly, the complexity of models is significantly reduced since all tasks are collected in a combined model. Also, since the number of parameters is decreased, the convergence rate is better, and more stable solutions can be obtained.

For multimodal data, we propose a novel method for biomedical data that is able to compute common and distinct patterns together from different modalities. For this purpose, conventional NN models are restructured to keep these two features in the solutions. As proved, the normalization layers have a critical importance for multimodal data, since it adaptively controls the data distributions. For this purpose, we reorder the normalization layer and add adversarial model to the objective to regularize the distributions as we defined. To this end, this objective intuitively clusters the data based on the visual relations. Experiments show that our methods achieve the best results compared to all configurations.

As discussed, this structure fails when multi-sliced data is used. Typically, performance must be boosted when sliced numbers are increased. We believe that this issue can solve by applying a slice registration step to the data.

For multispectral data, our contribution is minimal compared to other domains. For this domain, we show that the performance of regression tasks depends heavily on the number and dimensions of data, especially for multispectral data, since the regression tasks are continuous. Hence, more accurate results can be obtained by solving multiple objective functions. Better convergence is surely the main reason for this result.

In the objective function, the contributions of multiple objectives are empirically set. However, this step can be reproduced with the gradient-based adaptive algorithms.

5.2 Future Directions

In the future, we will continue to improve the performance of these tasks. For clarity, we can divide these possible contributions as follows:

- **Deep Neural Networks for Compressive Sensing MRI:** Undersampled MRI reconstruction is an ill-posed problem due to long acquisition durations. In theory, a small number of measurements related to data can be sufficient to reconstruct the original data with high quality. Likewise, this problem is formulated with the Fourier domain's coefficients to transfer sparse knowledge to similar statistics. We believe that our method is able to fill this missing information with its generalization capacity.

- **MultiModal MultiSpectral Unmixing:** As indicated, the dimensions of multispectral data differ based on sensor types. However, our DSCN model is able to compute latent representations from these sources, since convolution kernels are utilized. Hence, spectral unmixing can be solved for different scenes simultaneously to obtain robust latent features.

- **Multimodal-Normalization Layer:** As stated, our solution for multimodal data is still unimodal, even if the adversarial model pushes the solutions to the multimodal scenario. For this purpose, conventional normalization layers can be replaced by novel multimodal-normalization layers that calculate multiple normalization parameters in an unsupervised manner. Of course, these models must learn which normalization layer must be used for a specific data source.

- **Spectral estimation from RGB image:** As indicated, multispectral data has several drawbacks. Solving a regression problem for multispectral data can be tedious in particular. For this purpose, as in our thesis, we aim to exploit multiple objective functions (i.e., lidar, context etc.) to obtain more accurate reconstruction performance for spectral estimation task.

REFERENCES

[1] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 201–208, 2010.

[2] P. Vincent *et al.*, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, pp. 214–223, 2017.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.

[5] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[6] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[7] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[8] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[9] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[10] P. Werbos, “Beyond regression:" new tools for prediction and analysis in the behavioral sciences,” *Ph. D. dissertation, Harvard University*, 1974.

[11] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott, “Deep hierarchies in the primate visual cortex: What can we learn for computer vision?,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1847–1871, 2012.

[12] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, *et al.*, “Large scale distributed deep networks,” in *Advances in neural information processing systems*, pp. 1223–1231, 2012.

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

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[16] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*, vol. 11700. Springer Nature, 2019.

[17] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[18] L. Bottou and Y. L. Cun, “Large scale online learning,” in *Advances in neural information processing systems*, pp. 217–224, 2004.

[19] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.

[20] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, pp. 1139–1147, 2013.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, 2013.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] S. Wager *et al.*, "Dropout training as adaptive regularization," in *NIPS*, pp. 351–359, 2013.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[27] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 901–909, 2016.

[28] S. Kolouri, G. K. Rohde, and H. Hoffman, "Sliced wasserstein distance for learning gaussian mixture models," *arXiv preprint arXiv:1711.05376*, 2017.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[30] J. M. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical,

statistical, and sparse regression-based approaches," *IEEE JSTARS*, pp. 354–379, 2012.

[31] A. J. Brown *et al.*, "Hydrothermal formation of clay-carbonate alteration assemblages in the nili fossae region of mars," *Earth and Planetary Science Letters*, vol. 297, no. 1-2, pp. 174–182, 2010.

[32] M. A. Toksöz and I. Ulusoy, "Hyperspectral image classification via kernel basic thresholding classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 715–728, 2016.

[33] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, "Blind spectral unmixing based on sparse nonnegative matrix factorization," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 1112–1125, 2011.

[34] R. Heylen, D. Burazerovic, and P. Scheunders, "Non-linear spectral unmixing by geodesic simplex volume maximization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 534–542, 2011.

[35] J. J. Settle, "On the relationship between spectral unmixing and subspace projection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 4, pp. 1045–1046, 1996.

[36] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE signal processing magazine*, vol. 19, no. 1, pp. 44–57, 2002.

[37] B. Hapke, "Bidirectional reflectance spectroscopy: 1. theory," *Journal of Geophysical Research: Solid Earth*, pp. 3039–3054, 1981.

[38] Y. Zhou, A. Rangarajan, and P. D. Gader, "A gaussian mixture model representation of endmember variability in hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2242–2256, 2018.

[39] C.-I. Chang and A. Plaza, "A fast iterative algorithm for implementation of pixel purity index," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 63–67, 2006.

[40] M. E. Winter, "N-findr: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *SPIE*, pp. 266–275, 1999.

[41] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE TGRS*, pp. 898–910, 2005.

[42] C.-I. Chang, C.-C. Wu, W. Liu, and Y.-C. Ouyang, "A new growing method for simplex-based endmember extraction algorithm," *IEEE transactions on geoscience and remote sensing*, vol. 44, no. 10, pp. 2804–2819, 2006.

[43] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 3, pp. 765–777, 2007.

[44] 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 transactions on Geoscience and Remote Sensing*, vol. 42, no. 10, pp. 2085–2095, 2004.

[45] A. Zare and P. Gader, "Sparsity promoting iterated constrained endmember detection in hyperspectral imagery," *IEEE GRSL*, pp. 446–450, 2007.

[46] J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *IGARSS*, 2008.

[47] A. Signoroni, M. Savardi, A. Baronio, and S. Benini, "Deep learning meets hyperspectral image analysis: a multidisciplinary review," *Journal of Imaging*, vol. 5, no. 5, p. 52, 2019.

[48] Y. Qu and H. Qi, "udas: An untied denoising autoencoder with sparsity for spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1698–1712, 2018.

[49] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "Daen: Deep autoencoder networks for hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[50] R. Heylen *et al.*, "Fully constrained least squares spectral unmixing by simplex projection," *IEEE TGRS*, pp. 4112–4122, 2011.

[51] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Bilinear models for nonlinear unmixing of hyperspectral images," in *Hyperspectral Image and Signal Pro-*

cessing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on, pp. 1–4, IEEE, 2011.

[52] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4153–4162, 2011.

[53] 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," *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2951–2962, 2009.

[54] B. Zhang, L. Zhuang, L. Gao, W. Luo, Q. Ran, and Q. Du, "Pso-em: A hyperspectral unmixing algorithm based on normal compositional model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7782–7792, 2014.

[55] X. Du, A. Zare, P. Gader, and D. Dranishnikov, "Spatial and spectral unmixing using the beta compositional model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 1994–2003, 2014.

[56] R. A. Borsoi, T. Imbiriba, and J. C. M. Bermudez, "Deep generative endmember modeling: An application to unsupervised spectral unmixing," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 374–384, 2019.

[57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, pp. 807–814, 2010.

[58] F. Palsson *et al.*, "Neural network hyperspectral unmixing with spectral information divergence objective," *IGARSS*, 2017.

[59] R. Memisevic and D. Krueger, "Zero-bias autoencoders and the benefits of co-adapting features," *Stat*, p. 13, 2014.

[60] J. W. Boardman, "Geometric mixture analysis of imaging spectrometry data," in *IGARSS*, pp. 2369–2371, 1994.

[61] J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE TGRS*, pp. 779–785, 1994.

[62] M.-D. Iordache *et al.*, "Sparse unmixing of hyperspectral data," *IEEE TGRS*, pp. 2014–2039, 2011.

[63] M.-D. Iordache *et al.*, "Collaborative sparse regression for hyperspectral unmixing," *IEEE TGRS*, pp. 341–354, 2014.

[64] J. Chen *et al.*, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Transactions on Signal Processing*, pp. 480–492, 2013.

[65] C. Févotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE TIP*, pp. 4810–4819, 2015.

[66] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[67] L. Chunjie *et al.*, "Cosine normalization: Using cosine similarity instead of dot product in neural networks," *arXiv preprint arXiv:1702.05870*, 2017.

[68] P. E. Dennison *et al.*, "A comparison of error metrics and constraints for multiple endmember spectral mixture analysis and spectral angle mapper," *Remote Sensing of Environment*, pp. 359–367, 2004.

[69] G. Camps-Valls, "Kernel spectral angle mapper," *Electronics Letters*, pp. 1218–1220, 2016.

[70] G. Camps-Valls *et al.*, "Composite kernels for hyperspectral image classification," *IEEE GRSL*, pp. 93–97, 2006.

[71] M. Fauvel *et al.*, "Evaluation of kernels for multiclass classification of hyperspectral remote sensing data," in *ICASSP*, IEEE, 2006.

[72] H. Li and L. Zhang, "A hybrid automatic endmember extraction algorithm based on a local window," *IEEE GRSL*, pp. 4223–4238.

[73] K. Kavukcuoglu *et al.*, "Fast inference in sparse coding algorithms with applications to object recognition," *arXiv preprint arXiv:1010.3467*, 2010.

[74] R. Heylen *et al.*, "Nonlinear unmixing by using different metrics in a linear unmixing chain," *IEEE JSTARS*, pp. 2655–2664, 2015.

[75] S. Holzwarth *et al.*, "Hysens-dais 7915/rosis imaging spectrometers at dlr," in *Proceedings of the 3rd EARSeL Workshop on Imaging Spectroscopy*, pp. 3–14, 2003.

[76] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 353–360, Citeseer, 2011.

[77] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[78] S. Jalali, C. Nuzman, and I. Saniee, "Efficient deep approximation of gmms," in *Advances in Neural Information Processing Systems*, pp. 4552–4560, 2019.

[79] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307, 2016.

[80] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[81] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[82] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[83] C. Améndola, M. Drton, and B. Sturmfels, "Maximum likelihood estimates for gaussian mixtures are transcendental," in *International Conference on Mathematical Aspects of Computer and Information Sciences*, pp. 579–590, Springer, 2015.

[84] C. Jin, Y. Zhang, S. Balakrishnan, M. J. Wainwright, and M. I. Jordan, "Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences," in *Advances in Neural Information Processing Systems*, pp. 4116–4124, 2016.

[85] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2017.

[86] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[87] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[88] F. Zhu, "Hyperspectral unmixing: ground truth labeling, datasets, benchmark performances and survey," *arXiv preprint arXiv:1708.05125*, 2017.

[89] Y. Qian *et al.*, "Hyperspectral unmixing via $l_\{1/2\}$ sparsity-constrained nonnegative matrix factorization," *IEEE TGRS*, pp. 4282–4297, 2011.

[90] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5412–5427, 2014.

[91] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 101–118, 2014.

[92] F. A. Kruse *et al.*, "Comparison of aviris and hyperion for hyperspectral mineral mapping," in *11th JPL Airborne Geoscience Workshop*, 2002.

[93] P. Gader *et al.*, "Muufl gulfport hyperspectral and lidar airborne data set," *Univ. Florida, Gainesville, FL, USA, Tech. Rep. REP-2013-570*, 2013.

[94] Y. Zhou, A. Rangarajan, and P. D. Gader, "A spatial compositional model for linear unmixing and endmember uncertainty estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5987–6002, 2016.

[95] R. Heylen and P. Scheunders, "A multilinear mixing model for nonlinear spectral unmixing," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 1, pp. 240–251, 2016.

[96] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret, "Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3017–3025, 2012.

[97] D. Stein, "Application of the normal compositional model to the analysis of hyperspectral imagery," in *Advances in techniques for analysis of remotely sensed data, 2003 IEEE Workshop on*, pp. 44–51, IEEE, 2003.

[98] R. Heylen and P. Scheunders, "A multilinear mixing model for nonlinear spectral unmixing.," *IEEE TGRS*, pp. 240–251.

[99] A. Halimi *et al.*, "Nonlinear unmixing of hyperspectral images using a generalized bilinear model.," *IEEE TGRS*, pp. 4153–4162.

[100] Y. Altmann *et al.*, "Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery.," *IEEE TIP*, vol. 21, no. 6, pp. 3017–3025, 2012.

[101] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[102] E. A. R. Piedra, B. M. Ellingson, R. K. Taira, S. El-Saden, A. A. Bui, and W. Hsu, "Brain tumor segmentation by variability characterization of tumor boundaries," in *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 206–216, Springer, 2016.

[103] R. Ayachi and N. B. Amor, "Brain tumor segmentation using support vector machines," in *European conference on symbolic and quantitative approaches to reasoning and uncertainty*, pp. 736–747, Springer, 2009.

[104] K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A. V. Nori, A. Criminisi, D. Rueckert, and B. Glocker, "Deepmedic for brain tumor segmentation," in *International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*, pp. 138–149, Springer, 2016.

[105] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[106] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[107] B. Settles, "Biomedical named entity recognition using conditional random fields and rich feature sets," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pp. 107–110, 2004.

[108] M. Rezaei, K. Harmuth, W. Gierke, T. Kellermeier, M. Fischer, H. Yang, and C. Meinel, "A conditional adversarial network for semantic segmentation of brain tumor," in *International MICCAI Brainlesion Workshop*, pp. 241–252, Springer, 2017.

[109] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[110] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[111] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 240–248, Springer, 2017.

[112] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical image analysis*, vol. 35, pp. 18–31, 2017.

[113] I. Lavdas, B. Glocker, K. Kamnitsas, D. Rueckert, H. Mair, A. Sandhu, S. A. Taylor, E. O. Aboagye, and A. G. Rockall, "Fully automatic, multiorgan segmentation in normal whole body magnetic resonance imaging (mri), using classification forests (cf s), convolutional neural networks (cnn s), and a multi-atlas (ma) approach," *Medical physics*, vol. 44, no. 10, pp. 5210–5220, 2017.

[114] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[115] O. Hadad, R. Bakalo, R. Ben-Ari, S. Hashoul, and G. Amit, "Classification of breast lesions using cross-modal deep learning," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. 109–112, IEEE, 2017.

[116] V. V. Valindria, N. Pawlowski, M. Rajchl, I. Lavdas, E. O. Aboagye, A. G. Rockall, D. Rueckert, and B. Glocker, "Multi-modal learning from unpaired images: Application to multi-organ segmentation in ct and mri," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 547–556, IEEE, 2018.

[117] Y. Zhan, M. Dewan, and X. S. Zhou, "Cross modality deformable segmentation using hierarchical clustering and learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 1033–1041, Springer, 2009.

[118] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems*, pp. 506–516, 2017.

[119] H. Bilen and A. Vedaldi, "Universal representations: The missing link between faces, text, planktons, and cat breeds," *arXiv preprint arXiv:1701.07275*, 2017.

[120] X. Tang, B. Du, J. Huang, Z. Wang, and L. Zhang, "On combining active and transfer learning for medical data classification," *IET Computer Vision*, vol. 13, no. 2, pp. 194–205, 2018.

[121] C. Ouyang, K. Kamnitsas, C. Biffi, J. Duan, and D. Rueckert, "Data efficient unsupervised domain adaptation for cross-modality image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 669–677, Springer, 2019.

[122] S. A. Sriram, A. Paul, Y. Zhu, V. Sandfort, P. J. Pickhardt, and R. M. Summers, "Multilevel unet for pancreas segmentation from non-contrast ct scans through domain adaptation," in *Medical Imaging 2020: Computer-Aided Diagnosis*, vol. 11314, p. 113140K, International Society for Optics and Photonics, 2020.

[123] O. S. Gedik and A. A. Alatan, "Rgbd data based pose estimation: Why sensor fusion?," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 2129–2136, IEEE, 2015.

[124] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4003, 2016.

[125] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International Conference on Machine Learning*, pp. 794–803, 2018.

[126] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

[127] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Advances in Neural Information Processing Systems*, pp. 527–538, 2018.

[128] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6129–6138, 2017.

[129] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[130] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1871–1880, 2019.

[131] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[132] C. Örnek and E. Vural, "Nonlinear supervised dimensionality reduction via smooth regular embeddings," *Pattern Recognition*, vol. 87, pp. 55–66, 2019.

[133] Y. Xu, B. Du, L. Zhang, D. Cerra, M. Pato, E. Carmona, S. Prasad, N. Yokoya, R. Hänsch, and B. Le Saux, "Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 ieee grss data fusion contest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 6, pp. 1709–1724, 2019.

[134] T. Heimann, B. Van Ginneken, M. A. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes, *et al.*, "Comparison and evaluation of methods for liver segmentation from ct datasets," *IEEE transactions on medical imaging*, vol. 28, no. 8, pp. 1251–1265, 2009.

[135] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.