Sparse and Archetypal Decomposition Algorithms for Hyperspectral Image Restoration and Spectral Unmixing

Alexandre Zouaoui

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques et Informatique
Unité de recherche : Laboratoire Jean Kuntzmann

Algorithmes de décomposition parcimonieuse et archétypale pour la restauration d'images hyperspectrales et le démélange spectral

Sparse and Archetypal Decomposition Algorithms for Hyperspectral Image Restoration and Spectral Unmixing

Présentée par :

Alexandre ZOUAOUI

Direction de thèse :

Julien MAIRAL DIRECTEUR DE RECHERCHE, CENTRE INRIA UNIVERSITE GRENOBLE ALPES	Directeur de thèse
Jocelyn CHANUSSOT Directeur de recherche, INRIA	Co-directeur de thèse

Rapporteurs :

MARIO A. T. FIGUEIREDO
FULL PROFESSOR, INSTITUTO NACIONAL DE TELECOMUNICAÇÕES
NICOLAS DOBIGEON
PROFESSEUR, INP - ENSEEIHT TOULOUSE

Thèse soutenue publiquement le **19 mars 2024**, devant le jury composé de :

JERÔME MARS, PROFESSEUR DES UNIVERSITES, GRENOBLE INP	Président
JULIEN MAIRAL, DIRECTEUR DE RECHERCHE, CENTRE INRIA UNIVERSITE GRENOBLE ALPES	Directeur de thèse
JOCELYN CHANUSSOT, PROFESSEUR DES UNIVERSITES, GRENOBLE INP	Co-directeur de thèse
MARIO A. T. FIGUEIREDO, FULL PROFESSOR, INSTITUTO NACIONAL DE TELECOMUNICAÇÕES	Rapporteur
NICOLAS DOBIGEON, PROFESSEUR, INP - ENSEEIHT TOULOUSE	Rapporteur
XIUPING JIA, ASSOCIATE PROFESSOR, UNIVERSITY OF NEW SOUTH WALES	Examinatrice

Invités :

BEHNOOD RASTI
SENIOR SCIENTIST, HELMHOLTZ-ZENTRUM DRESDEN-ROSSENDORF (HZDR)

ABSTRACT

Hyperspectral imagery has emerged as a powerful tool for monitoring remote areas and accurately identifying materials within a scene, contributing significantly to our understanding of the environment. While it finds applications in fields as diverse as precision farming, astronomy and plastics recycling, there are still considerable challenges to address for its wider adoption. Traditionally confined to domain experts, the increasing volume of hyperspectral data requires more efficient processing methods than manual labor can provide to assist decision-makers effectively.

The rise of deep learning, primarily driven by advancements in understanding natural images from the computer vision research community, has introduced new techniques able to automatically handle the inherent complexity of hyperspectral images in tasks such as land cover classification, image restoration, spectral unmixing or data fusion. The main ingredients for the success of artificial neural networks include a massive quantity of data, improved computing hardware and advances in learning algorithms. Moreover, these purely data-driven systems rely on sophisticated design choices based on prior knowledge about the task at hand. Yet, the rationale behind deep model predictions is often missing, hindering their usability in key scientific applications where accountability is crucial. Conversely, model-based learning methods rely on a set of core assumptions that the predictor uses to output a value given an unseen input, allowing the prediction to be logically explained and providing a potentially useful pattern to inform end-users. *Model-based deep learning* has emerged as a candidate to combine the merits of both approaches, effectively addressing the drawbacks associated to each paradigm. In a nutshell, this hybrid framework integrates principled mathematical models within data-driven systems.

In this thesis, we contribute novel algorithms based on sparse and archetypal signal decomposition designed to address hyperspectral image restoration and spectral unmixing, two fundamental tasks prevalent in applications leveraging hyperspectral data. Inspired by the aforementioned research direction, we developed efficient learning methods that are mathematically grounded from a signal processing standpoint.

We begin by introducing T3SC, a novel deep unfolding architecture derived from a sparse coding model, specifically tailored to overcome challenges posed by hyperspectral images. This algorithm unrolling approach, mixing deep learning and classical model-based image processing principles, enables efficient end-to-end training of model parameters without requiring massive data.

Moving on to spectral unmixing, a critical step in further analyzing hyperspectral images, we first develop an efficient approach for blind linear unmixing, EDAA, based on archetypal analysis. Under this framework, the spectra of the materials of interest, i.e.

endmembers, are modeled as convex combinations of pixels present in the scene. Our method achieves state-of-the-art results on a broad range of unmixing datasets by leveraging an adequate model selection procedure and without requiring extensive hyperparameter tuning.

Our next contribution extends the archetypal analysis framework to the semi-supervised unmixing setting, assuming knowledge of the number of materials of interest in the scene. Despite the non-convex nature of the associated minimization problem, our approach, SUnAA, demonstrates its viability in tackling hard unmixing scenarios in which traditional sparse unmixing techniques tend to fail.

Finally, we conduct a comprehensive comparison of various unmixing techniques across diverse scenarios, highlighting the advantages of supervised, semi-supervised, and blind unmixing categories depending on the experimental setup. To facilitate exploration and application of these methods, we provide an open-source Python package named HySUPP.

All these contributions have led to the development of an open-source software package, enhancing accessibility, reproducibility and usability in the hyperspectral research community.

Keywords: machine learning, signal processing, hyperspectral imaging, image restoration, spectral unmixing, sparse coding, algorithm unrolling, archetypal analysis.

RÉSUMÉ

L'imagerie hyperspectrale est devenue un puissant outil pour surveiller les zones éloignées et identifier avec précision les matériaux présents dans une scène, contribuant ainsi de manière significative à notre compréhension de l'environnement. Bien qu'elle trouve des applications dans des domaines aussi variés que l'agriculture de précision, l'astronomie et le recyclage des plastiques, il reste encore des défis considérables à relever pour son adoption à plus grande échelle. Traditionnellement réservé aux experts du domaine, le volume croissant de données hyperspectrales nécessite des méthodes de traitement plus efficaces que celles que le travail manuel peut fournir pour aider les décideurs de manière pertinente.

L'essor de l'apprentissage profond a introduit de nouvelles techniques capables de gérer automatiquement la complexité inhérente des images hyperspectrales dans des tâches telles que la classification des sols, la restauration d'images et le démélange spectral. Les principaux ingrédients du succès des réseaux de neurones artificiels tiennent en l'abondance de données, l'amélioration des capacités de calculs informatiques ainsi qu'aux progrès dans les algorithmes d'apprentissage. De plus, ces systèmes qui reposent exclusivement sur les données nécessitent des choix de conception sophistiqués fondés sur une connaissance préalable de la tâche à accomplir. Pourtant, la justification des prédictions des modèles profonds fait souvent défaut. À l'inverse, les méthodes d'apprentissage dites classiques, fondées sur des modèles, reposent sur un ensemble d'hypothèses de base que le prédicteur utilise pour générer une valeur à partir d'une nouvelle entrée, permettant d'expliquer logiquement la prédiction et fournissant un modèle potentiellement utile pour informer les utilisateurs. Le nouveau paradigme intitulé *model-based deep learning* a pour but de combiner les avantages de chaque approche. Cette démarche hybride intègre des principes mathématiques hérités de la littérature du traitement du signal au sein de systèmes qui reposent sur les données à disposition pour l'apprentissage.

Dans cette thèse, nous proposons de nouveaux algorithmes qui reposent sur la notion de décomposition parcimonieuse et archétypale du signal, et qui ont été spécifiquement conçus pour traiter la restauration d'images hyperspectrales et le démélange spectral. En s'inspirant de la direction de recherche prélablement présentée, nous avons développé des méthodes d'apprentissage efficaces qui s'appuient sur principes de modélisation propres au traitement du signal.

Nous commençons par présenter T3SC, une nouvelle architecture dite déroulée, qui repose sur un modèle de codage parcimonieux, conçue pour surmonter les défis spécifiques posés par les images hyperspectrales. Cette approche hybride, mêlant apprentissage profond et principes classiques de traitement d'images permet un entraînement efficace de

bout en bout des paramètres du modèle sans nécessiter une quantité massive de données.

Dans le cadre du démélange spectral, une étape critique dans l'analyse plus approfondie des images hyperspectrales, nous développons d'abord une approche efficace pour le démélange linéaire dit aveugle, EDAA, fondé sur l'analyse archétypale. Sous cette formulation, les spectres des matériaux d'intérêt sont modélisés sous forme de combinaisons convexes de pixels présents dans la scène. Notre méthode permet d'obtenir des résultats état-de-l'art sur une large gamme de jeux de données en tirant parti d'une procédure de sélection de modèle adéquate et sans nécessiter un réglage fastidieux des hyperparamètres.

Notre contribution suivante étend le cadre d'analyse archétypale au contexte de démélange semi-supervisé. Malgré la nature non convexe du problème de minimisation associé, notre approche, SUnAA, démontre sa pertinence dans la résolution de scénarios de démélange difficiles dans lesquels les techniques traditionnelles de démélange dites parcimonieuses ont tendance à échouer.

Enfin, nous effectuons une comparaison complète de diverses techniques de démélange dans plusieurs scénarios simulés, mettant en évidence les avantages des catégories de démélange supervisé, semi-supervisé et aveugle en fonction de la configuration expérimentale. Pour encourager l'adoption de ces méthodes, nous proposons un package Python open source nommé HySUPP.

Toutes ces contributions ont conduit au développement d'un logiciel libre dédié, améliorant l'accessibilité et la reproductibilité au sein de la communauté d'utilisateurs et de chercheurs en imagerie hyperspectrale.

Mots-clés: apprentissage machine, traitement du signal, imagerie hyperspectrale, restauration d'images, démélange spectral, codage parcimonieux, analyse archétypale.

ACKNOWLEDGMENTS

This thesis would not have seen the light of day without my supervisors relentless support. Julien, I owe you a great deal! It all started when you reached out during my internship at NLE in 2019 and offered me to join Thoth as a research engineer. I could not be more grateful for the opportunities that followed. Your cleverness, sense of rigor, and overall scientific qualities are clear to any outsider, but what struck me the most is your patience and kindness, that were both instrumental during my PhD, especially when I hit rock-bottom health-wise. Jocelyn, I truly feel honored to having had you by my side during this roller-coaster-esque journey. I am fortunate to have gotten to know the great person that is behind the impressive scholar and I am looking forward to continuing our fruitful collaboration at DSE. This thesis would have been much different had I not met Behnood thanks to Jocelyn. Behnood took me under his wing and quickly brought me up to date on hyperspectral unmixing. His acute knowledge on many more topics inspired me to further develop my research skills. I will always remember your 2023 stay in Grenoble and our trips to Geneva and Marseille, those were fun times!

I could not have dreamt of a better jury willing to evaluate my work: Xiuping Jia, Jérôme Mars, and I particularly thank Mário Figueiredo and Nicolas Dobigeon for accepting to review this manuscript.

The Thoth team at Inria proved to be the perfect environment for me to thrive and I thank Karteek Alahari, Pierre Gaillard, Michael Arbel, and Hadrien Hendrycks for making it so enjoyable. My special thanks go to Nathalie Gillot, whose utmost reliability and kindness will be sincerely missed!

As you will shortly notice, I could not have done all the work myself, and I owe much to my collaborators Théo Bodrito and Gedeon Muhawenayo.

The list of friends and colleagues I made here in Grenoble is virtually endless. At Thoth, I feel lucky to have met Bianca, Camila, Ieva, Juliette, Lina, Margot, Mathilde, Minttu, Pia, Alberto, Alexander, Andrei, Bruno, Dexiong, Emmanuel, Enrico, Florent, Gaspard, Heeseung, Houssam, Hubert, Jules, Julien Z., Loic, Bulent, Nassim, Nikita, Ricardo, Romain, Thomas D.M., Thomas L., Thomas R., Timothée, Valentin, Vladyslav, and Zhiqi.

At Inria, I felt privileged to work under the same roof as Nachi, who brought me to crossfit, one of my core stability pillars. There I'm grateful to have met Laila, who then introduced me her wonderful LIPhy lab mates that quickly became friends. Special thanks to my bike ride partners: Caroline, Ivan, and Nathalie.

To my family, for their flawless support and encouragement throughout the PhD.

CONTENTS

1

INTRODUCTION

Hyperspectral (HS) imaging, also referred to as image spectrometry, stands as a significant advancement within geoscience and remote sensing (RS) owing to its ability to precisely identify a broad range of materials thanks to their spectral fingerprints. In the past decade, substantial efforts have been directed towards the processing and analysis of HS data, predominantly under the guidance of domain experts. Nevertheless, as the quantity of data continues to grow, new challenges arise to alleviate expensive and tedious manual labor, in order to ultimately enhance productivity. Consequently, there is a pressing need for the development of more efficient and automated methods to address diverse applications of HS RS. Examples of such tasks include large-scale land cover mapping, HS image restoration, HS unmixing, data fusion and multimodal data analysis.

Boosted by the recent advances in deep learning for computer vision, earth observation (EO) has witnessed a surge of deep models aiming to improve prediction performance over classical model-based image processing methods. While the latter techniques have proven effective under highly idealized assumptions, their limited modeling capability often fail to grasp intricate real-world scenarios due to the existence of unknown, uncertain and unpredictable factors [Hong et al., 2021]. On the other hand, deep learning models benefit from extensive modeling capacity enabling them to handle considerably more complex scenes at the cost of potential overfitting. That being said, the effectiveness of deep models primarily hinges on their ability to leverage prior knowledge through advanced network architecture design, empirical training techniques and huge amounts of annotated data. Moreover, deep networks tend to lack *interpretability*, a desirable property of learning models for scientific applications, due to their “black box" nature. Ideally, a reliable prediction model should not only perform accurately but also be able to explain its predictions. To address this limitation, the field of explainable artificial intelligence (XAI) has emerged as a burgeoning area of research.

Concurrently, at the crossroad between model-based methods and deep learning stands the emerging field of *model-based deep learning* [Shlezinger et al., 2023] whose aim is to combine the advantages of both approaches, by embedding principled mathematical models within data-driven systems. These hybrid approaches are generally task-specific, yet they may be relevant to a multitude of research domains provided that one has access to some level of reliable mathematical modeling. One of the key strategies within model-based deep learning that we will delve into is *algorithm unrolling* or *deep unfolding* [Monga et al., 2021], aiming to transform iterative optimizers into trainable parameterized architectures.

Unrolling methods were first proposed to develop fast neural network approximations for sparse coding [Gregor and LeCun, 2010]. The core concept revolves around training a

non-linear, feed-forward predictor with a specific architecture and a fixed depth to produce the best possible approximation of the sparse code. Unrolling methods have gained substantial popularity owing to their capacity to craft efficient, high-performance, and interpretable network architectures, all achieved with reasonably sized training datasets.

In this thesis, we present our contributions aligned with the aforementioned research directions, with a particular emphasis on their applications in HS imaging. Building upon recent advancements in the restoration of natural images, we introduce a novel trainable two-layered spectral-spatial sparse coding model to tackle HS denoising. We then shift our focus to blind HS unmixing, offering an efficient solution for an archetypal analysis formulation within a pure model-based framework. Next, we develop a new mixing model tailored for semi-supervised HS unmixing, addressing library mismatch. Finally, we provide a comprehensive overview of image processing and machine learning techniques for HS unmixing, complemented by a Python package designed to facilitate the reproducibility of research findings in the field of unmixing. For a deeper dive into these contributions, please refer to Section 1.1. The remaining sections of this chapter offer an essential groundwork for the upcoming chapters. First, we provide some background on HS imaging in Section 1.2. Next, we delve into the realm of HS restoration in Section 1.3. Lastly, in Section 1.4, we introduce the fundamental components of HS unmixing.

1.1 Contributions of the thesis

This thesis makes significant contributions to the field of HS image processing, with a particular emphasis on the creation of data and computationally efficient methodologies. Each of these contributions has resulted in the development of a corresponding open-source software package. These contributions are presented hereafter.

Chapter 2 introduces a new trainable network architecture, called T3SC, derived from a sparse coding model tailored to address the specificity of HS images in the context of denoising. We extend the approach developed by [Lecouat et al., 2020b] for natural images restoration by adopting a two-layer model as a mean to provide a shared architecture for different HS sensors. Beyond mere quantitative outcomes, we believe that our work sheds light on the efficiency of our models in learning a relatively concise set of parameters, while achieving state-of-the-art results. This efficiency is especially critical due to the limited availability of training data for HS denoising and the challenges associated with training deep learning models for this purpose.

T. Bodrito*, A. Zouaoui*, J. Chanussot, and J. Mairal. A trainable spectral-spatial sparse coding model for hyperspectral image restoration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021

(*equal contributions: Théo and I started this fruitful collaboration at the start of my PhD. Théo handled the implementation while I was involved in the design of the architecture based on my previous experience with trainable sparse coding models tailored to natural images classification. Unfortunately, I could not assist Théo towards the end of the project due to serious health issues. Théo successfully improved the original design by developing a noise adaptive sparse coding component and proposed a self-supervised extension that are not mentioned in this manuscript.)

In Chapter 3, our attention turns to another critical task inherent to HS image processing: spectral unmixing. We introduce a novel algorithm based on archetypal analysis for blind hyperspectral unmixing. Our approach, called EDAA, leverages an entropic gradient descent strategy, steering the optimization process within the geometry induced by the negative entropy function. This framework is tailored to accommodate the constraints imposed by the linear mixing model. Notably, thanks to an efficient GPU-compatible implementation, we propose a model selection procedure that enhances the robustness of our method to hyper-parameter choices, all the while maintaining reasonable computational complexity. Finally, our work highlights the effectiveness of the archetypal analysis formulation as a viable approach to tackle blind HS unmixing.

A. Zouaoui, G. Muhawenayo, B. Rasti, J. Chanussot, and J. Mairal. Entropic descent archetypal analysis for blind hyperspectral unmixing. In *IEEE Transactions on Image Processing*, 2023

Chapter 4 further exploits archetypal analysis in the context of semi-supervised HS unmixing. Our novel formulation, called SUnAA, assumes that the spectra of the materials of interest, or *endmembers*, are a convex combination of endmembers provided by a spectral library, with the added knowledge that the number of endmembers in the scene is known. Unlike most conventional sparse unmixing methods, the objective function in this formulation is non-convex. As a result, we employ an iterative approach to minimize this objective, leveraging the active set algorithm proposed in [Chen et al., 2014].

B. Rasti*, A. Zouaoui*, J. Mairal, and J. Chanussot. SUnAA: Sparse Unmixing using Archetypal Analysis. In *IEEE Geosciences and Remote Sensing Letters*, 2023

(*equal contributions: Behnood and I started collaborating one year into my PhD. Behnood shared with me his acute knowledge of spectral unmixing while I conducted the experimental study. Inspired by EDAA presented in Chapter 3, Behnood formulated SUnAA and I investigated several optimization methods and eventually selected the active-set optimization algorithm that best suited the semi-supervised unmixing setup.)

Chapter 5 serves as a comprehensive overview of both advanced and conventional unmixing approaches. We offer a critical comparison between advanced and conventional techniques depending on the level of prior knowledge available about the endmembers. We thoroughly assess the performance of various unmixing techniques using a diverse set of unmixing datasets, encompassing both simulated and real-world examples. Notably, we highlight the advantages of the three main unmixing categories - supervised, semi-supervised, and blind - in different unmixing scenarios. Finally, we provide an open-source Python package, named HySUPP, that contains an exhaustive list of unmixing methods, in order to foster research results reproducibility.

B. Rasti, A. Zouaoui, J. Mairal and J. Chanussot. Image Processing and Machine Learning for Hyperspectral Unmixing: An Overview and the HySUPP Python Package. *arXiv preprint arXiv:2308.09375*, 2023

Behnood played a leading role in this comprehensive work. He completed the overview by himself while I designed and implemented the toolbox. In Chapter 5 I primarily focus on my own contributions related to the HySUPP toolbox and leave aside the significant survey on the existing spectral unmixing works by Behnood.

1.2 Background on Hyperspectral Imaging

Hyperspectral (HS) imaging consists of measuring the electromagnetic spectrum at each pixel location within a scene by using multiple contiguous spectral bands. Thanks to its richer spectral information compared to traditional RGB images, HS imaging enables more precise materials identification. This capability has led to a diverse array of applications in fields such as astronomy, agriculture, biomedical imaging and geosciences. However, this technology also presents numerous challenges. As a relatively new analytical technique, the potential of HS imaging has not yet been fully realized.

1.2.1 Characteristics

HS imaging falls within the broader category of techniques commonly known as spectral imaging. The term “hyperspectral" has gained prominence over its counterpart, “spectroscopy". NASA played a pivotal role in developing this technology in the mid-1980s. HS imaging is closely related to multispectral (MS) imaging. HS imaging uses continuous and contiguous wavelength ranges (e.g., 400 to 1000 nm in 5 nm intervals), while MS imaging focuses on specific, selected wavelengths at discrete locations (e.g., 400-425, 550-570, 760-790, and 990-1010 nm, resulting in a four-band MS image).

In this thesis, we develop methods specifically tailored to HS imaging, drawing inspiration from various research domains, including computer vision, machine learning and signal processing.

1.2.2 Acquisition

HS sensors acquire data in the form of a series of images, with each image corresponding to a specific narrow wavelength range within the electromagnetic spectrum, often referred to as a spectral band. These individual images are combined to construct a three-dimensional HS datacube (x, y, λ) for subsequent analysis. In this context, x and y denote the two spatial dimensions of the scene, while λ represents the spectral dimension.

There are four main techniques for acquiring the three-dimensional dataset of a HS cube. The choice of technique ultimately depends on the specific application, given that each technique has context-dependent advantages and drawbacks.

Spatial scanning Each two-dimensional sensor output represents a full slit spectrum (x, λ). A push broom scanner is commonly used to obtain slit spectra by projecting a strip of the scene onto a slit and dispersing the slit image with a prism or a grating. To ensure accurate image reconstruction, this technique requires stable mounts or precise pointing

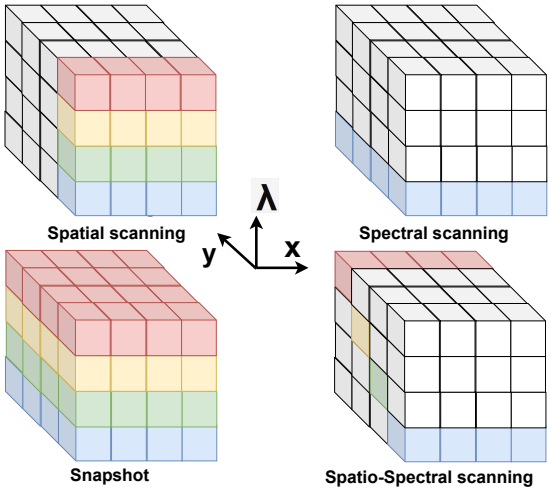


Figure 1.1: Hyperspectral imaging acquisition techniques.

information. It finds practical applications in remote sensing (RS) and scanning materials in motion on a conveyor belt.

Spectral scanning Each two-dimensional sensor output represents a monochromatic map of the scene (x, y). Optical band-pass filters are commonly used to capture the scene by cycling through one after another while the platform remains stationary. However, it is important to note that spectral smearing can occur when objects within the scene are not stationary. While these sensors can be deployed on aircraft, it is often necessary to employ techniques that extract spatial features from each monochromatic image to realign the pixels and correct for any misalignment caused by the movement of the platform.

Snapshot HS devices designed for snapshot or non-scanning operations provide the complete datacube in a single capture, without the need for scanning. In this approach, a single snapshot captures a perspective projection of the entire datacube, from which its three-dimensional structure can be subsequently reconstructed. While various systems have been developed with a focus on increased light throughput and reduced acquisition time, it is worth noting that this approach often entails high computational demands and manufacturing costs.

Spatiospectral scanning Each two-dimensional sensor output represents a wavelength-coded spatial (x, y) map of the scene (i.e. with $\lambda = \lambda(y)$). This method uses a camera at some non-zero distance behind a basic slit spectroscope. Spatiospectral scanning combines some of the benefits of both spatial and spectral scanning techniques, effectively mitigating some of the drawbacks associated with each approach.

In this thesis, our techniques are designed to operate seamlessly across various acquisition modes, as they directly process the HS datacube. However, in the section dedicated to image restoration, we concentrate on addressing a specific issue, namely *stripe* noise. These artifacts commonly arise from defective spatial scanners.

1.2.3 Applications

Hyperspectral remote sensing (HS RS) finds diverse applications across numerous fields. Initially designed for mining and geology, its utility has since expanded into a wide range of areas, including ecology, waste sorting, and historical manuscript research. The accessibility of this technology to the public has grown steadily. Leading organizations like NASA and USGS have compiled catalogs containing spectral signatures of various minerals and made them easily accessible to researchers through online repositories. In what follows, we provide a brief overview of some significant applications of HS imaging.

Agriculture HS imaging serves as a valuable tool for a variety of agricultural applications. It can be employed to identify different plant species and establish early warning systems for disease outbreaks. Furthermore, research efforts are focused on utilizing HS imaging to detect the chemical composition of plants, facilitating the monitoring of nutrient and water levels in irrigated wheat systems. HS cameras also offer the capability to identify stress caused by heavy metal accumulation in plants, providing an earlier and faster alternative to traditional post-harvest wet chemical methods.

Astronomy HS imaging plays a crucial role in astronomy, enabling the acquisition of spatially-resolved spectral images. Given the significance of spectral data as a diagnostic tool, the ability to obtain a spectrum for each pixel enhances the range of scientific inquiries that can be addressed in this field.

Ecology The imperative to monitor greenhouse gas emissions from various industrial sources, including coal and oil-fired power plants and landfills, has become increasingly critical in recent times. By leveraging the distinctive spectral signatures of methane (CH_4), researchers can compile a comprehensive list of the largest emitters on a global scale, facilitated by satellite-based monitoring systems with worldwide coverage.

Food processing In the food processing industry, the synergy of HS imaging with advanced software empowers digital sorters to detect and eliminate defects and foreign materials that would otherwise remain unseen by conventional camera and laser sorting systems. A key advantage of this approach is its non-destructive nature. Consequently, this technology holds the potential to address a multitude of persistent product quality issues, offering a promising solution for the industry.

Historical studies The Archimedes Palimpsest is a significant parchment codex that holds two invaluable works by Archimedes (the *Ostomachion* and the *Method of Mechanical Theorems*), which were previously believed to have been lost. It also contains the sole surviving original Greek edition of his work, *On Floating Bodies*. In the Middle Age, the manuscript was overwritten with a religious text. Shortly before its latest acquisition in 1998, counterfeit images were painted over portions of the text, obscuring their content. However, by analyzing 12-band multispectral (MS) images, researchers were able to unveil the concealed texts underneath the forged pictures, as well as decipher previously unreadable portions of the manuscript.

Waste sorting and recycling HS imaging plays a pivotal role in waste sorting and recycling by offering insights into the chemical composition of materials. This capabil-

ity proves invaluable in distinguishing between various substances with different fabric compositions, including the identification of natural, animal, and synthetic fibers. HS cameras can seamlessly integrate with computer vision systems, enabling the automated separation of plastic waste products and packaging by distinguishing between different types of plastics.

In this thesis, we do not focus on a specific application. Rather, we tackle research problems, namely image restoration and spectral unmixing, that are fundamental and pervasive across all applications within the realm of HS imaging.

1.2.4 Challenges

Acquistion and storage The primary drawbacks of HS imaging revolve around cost and complexity. To effectively analyze HS data, it requires powerful computing resources, sensitive detectors, and substantial data storage capabilities. The data storage requirement is especially notable because uncompressed HS cubes constitute large, multidimensional datasets, often exceeding hundreds of megabytes in size. These factors collectively contribute to the elevated costs associated with acquiring and processing HS data. Furthermore, researchers are confronted with the challenge of programming HS satellites to autonomously sift through data and transmit only the most critical images. This is essential, as the transmission and storage of such voluminous data can prove to be both technically challenging and financially burdensome.

Processing In recent years, a multitude of HS RS missions have been launched with the aim of advancing our comprehension of Earth and its environment. As the availability of RS data continues to expand, originating from satellite and airborne sensors on both large and global scales, traditional data processing and analysis tools have encountered limitations, unable to keep pace with the demands of the big data era. Consequently, data-driven signal and image processing, along with machine learning (ML) models, have drawn increasing attention among researchers within the RS community.

Complexity HS imaging often faces limitations in either spectral or spatial resolution. Moreover, a multitude of factors, including physical and chemical atmospheric effects, environmental conditions (e.g., soil interference, fluctuations in illumination, unpredictable shadows cast by clouds or buildings, variations in topography, and intricate noise patterns), have historically hindered traditional RS techniques, leading to predominantly qualitative analysis. However, this paradigm has shifted in recent times with the emergence of non-convex optimization techniques, such as deep learning. These advanced approaches have demonstrated their capability to effectively address increasingly complex scenarios.

Interpretability Nonetheless, a significant drawback consists in the inherent lack of interpretability coming from these deep networks, which has emerged as a major concern for decision-makers. Hence, there is a pressing need to develop techniques that are not only efficient but also interpretable, tailored to address the diverse spectrum of research problems specific to HS imaging.

This thesis focuses on two fundamental tasks related to HS data: image restoration and spectral unmixing. These tasks are presented in the following sections, accompanied by a comprehensive list of related works.

1.3 Image Restoration

Hyperspectral (HS) signals contain a wealth of information compared to RGB images. However, this advantage comes with the challenge of managing intricate degradations originating from multiple sources, including sparse noise patterns like stripes, as well as photon and thermal noise [Kerekes and Baum, 2003, Rasti et al., 2018]. Consequently, HS image denoising serves as a pivotal pre-processing step to increase image quality, facilitating subsequent tasks such as semantic segmentation and spectral unmixing.

Another substantial challenge arises from the scarcity of extensive, high-quality ground-truth datasets and the sheer diversity of sensor types employed in HS imaging. This complexity poses significant hurdles in training ML models, such as convolutional neural networks (CNNs), for restoration tasks.

To navigate the scarcity of ground-truth data, the most successful approaches typically incorporate robust prior knowledge directly into the model architecture. This prior knowledge may manifest as low-rank representations of input patches [Fan et al., 2018, Gong et al., 2020, Rasti et al., 2017, Wang et al., 2021, Zhao and Yang, 2015], sparse coding techniques [Dabov et al., 2007, Fu et al., 2015, Gong et al., 2020], or leveraging image self-similarities [Maggioni et al., 2013, Peng et al., 2014, Zhuang and Bioucas-Dias, 2017]. These strategies have proven successful in the realm of RGB image processing [Buades et al., 2005] and are adapted to address the unique challenges posed by HS data.

We are now in shape to delve into the multitude of approaches employed for restoring HS data. Our journey begins with an examination of model-based techniques, followed by an introduction to data-driven methods. Subsequently, we will explore hybrid approaches and eventually focus on the more specific algorithm unrolling techniques.

1.3.1 Model-based approaches

HS denoising can be framed as an inverse problem, wherein the goal is to determine the model parameters that describe the noisy observation. Given a noisy observation $\mathbf{Y}$, our objective is, in fact, to recover the clean signal, denoted as $\mathbf{X}$, which has been degraded by an unknown forward operator F. This relationship can be expressed as:

$$\mathbf{Y} = F(\mathbf{X}). \tag{1.1}$$

To approach this problem, we introduce a model denoted as $\mathcal{M}$, characterized by parameters θ. This model allows us to approximate the inverse operator of F:

$$\mathbf{X} \approx \mathcal{M}_\theta(\mathbf{Y}). \tag{1.2}$$

Classical signal processing models for grayscale image denoising such as BM3D [Dabov et al., 2007] may be applied independently to each spectral band of HS signals. Not surprisingly, such an approach fails to capture relations between different channels and multi-band techniques have been shown to perform better. An extension of BM3D to volumetric data is proposed with BM4D [Maggioni et al., 2013], which is more relevant for HS imaging. GLF [Zhuang and Bioucas-Dias, 2017] builds upon the same principles as BM3D – namely, exploiting self-similarities by grouping 3D patches and collaborative filtering – while introducing subspace identification and data-adaptive filtering based on principal component analysis. GLF appears to be a very strong baseline, but is unfortunately very computationally demanding at test time. Tensor-based methods such as LLRT [Chang et al., 2017] are able to exploit the underlying low-rank structure of HS

signals [Fan et al., 2018, Rasti et al., 2017, Zhang et al., 2014] and have shown particularly effective when combined with a non-local image prior as in NGMeet [He et al., 2020]. Finally, other approaches adapt traditional image processing priors such as total variation [Yuan et al., 2014, Wang et al., 2021], or wavelet sparsity [Othman and Qian, 2006, Rasti et al., 2014] to HS denoising, but they tend to perform worse than GLF, LLRT, or NGMeet, see [Kong et al., 2020] for a survey on denoising techniques for HS imaging.

1.3.2 Data-driven approaches

Sparse coding and dictionary learning

Dictionary learning [Olshausen and Field, 1996] is an unsupervised learning technique consisting of representing a signal as a linear combination of a few elements from a learned dictionary, which has shown to be very effective for various image restoration tasks [Elad and Aharon, 2006, Mairal et al., 2007]. Several approaches have then combined dictionary learning and low-rank regularization. For instance, 3D patches are represented as tensors in [Peng et al., 2014] and are encoded by using spatial-spectral dictionaries [Tucker, 1966]. In [Zhao and Yang, 2015], 2D patches are extracted from the band-vectorized representation of the 3D HS data and sparsely encoded on a dictionary, while encouraging low-rank representations with a trace norm penalty on the reconstructed image. The low-rank constraint can also be enforced by designing the dictionary as the result of the matrix multiplication between spatial and spectral dictionaries learned by principal component analysis as in [Fu et al., 2015]. However, these methods typically compute sparse representations with an iterative optimization procedure, which may be computationally demanding at test time.

Deep learning

Like BM3D above, convolutional neural networks for grayscale image denoising (*e.g.*, DnCNN [Zhang et al., 2017]) may also be applied to each spectral band, which is of course sub-optimal. Because deep neural networks have been highly successful for RGB images with often low computational inference cost, there have been many attempts to design deep neural networks dedicated to HS denoising. For instance, to account for the large number of hyperspectral bands, several approaches based on CNNs are operating on sliding windows in the spectral domain, [Maffei et al., 2020, Shi et al., 2021a, Yuan et al., 2019], which allows training models on signals with different number of spectral bands, but the sliding window significantly increases the inference time. More precisely, attention layers are used in [Shi et al., 2021a], while more traditional CNNs are used in [Maffei et al., 2020], possibly with residual connections [Yuan et al., 2019]. Recently, an approach based on recurrent architecture was proposed in [Wei et al., 2020] to process signals with an arbitrary number of bands, achieving impressive results for various denoising tasks.

1.3.3 Hybrid approaches

SMDS-Net [Xiong et al., 2020] adopts a hybrid approach between sparse coding and deep learning models by adapting the RGB image restoration method of [Lecouat et al., 2020b] to HS images. SMDS-Net first denoises the input image with non-local means [Buades et al., 2005], then performs subspace projection in the spectral domain using HySime [Bioucas-Dias and Nascimento, 2008], before sparsely encoding 3D patches (cubes) with a trainable version of Tensor-based ISTA [Qi et al., 2016]. Although this method reduces

considerably the number of parameters in comparison to vanilla deep learning models, the spectral sliding window approach lacks interpretability since the same denoising procedure is applied across different bands, which may not suffer from the same level of noise.

1.3.4 Algorithm Unrolling

Algorithm unrolling, or deep unfolding, has found applications in recent years across a broad spectrum of domains, with computational imaging, including hyperspectral imaging, standing out as a prominent area of interest. Many computational imaging research problems revolve around solving inverse problems, where retrieving the original data or image from observed measurements can be highly complex. Traditional approaches to tackle these inverse problems have often followed a model-based strategy. Within this paradigm, various techniques have emerged, such as sparse coding and low-rank matrix pursuit, which aim to find parsimonious representations of the underlying data. Model-based techniques often require iterative approaches due to the scarcity of closed-form solutions. These iterative algorithms form a rich domain of research, providing a robust foundation and ample prospects for the application of algorithm unrolling techniques.

For instance, SMDS-Net [Xiong et al., 2020] harnesses the power of algorithm unrolling, a technique initially introduced in the seminal work of [Gregor and LeCun, 2010]. The primary aim of algorithm unrolling is to establish a connection between iterative algorithms and neural network architectures. Figure 1.2 offers a high-level visualization of this framework, wherein each iteration of the algorithm corresponds to a distinct layer within the network. Concatenating these layers effectively constructs a deep neural network. The process of passing data through this network is equivalent to executing the iterative algorithm for a finite number of iterations. Furthermore, the algorithm's parameters, including model parameters and regularization coefficients, directly translate into network parameters. Consequently, this network can be trained through backpropagation, leading to model parameters that are learned from real-world training datasets. This trained network can be naturally perceived as a parameter-optimized algorithm, effectively addressing the interpretability limitations commonly encountered in conventional neural networks.

LISTA for image denoising with dictionary learning

We now focus on the Learned Iterative Shrinkage and Thresholding Algorithm (LISTA) applied to image denoising with dictionary learning.

A classical approach introduced by [Elad and Aharon, 2006] for image denoising consists in considering the set of small overlapping image patches (*e.g.*, 8×8 pixels) from a noisy image, and compute a sparse approximation of these patches onto a learned dictionary. The clean estimates for each patch are then recombined to produce the full image.

Formally, let us consider a noisy image $\mathbf{y} \in \mathbb{R}^{c \times h \times h}$ with c channels and two spatial dimensions. We denote by $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ the n overlapping patches from $\mathbf{y}$ of size $c \times s \times s$, which we represent as vectors in $\mathbb{R}^m$ with $m = cs^2$. Assuming that a dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p] \in \mathbb{R}^{m \times p}$ is given – we will discuss later how to obtain a "good" dictionary – each patch $\mathbf{y}_i$ is processed by computing a sparse approximation:

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 , \tag{1.3}$$

where $\|.\|_1$ is the ℓ_1-norm, which is known to induce sparsity in the problem solution [Mairal et al., 2014], and $\boldsymbol{\alpha}_i$ is the sparse code representing the patch $\mathbf{y}_i$, while $\lambda > 0$

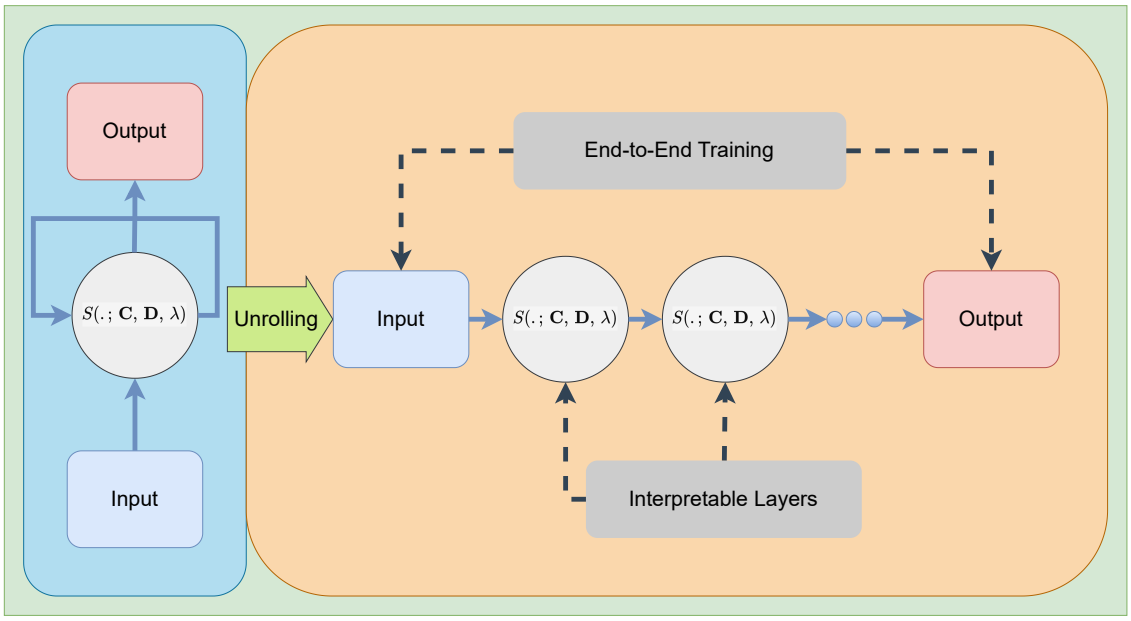


Figure 1.2: A high-level overview of algorithm unrolling inspired from [Monga et al., 2021]. Given the iterative algorithm ISTA pictured on the left, a corresponding deep network can be generated by cascading the algorithm iterations S. The iteration step S is executed a number of times, resulting in as many network layers. Each iteration S depends on the ISTA parameters $\mathbf{C}, \mathbf{D}$, and $\boldsymbol{\lambda}$, which are transferred into learnable weights of the network. Instead of determining these parameters through cross-validation and analytical derivations, $\mathbf{C}, \mathbf{D}$, and $\boldsymbol{\lambda}$ are learned from training data sets through end-to-end training. In this way, the resulting network can achieve better performance than the original iterative algorithm. In addition, the network layers naturally inherit interpretability from the iteration procedure.

is a regularization parameter controlling sparsity. Note that the ℓ_0-penalty, which counts the number of non-zero elements, could also be used, leading to a combinatorial problem whose solution is typically approximated by a greedy algorithm. After solving the n problems (1.3), each patch $\mathbf{y}_i$ admits a "clean" estimate $\mathbf{D}\boldsymbol{\alpha}_i$. Because each pixel belongs to several patches, the full restored image $\hat{\mathbf{x}}$ is obtained by averaging these estimates.

Finding a good dictionary can be achieved in various manners. In classical dictionary learning algorithms, $\mathbf{D}$ is optimized such that the sum of the loss functions (1.3) is as small as possible, see [Mairal et al., 2014] for a review. Adapting the dictionary with supervision is also possible [Mairal et al., 2012], as discussed next.

The proximal gradient descent method called Iterative Shrinkage and Thresholding Algorithm (ISTA) [Figueiredo and Nowak, 2003, Daubechies et al., 2004] is a popular method for solving the optimization problem (1.3) iteratively. In its simplest form, ISTA performs the following iterations:

$$\boldsymbol{\alpha}_i^{(t+1)} = S_\lambda \left[\boldsymbol{\alpha}_i^{(t)} + \eta \mathbf{D}^\top \left(\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i^{(t)} \right) \right], \quad t = 0, 1, \ldots, \tag{1.4}$$

where η is a positive parameter controlling the iteration step size, and $S_\lambda(.)$ is the soft-

thresholding operator defined elementwise as

$$S_\lambda(u) = \text{sign}(u) \cdot \max\{|u| - \lambda, 0\}. \tag{1.5}$$

In essence, ISTA combines a gradient step of $\left|\left|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\right|\right|_2^2$ with a projection onto the ℓ_1 ball.

LISTA corresponds to an adaptation of ISTA by recasting its iteration into a single network layer. This layer encompasses various analytic operations, including matrix-vector multiplication, summation, and soft-thresholding. These operations are reminiscent of those found in a neural network. Executing LISTA for T iterations is akin to cascading T such layers, effectively creating a T-layer deep network.

Within this unrolled network, various parameter substitutions can be made, such as introducing $\mathbf{C} \in \mathbb{R}^{m \times p}$ and turning the scalar regularization parameter λ into a vector $\boldsymbol{\lambda} \in \mathbb{R}^p$ to effectively transform (1.4) into

$$\boldsymbol{\alpha}_i^{(t+1)} = S_{\boldsymbol{\lambda}}\left[\boldsymbol{\alpha}_i^{(t)} + \mathbf{C}^\top\left(\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i^{(t)}\right)\right], \quad t = 0, 1, \ldots, \tag{1.6}$$

as done in [Simon and Elad, 2019, Lecouat et al., 2020b], where $S_{\boldsymbol{\lambda}}$ corresponds to the vectorized soft-thresholding operator defined in (1.5). Lastly, another dictionary $\mathbf{W}$ is used to obtain a clean estimate $\mathbf{W}\boldsymbol{\alpha}_i^{(T)}$ for each patch $\mathbf{y}_i$, where T is the number of LISTA steps. The reason for allowing a different dictionary $\mathbf{W}$ than $\mathbf{D}$ is to correct the potential bias due to ℓ_1-minimization. These substitutions expand the representation power of the unrolled network and provide a more generalized parametrization compared to the original ISTA.

Finally, the denoised image $\hat{\mathbf{x}}$ is reconstructed by averaging the patch estimates:

$$\hat{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{n} \mathbf{R}_i \mathbf{W}\boldsymbol{\alpha}_i^{(T)}, \tag{1.7}$$

where $\mathbf{R}_i$ is the linear operator that places the patch $\hat{\mathbf{x}}_i$ at position i in the image, and we assume – by neglecting border effects for simplicity – that each pixel admits the same number m of estimates.

The unrolled network's parameters, namely $\mathbf{C}$, $\mathbf{D}$, $\mathbf{W}$, and $\boldsymbol{\lambda}$, are optimized by training with a set of pairs of noisy/clean images in a supervised fashion. We remark that the estimate $\hat{\mathbf{x}}$ is obtained from a noisy image $\mathbf{y}$ by a sequence of operations that are differentiable almost everywhere, as in typical neural networks with rectified linear unit activation functions, which allows the use of backpropagation to update the network's parameters. A typical loss, which we optimize by stochastic gradient descent [LeCun et al., 2002], is then:

$$\min_{\mathbf{C},\mathbf{D},\mathbf{W},\boldsymbol{\lambda}} \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\left\|\hat{\mathbf{x}}(\mathbf{y}) - \mathbf{x}\right\|^2\right], \tag{1.8}$$

where $(\mathbf{x}, \mathbf{y})$ is a pair of clean/noisy images drawn from some training distribution from which we can sample, and $\hat{\mathbf{x}}(\mathbf{y})$ is the clean estimate obtain from (1.7), given the noisy image $\mathbf{y}$.

It is worth noting that empirical evidence suggests that the number of layers T in the trained LISTA can be significantly smaller than the number of iterations required for ISTA to converge to a solution that corresponds to a given input observation [Gregor and LeCun, 2010].

1.4 Spectral Unmixing

Spectral unmixing is a crucial processing technique in hyperspectral (HS) remote sensing (RS). The ability to separate and identify distinct materials within an image is made possible by the continuous spectra recorded by HS sensors. Employing endmembers, which represent distinctive spectral signatures of macroscopic materials, unmixing algorithms can disentangle the blended spectral data into its individual components. Nevertheless, the inherent challenges of low spatial resolution, multiple scattering, and intimate mixture typically yield measured spectra within a pixel that is a complex mixture of pure spectra from constituent materials, making unmixing a challenging task. Figure 1.3 illustrates the spectral mixing occurring at the pixel level within a scene captured by a HS camera.

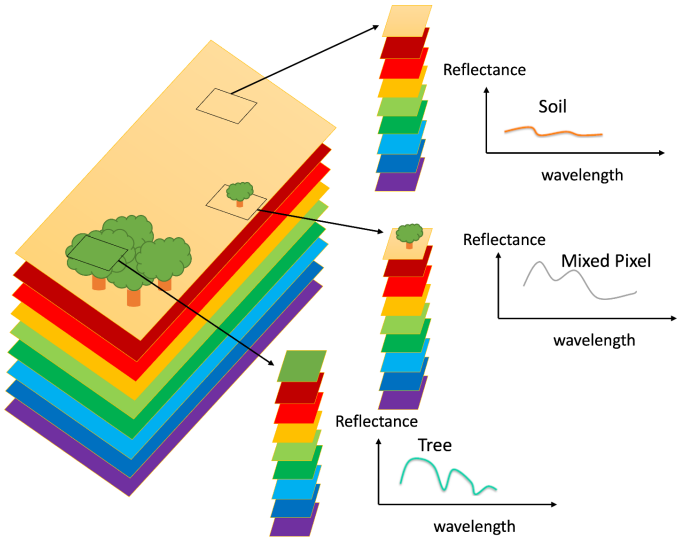


Figure 1.3: Illustration of pure and mixed pixels within an HS image from [Rasti et al., 2023a]. In this context, each pixel comprises a spectrum generated by concatenating spectral bands, with each spectrum corresponding to reflectance at a specific wavelength. The top and bottom pixels are considered pure, as they consist of a single macroscopic material. In contrast, the middle pixel represents a mixture of tree and soil components.

In the realm of HS RS, a mixing model serves as a representation of the observed spectral pixel, encapsulating the interplay between endmembers and their corresponding fractional abundances within the pixel's spatial domain. Unmixing, in turn, involves the estimation of these fractional abundances. This estimation can occur through various means, such as direct estimation or extraction of endmembers or reliance on a pre-existing library of endmembers. Additionally, unmixing may encompass the determination of the number of endmembers present in the scene. The nature of the mixing model can be either linear or nonlinear, depending on the interaction between incident light and the materials present within the scene or sample. This thesis mainly focuses on the linear mixing model (LMM), whereby the endmembers are assumed to be linearly mixed.

In the subsequent sections, we will delve deeper into the underlying assumptions of the LMM. We will then introduce various unmixing scenarios based on the level of prior knowledge available regarding the endmembers. Following that, our focus will shift to archetypal analysis as a model formulation for linear unmixing. Lastly, we will explore the different optimization strategies employed for conducting unmixing in HS imaging.

1.4.1 Linear mixing model

The LMM assumption holds true when each incident light ray interacts with a single material prior to reaching the HS sensor. It is a common and valid assumption in Earth Observation (EO) applications, especially in macroscopic scenarios. In these scenarios, the sensor's spatial resolution plays a pivotal role, as pixels may encompass multiple materials, leading to spectra that are mixtures of various substances.

Figure 1.4 provides a simplified illustration of the sensing process using a satellite equipped with a HS sensor. The sensor records a pixel containing three distinct materials: water, tree, and soil. At the sensor level, radiance is converted to reflectance, and atmospheric corrections are applied to compensate for atmospheric absorbance and light scattering effects. This correction results in reflectance values ranging from zero to one. Additionally, two physical constraints are imposed on the abundance fractions, namely the abundance non-negativity constraint (ANC) and the abundance sum-to-one constraint (ASC).

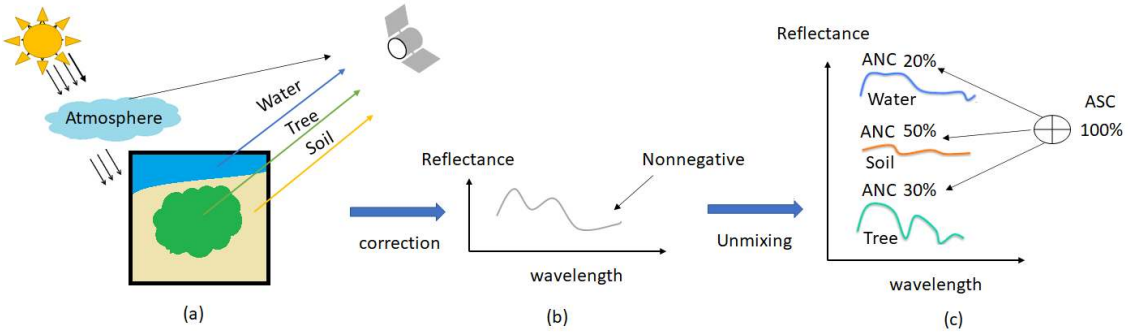


Figure 1.4: Illustration of the linear unmixing pipeline from [Rasti et al., 2023a]. (a) Sensing a mixed pixel; (b) Reflectance of the mixed pixel; (c) Schematic of linear unmixing.

Formally, assuming the sensor records p spectral bands, the LMM states that a p-dimensional pixel $\mathbf{y}$ is represented as a linear combination of the endmembers within the pixel. Let $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_r] \in \mathbb{R}^{p \times r}$ be a matrix containing r endmembers, then:

$$\mathbf{y} = \mathbf{E}\mathbf{a} + \mathbf{n}, \quad \text{s.t.} \quad \sum_{i=1}^{r} \mathbf{a}_i = 1, \ \mathbf{a}_i \geq 0, \ i = 1, 2, \ldots, r, \tag{1.9}$$

where $\mathbf{a}$ is a r-dimensional vector corresponding to the abundance fractions associated to each endmembers, and $\mathbf{n}$ is a p-dimensional random vector encompassing noise and model error.

Using matrix notations, we can represent the n pixels by $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{p \times n}$. The LMM now writes as

$$\mathbf{Y} = \mathbf{E}\mathbf{A} + \mathbf{N}, \quad \text{s.t.} \quad \mathbf{A} \geq 0, \ \mathbf{1}_r^\top \mathbf{A} = \mathbf{1}_n^\top, \tag{1.10}$$

where $\mathbf{N} \in \mathbb{R}^{p \times n}$ denotes the noise and model error, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{R}^{r \times n}$ is the abundance matrix containing the fractional abundances of each endmembers at every pixel, and $\mathbf{1}_d$ is a d-dimensional vector of ones.

In this configuration, the LMM assumes a known and fixed number of endmembers, denoted as r. In this context, the LMM can be interpreted as a low-rank mixture model, where the rank corresponds to the number of endmembers. Estimating the number of endmembers present in a scene is a challenging task that falls outside the scope of this

thesis. It is important to note that this information is not always required for unmixing. Alternatively, one can consider the redundant linear mixture model, which can be expressed as:

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{N} \quad \text{s.t.} \quad \mathbf{X} \geq 0. \tag{1.11}$$

Here $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_m] \in \mathbb{R}^{p \times m}$ represents a spectral library containing m endmembers, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ represents the unknown abundances to be estimated. It is worth noting that the dictionary $\mathbf{D}$ is often overcomplete, meaning that $p < m$, and it should be well-designed to include the endmembers present in the scene. When a well-designed dictionary is employed, $\mathbf{X}$ often exhibits sparsity properties because the pixels in the scene are usually composed of only a small number of dictionary elements, or atoms.

1.4.2 Unmixing scenarios

Unmixing techniques can be categorized into three main groups based on the level of prior knowledge about endmembers. Supervised and unsupervised (blind) approaches employ the low-rank mixing model. while semi-supervised unmixing relies on the redundant linear model. It is important to clarify that the terminology used for these configurations should not be confused with the supervision setups commonly encountered in classical machine learning. In this context, we exclusively refer to methods that do not have access to ground-truth abundances. Therefore, all learning-based approaches are inherently unsupervised when it comes to learning.

Supervised unmixing assumes that a set of endmembers, denoted as $\mathbf{E} \in \mathbb{R}^{p \times r}$, is known in advance, leaving only the task of estimating the abundances. In contrast, blind unmixing aims to simultaneously estimate both endmembers and abundances without prior knowledge other than assuming the number of endmembers, r, to be known. Semi-supervised unmixing leverages an existing library of endmembers, which should be sufficiently well-designed to include the materials present in the scene. When the dictionary $\mathbf{D} \in \mathbb{R}^{p \times m}$ is overcomplete ($p < m$), the goal in semi-supervised unmixing is often to achieve sparse estimates for the abundances, leading to its categorization as sparse unmixing. Figure 1.5 provides an illustration of the three main types of unmixing.

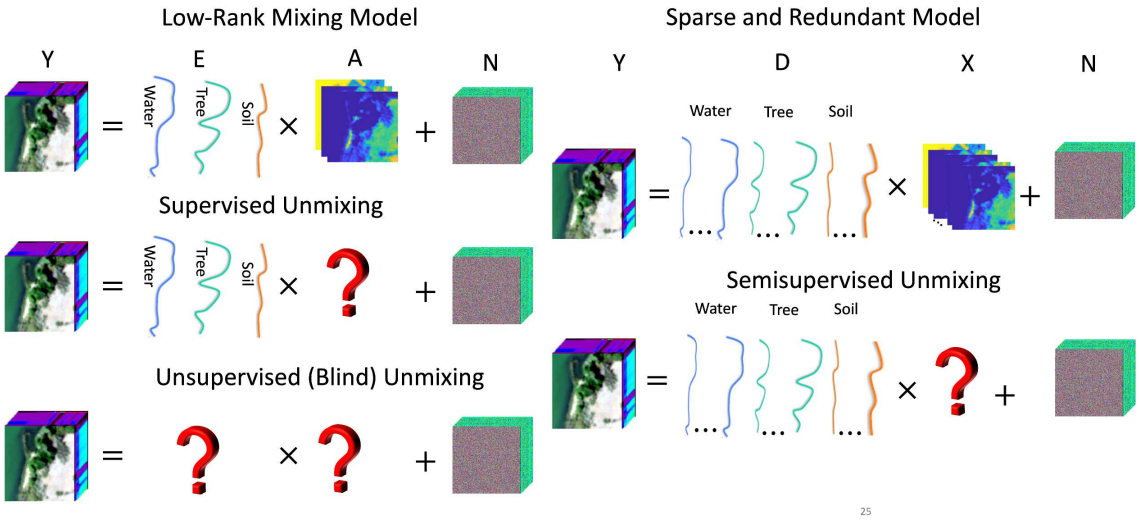


Figure 1.5: Illustration of different types of linear unmixing from [Rasti et al., 2023a].

Supervised unmixing

In supervised unmixing, we assume that the endmembers are known, and the goal is to estimate the abundance matrix $\mathbf{A} \in \mathbb{R}^{r \times n}$. These endmembers can be obtained through various means, including field or laboratory measurements, selection from an existing spectral library, or direct extraction from the data. However, selecting endmembers from a spectral library may not always yield optimal abundance estimations due to variations in the imaging setup.

Alternatively, endmembers can be directly extracted or estimated from the data points using geometrical approaches. This extraction process can be challenging because the recorded data may not contain pure pixels, which are pixels containing a single macroscopic material, for all the materials present in the scene. While many endmember extraction techniques are built on the assumption of having pure pixels, this assumption may not hold in practice.

It is important to note that we consider methods involving a sequential process of endmember extraction and subsequent abundance estimation to be categorized as supervised unmixing. In general, abundance estimation does not affect endmember estimation due to the order in the processing chain.

Blind unmixing

In contrast, blind unmixing encompasses approaches that simultaneously estimate both endmembers and abundances. This field of research comprises various paradigms and techniques. It is worth noting that blind unmixing methods often face challenges related to their inherent non-convexity. As a result, the optimization procedures used in these methods can be sensitive to initialization. Therefore, it is common practice to initialize the endmember estimates using a geometrical endmember extraction approach to provide a starting point for the optimization process.

Semi-supervised unmixing

Finally, semi-supervised unmixing relies on the availability of an endmember library, making the selection or construction of this library a critical step in the process. Blindly selecting a library without careful consideration and additional processing steps can lead to suboptimal results. Two major paradigms are commonly used in semi-supervised unmixing: Multiple Endmember Spectral and Mixture Analysis (MESMA [Roberts et al., 1998]) and sparse unmixing. MESMA was developed to address spectral variability by incorporating endmember variability into the library. In this setup, the spectral library is designed to represent the variability of the endmembers present. On the other hand, sparse unmixing, introduced in [Bioucas-Dias and Figueiredo, 2010], seeks a sparse solution based on the available library. In both paradigms, the library must adequately represent the materials present in the scene, meaning it should contain all the endmembers present. Various methods can be used to obtain such a library, including field or laboratory measurements, construction using observed data (often assuming the existence of pure pixels), and construction using physical models.

1.4.3 Archetypal Analysis

Now, we turn our attention to archetypal analysis (AA), a natural formulation for blind unmixing under the assumption of the LMM. AA essentially represents endmembers as

convex combinations of a few pixels found in the HS image. In real HS data, pure pixels, which contain a single material and are essential for traditional endmember extraction techniques, are often missing due to various factors such as atmospheric conditions, changes in illumination, and environmental influences. In the AA framework, not only are pixels viewed as linear combinations of estimated endmembers under the LMM, but the estimated endmembers themselves are modeled as convex combinations of pixels. AA, initially introduced by Cutler and Breiman in 1994 [Cutler and Breiman, 1994], is a specific case of non-negative matrix factorization (NMF) [Lee and Seung, 2000, Paatero and Tapper, 1994], a popular approach for blind unmixing. AA has the advantage to be more interpretable than NMF because the basis elements (*i.e.*, endmembers) are directly constructed from the data points (*i.e.*, pixels). In addition, since the estimated endmembers generally correspond to averaging the contributions of several pixels, the resulting spectra appear more robust to noise and spectral variability than pure pixel methods that only rely on a single pixel per endmember. However, AA usually suffers from a high data fitting error because the basis elements are constrained to be contained in the convex cone of the data points [De Handschutter et al., 2019].

Non-negative matrix factorization

Formally, within the LMM framework presented in (1.10), we investigate the blind unmixing scenario where both the mixing matrix, denoted as $\mathbf{E}$, and the abundance matrix, referred to as $\mathbf{A}$, are unknown. The only prior knowledge assumed is the number of endmembers present in the scene, denoted as r. As $\mathbf{E}$ represents the reflectance of the materials of interest across p spectral channels, it is essential that its elements are non-negative, satisfying $\mathbf{E} \geq 0$. Similarly, $\mathbf{A}$, which contains the abundances for each pixel in its columns, must also adhere to non-negativity constraints. Furthermore, each column of $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{R}^{r \times n}$ should sum to one, ensuring that it forms a valid probability distribution. This condition is equivalent to stating that each column of $\mathbf{A}$ belongs to the simplex Δ_r, defined as:

$$\Delta_r = \left\{ \mathbf{a} \in \mathbb{R}^r \text{ s.t. } \mathbf{a} \geq 0 \text{ and } \sum_{j=1}^{r} \mathbf{a}[j] = 1 \right\}. \tag{1.12}$$

The LMM (1.10) yields the classical optimization problem

$$\begin{aligned} \underset{\mathbf{E},\, \mathbf{A}}{\arg\min} \quad & \frac{1}{2} \|\mathbf{Y} - \mathbf{E}\mathbf{A}\|_F^2, \\ \text{s.t.} \quad & \mathbf{E} \geq 0, \\ & \mathbf{a}_i \in \Delta_r \text{ for } 1 \leq i \leq n, \end{aligned} \tag{1.13}$$

which is a variant of NMF. In essence, NMF involves decomposing a matrix representing the HS signal into the product of two matrices, both containing non-negative entries. One matrix represents the endmembers, and the other represents the abundances for each pixel, typically with the constraint that the abundances sum to one. Various NMF variants have been proposed for blind unmixing.

For instance, the Minimum Volume Constrained Non-Negative Matrix Factorization (MVC-NMF) method [Miao and Qi, 2007] incorporates a minimum volume term for endmembers, eliminating the pure pixel assumption. Minimum Dispersion Constraint NMF (MiniDisCo) [Huck et al., 2010] introduces a regularization function known as dispersion

to encourage endmembers with minimum variance, thus preventing degenerate solutions and enhancing unmixing robustness, particularly for flat spectra. Another framework proposed in [Zhuang et al., 2019] combines a data fidelity term with a minimum volume regularization term for endmembers, offering flexibility in the choice of regularization form. The authors also present an approach for automatic regularization parameter tuning.

Furthermore, various regularization functions have been designed for abundances in NMF, as seen in [Zymnis et al., 2007, Yang et al., 2010, Yao et al., 2019]. It is important to note that standard NMF employs a least-squares objective function as in (1.13), which may not be optimal for handling noise and outliers in the data. Consequently, alternative approaches like general loss-based NMF [Peng et al., 2020] and self-paced NMF [Peng et al., 2021] have been proposed. These methods employ different optimization criteria that can enhance robustness when dealing with noisy data and outliers.

Modeling assumptions

The AA formulation we consider introduces a constraint that enforces endmembers to be expressed as convex combinations of the pixels within the HS image $\mathbf{Y}$. In other words, there exists a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r] \in \mathbb{R}^{n \times r}$ such that $\mathbf{E} = \mathbf{YB}$, and the columns of $\mathbf{B}$ are constrained to lie within the simplex Δ_n. This results in the following optimization problem:

$$\begin{aligned} \underset{\mathbf{B},\,\mathbf{A}}{\arg\min} \quad & \frac{1}{2}\left\|\mathbf{Y} - \mathbf{YBA}\right\|_F^2, \\ \text{s.t.} \quad & \mathbf{b}_j \in \Delta_n \text{ for } 1 \leq j \leq r, \\ & \mathbf{a}_i \in \Delta_r \text{ for } 1 \leq i \leq n. \end{aligned} \tag{1.14}$$

In the context of the LMM (1.10), we observe that individual pixels within the HS image are represented as linear combinations of archetypes, which are essentially the endmembers. Notably, these archetypes are expressed as convex combinations of individual pixels found in the data under the AA formulation (1.14). This unique characteristic enhances model interpretability, as the estimated endmembers can be directly related to the pixels present in the scene.

Building upon the interpretability of AA, [Zhao and Yang, 2015] introduces a kernelized variant of AA. This kernelized AA offers enhanced modeling flexibility, albeit at the expense of an additional hyperparameter – the bandwidth of the Radial Basis Function (RBF) kernel functions. Importantly, they adopt the relaxation form from [Mørup and Hansen, 2012] to handle scenarios where endmembers lie outside the convex hull of the data.

Drawing inspiration from the robust AA formulation introduced in [Chen et al., 2014], which incorporates the Huber loss to mitigate the impact of noise and outliers, [Sun et al., 2017] proposed a Robust Kernel Archetypal Analysis (RKADA) method for blind HS unmixing. Their approach refines the standard AA formulation (1.14) by introducing a binary sparse constraint on pixel contributions. Consequently, this method ensures that each endmember corresponds to actual pixels rather than a sparse linear combination of all pixels.

In a recent development, [Xu et al., 2022] put forth an ℓ_1 sparsity-constrained AA algorithm aimed at increasing the sparsity of the abundances. Lastly, the Near-Convex Archetypal Analysis (NCAA) method [De Handschutter et al., 2019] was introduced to combine the strengths of both AA and NMF. NCAA requires endmembers to be linear

combinations, rather than convex combinations, of the pixels, offering a unique approach to the unmixing problem.

Extension to semi-supervised unmixing

Both NMF (1.13) and AA (1.14) offer viable solutions for addressing blind unmixing scenarios. However, the semi-supervised unmixing setup, relying on the redundant LMM (1.11), requires a different approach.

In conventional sparse unmixing, the endmember library $\mathbf{D}$ is assumed to be fixed, and the primary focus lies on estimating the (redundant) abundances. Nevertheless, even with a carefully pruned and well-designed spectral library, it is challenging to perfectly represent the spectral signatures of materials in real-world datasets. Several factors, including noise, atmospheric effects, illumination variations, and intrinsic material variations, can affect endmembers and induce scaling factors compared to those provided by the library.

To address this challenge, we build upon the AA formulation by assuming that the endmembers present in the scene can be modeled as a convex combination of the library spectra. This assumption introduces an additional matrix $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_r] \in \mathbb{R}^{m \times r}$, where the endmembers in the scene can be expressed as $\mathbf{DB}$. It is important to note that one additional assumption is required in this formulation: the number of endmembers of interest, r, is presumed to be known.

This formulation leads to the following optimization problem:

$$\begin{aligned} \underset{\mathbf{B},\,\mathbf{A}}{\arg\min} \quad & \frac{1}{2}\left\|\mathbf{Y} - \mathbf{DBA}\right\|_F^2, \\ \text{s.t.} \quad & \mathbf{b}_j \in \Delta_m \text{ for } 1 \leq j \leq r, \\ & \mathbf{a}_i \in \Delta_r \text{ for } 1 \leq i \leq n. \end{aligned} \tag{1.15}$$

In this setup, $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{r \times n}$ represents the abundances of the scene's endmembers, while $\mathbf{B}$ introduces a flexibility that allows the scene's endmembers to be a combination of library spectra. This approach accommodates the complexities and variations introduced by real-world data, providing a more accurate representation of the materials present in the scene. It is worth noting that the redundant abundances can easily be recovered by considering $\mathbf{BA} \in \mathbb{R}^{m \times n}$.

1.4.4 Optimization

Up until now, we have delved into the modeling assumptions within the AA framework without focusing on the practical aspect of solving the resulting optimization problems. Solving (1.14) and (1.15) poses a challenge since the objective function is not jointly convex in both $(\mathbf{B}, \mathbf{A})$. However, it exhibits convexity with respect to one variable when the other is held fixed, as demonstrated in [Mørup and Hansen, 2012].

Given this characteristic, it is natural to consider an alternating minimization scheme between $\mathbf{B}$ and $\mathbf{A}$. This approach is known to asymptotically converge to a stationary point of the problem, as shown in [Bertsekas, 1997]. However, due to the non-convex nature of the objective function, the choice of the optimization algorithm plays a crucial role. Different optimization procedures may lead to distinct stationary points, which can vary in terms of statistical estimation quality. This phenomenon, often referred to as "implicit bias", has garnered significant attention in machine learning, particularly in the context of

deep learning models [Pesme et al., 2021], and it may be important in HS unmixing as well, as detailed in Chapter 3.

As highlighted by [Cutler and Breiman, 1994], when fixing all variables but a column $\mathbf{a}_i$ of $\mathbf{A}$ and minimizing with respect to $\mathbf{a}_i$, the problem becomes a quadratic problem (QP) defined as:

$$\min_{\mathbf{a}_i \in \Delta_r} \|\mathbf{y}_i - \mathbf{Z}\mathbf{a}_i\|_2^2 , \tag{1.16}$$

where $\mathbf{Z}$ is equal to $\mathbf{YB}$ or $\mathbf{DB}$ depending on the unmixing setup. Similarly, a QP can be obtained when fixing all variables but one column $\mathbf{b}_j$ of $\mathbf{B}$, as shown in [Chen et al., 2014]. The primary challenge here lies in finding efficient methods to solve QP subject to simplex constraints. Therefore we focus on finding an algorithm for solving:

$$\min_{\mathbf{a} \in \Delta_r} \left[f(\mathbf{a}) = \|\mathbf{y} - \mathbf{Z}\mathbf{a}\|_2^2 \right] , \tag{1.17}$$

which is a smooth (least-squares) optimization problem with a simplicial constraint. Now, we will outline two efficient approaches to address (1.17), with the first one optimized for CPU and the second one designed for GPU acceleration.

Active-set algorithm

While it is possible to employ general QP solvers, achieving significantly faster convergence becomes feasible by devising dedicated algorithms that can harness the inherent "sparsity" of the solution in (1.17) [Bach et al., 2012].

In the work by [Chen et al., 2014], the authors introduce an active-set algorithm [Nocedal and Wright, 1999] designed to capitalize on the sparsity of the solution. They observe that, at the optimum, typically only a small subset A of the variables will be non-zero. Active-set algorithms take an aggressive approach to leverage this property. Given a current estimate $\mathbf{a}$ within Δ_r at a particular iteration, they define a subset $\mathrm{A} = \{j \text{ s.t. } \mathbf{a}[j] > 0\}$ and find a direction $\mathbf{q} \in \mathbb{R}^r$ by solving the reduced problem defined as:

$$\min_{\mathbf{q} \in \mathbb{R}^r} \|\mathbf{y} - \mathbf{Z}(\mathbf{a} + \mathbf{q})\|_2^2 \quad \text{s.t.} \quad \sum_{j=1}^{r} \mathbf{q}[j] = 0 \text{ and } \mathbf{q}_{\mathrm{A}^C} = 0, \tag{1.18}$$

where A^C represents the complement of A within the index set $\{1, \ldots, r\}$. Subsequently, they obtain a new estimate $\mathbf{a}' = \mathbf{a} + \gamma\mathbf{q}$ by moving $\mathbf{a}$ in the direction of $\mathbf{q}$ and ensuring that $\mathbf{a}'$ remains within Δ_r through the choice of $\gamma \in [0, 1]$. The algorithm iteratively updates the set A until it converges to an optimal solution within Δ_r. This strategy is elaborated in Algorithm 2 of [Chen et al., 2014] and we will describe in Chapter 4 how to leverage this algorithm in the context of semi-supervised unmixing.

Entropic gradient descent

An alternative approach for efficiently addressing (1.17) is to utilize an optimization method known as entropic gradient descent. This approach exhibits superior theoretical convergence properties compared to projected gradient descent when optimizing over the simplex [Beck and Teboulle, 2003]. Notably, entropic descent eliminates the need for orthogonal projections onto the simplex and the complexities associated with active-set rules. As a result, it becomes feasible to harness the computational power of modern GPUs.

As noted in [Beck and Teboulle, 2003], the entropic descent algorithm (EDA) is simply a gradient descent method with a particular choice of a Bregman-like distance [Bregman, 1967] generated by a specific function, namely the negative entropy. As explained in [Teboulle, 1992], the choice of an appropriate distance-like function tailored to the geometry of the constraints, here the simplex, provides theoretical benefits in terms of convergence rates.

Formally, the negative entropy function h is defined as follows: for $\mathbf{a} \in \mathbb{R}^r$,

$$h(\mathbf{a}) = \sum_{j=1}^{r} \mathbf{a}_j \ln(\mathbf{a}_j) \quad \text{if } \mathbf{a} \in \Delta_r, \quad +\infty \text{ otherwise,} \tag{1.19}$$

with the convention that $0 \ln 0 \equiv 0$.

h exhibits desirable properties, such as convexity on Δ_r. This enables us to consider D_h, the Bregman divergence [Bregman, 1967] with respect to h, defined, for $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^r$:

$$D_h(\mathbf{u}, \mathbf{v}) = h(\mathbf{u}) - h(\mathbf{v}) - \nabla h(\mathbf{v})^\top (\mathbf{u} - \mathbf{v}), \tag{1.20}$$

which is also called the Kullback-Leibler divergence. By convexity of h, we naturally have $D_h(\mathbf{u}, \mathbf{v}) \geq 0$.

To solve (1.17), we now consider the following iterates, given $\mathbf{a}^k \in \Delta_r$,

$$\mathbf{a}^{k+1} \leftarrow \arg\min_{\mathbf{a} \in \Delta_r} \left\{ \nabla f(\mathbf{a}^k)^\top (\mathbf{a} - \mathbf{a}^k) + \frac{1}{\eta^k} D_h(\mathbf{a}, \mathbf{a}^k) \right\}, \tag{1.21}$$

where $\nabla f(\mathbf{a}^k)$ denotes the gradient of f, which is convex and Lipschitz continuous, at $\mathbf{a}^k \in \Delta_r$.

If D_h was simply a squared Euclidean norm, we would recover a projected gradient descent algorithm. Instead, by using the Bregman distance function D_h induced from the negative entropy (1.19), we obtain the entropic descent method. Here D_h measures the distance between two vectors in Δ_r. As such, the next iterate $\mathbf{a}^{k+1}$ should aim for the optimal balance between taking a gradient step and moving the least from the current iterate $\mathbf{a}^k$ according to the geometry induced by h, with η^k controlling this trade-off. The negative entropy h yields explicit steps that effectively enforce the simplicial constraints. As demonstrated in Chapter 3, it is possible to show that the update (1.21) is equivalent to the following one, for all j in $\{1, \ldots, r\}$,

$$\mathbf{a}_j^{k+1} = \frac{\mathbf{a}_j^k e^{-\eta^k \nabla f(\mathbf{a}^k)_j}}{\sum_{l=1}^{r} \mathbf{a}_l^k e^{-\eta^k \nabla f(\mathbf{a}^k)_l}}, \tag{1.22}$$

where $\mathbf{a}_j^k$ is the j-th entry of the vector $\mathbf{a}^k$ and similarly, $\nabla f(\mathbf{a}^k)_j$ is the j-th entry of $\nabla f(\mathbf{a}^k)$.

It is thus easy to see that the iterates $(\mathbf{a}^k)_{k \in \mathbb{N}}$ stay in the simplex Δ_r, and it is possible to show (see [Beck and Teboulle, 2003]) that the sequence $(\mathbf{a}^k)_{k \in \mathbb{N}}$ converges to the set of solutions of (1.17) with the appropriate step sizes η^k. Interestingly, the update (1.22) can be implemented efficiently by using the softmax function, assuming the entries of $\mathbf{a}^k$ are positive:

$$\mathbf{a}^{k+1} = \text{softmax}\left(\log(\mathbf{a}^k) - \eta^k \nabla f(\mathbf{a}^k)\right), \tag{1.23}$$

where $\log(\mathbf{a}^k)$ is the vector carrying the logarithm of each entry of $\mathbf{a}^k$. This update immediately suggests a high compatibility with GPUs and we will describe in Chapter 3 how to leverage this approach to tackle blind unmixing.

2

Model-based Deep Learning for Hyperspectral Image Restoration

Chapter abstract: Hyperspectral imaging offers new perspectives for diverse applications, ranging from the monitoring of the environment using airborne or satellite remote sensing, precision farming, food safety, planetary exploration, or astrophysics. Unfortunately, the spectral diversity of information comes at the expense of various sources of degradation, and the lack of accurate ground-truth "clean" hyperspectral signals acquired on the spot makes restoration tasks challenging. In particular, training deep neural networks for restoration is difficult, in contrast to traditional RGB imaging problems where deep models tend to shine. In this chapter, we advocate instead for a hybrid approach based on sparse coding principles that retains the interpretability of classical techniques encoding domain knowledge with handcrafted image priors, while allowing to train model parameters end-to-end without massive amounts of data. We show on various denoising benchmarks that our method is computationally efficient and significantly outperforms the state of the art.

The source code is freely available at https://github.com/inria-thoth/T3SC.

The chapter is based on the following publication:

T. Bodrito*, A. Zouaoui*, J. Chanussot, and J. Mairal. A trainable spectral-spatial sparse coding model for hyperspectral image restoration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021

*equal contributions

2.1 Introduction

Hyperspectral (HS) imaging enables measurements of the electromagnetic spectrum of a scene on multiple bands (typically about a hundred or more), which offers many perspectives over traditional color RGB imaging. For instance, the high-dimensional information present in a single pixel is sometimes sufficient to identify the signature of a particular material, which is of course infeasible in the RGB domain. Not surprisingly, HS imaging is then of utmost importance and has a huge number of scientific and technological applications such as remote sensing [Bioucas-Dias et al., 2013, Goetz, 2009, Manolakis et al., 2016], quality evaluation of food products [Elmasry et al., 2012, Feng and Da-Wen Sun, 2012, Liu et al., 2017], medical imaging [Akbari et al., 2012, Fei, 2020, Lu and Fei, 2014], agriculture and forestry [Adão et al., 2017, Lu et al., 2020, Mahesh et al., 2015], microscopy imaging in biology [Gowen et al., 2015, Studer et al., 2012], or exoplanet detection in astronomy [Gonzalez et al., 2018].

In this chapter, we propose a fully interpretable machine learning model for HS images that may be seen as a hybrid approach between deep learning techniques, where parameters can be learned end-to-end with supervised data, and classical methods that essentially rely on image priors. Since designing an appropriate image prior by hand is very hard, our goal is to benefit from deep learning principles (here, differentiable programming [Baydin et al., 2018]) while encoding domain knowledge and physical rules about HS data directly into the model architecture, which we believe is a key to develop robust approaches that do not require massive amounts of training data.

More precisely, we introduce a novel trainable spectral-spatial sparse coding model with two layers, T3SC, which performs the following operations: (i) The first layer decomposes the spectrum measured at each pixel as a sparse linear combination of a few elements from a learned dictionary, thus performing a form of linear spectral unmixing per pixel, where dictionary elements can be seen as basis elements for spectral responses of materials present in the scene. (ii) The second layer builds upon the output of the first one, which is represented as a two-dimensional feature map, and sparsely encodes patches on a dictionary in order to take into account spatial relationships between pixels within small receptive fields. To further reduce the number of parameters to learn and leverage classical prior knowledge about spectral signals [Wang et al., 2021], we also assume that the dictionary elements admit a low-rank structure – that is, dictionary elements are near separable in the space and spectrum domains, as detailed later. Even though dictionary learning has been originally introduced for unsupervised learning [Mairal et al., 2014, Olshausen and Field, 1996], we adopt an unrolled optimization procedure inspired by the LISTA algorithm [Gregor and LeCun, 2010], which has been very successful in imaging problems for training sparse coding models from supervised data [Lecouat et al., 2020b, Lecouat et al., 2020a, Simon and Elad, 2019, Xiong et al., 2020].

Our motivation for adopting a two-layer model is to provide a shared architecture for different HS sensors, which often involve a different number of bands with different spectral responses. Our solution consists of learning sensor-specific dictionaries for the first layer, while the dictionary of second layer is shared across modalities. This allows training simultaneously on several HS signals, the first layer mapping input data to a common space, before processing data by the second layer.

We experimentally evaluate our HS model on standard denoising benchmarks, showing a significant improvement over the state of the art (including deep learning models and more traditional baselines), while being computationally very efficient at test time. Perhaps more important than pure quantitative results, we believe that our work also draws interesting

conclusions for machine learning. First, by encoding prior knowledge within the model architecture directly, we obtain models achieving excellent results with a relatively small number of parameters to learn, a conclusion also shared by [Lecouat et al., 2020a, Lecouat et al., 2020b] for RGB imaging; nevertheless, the effect is stronger in our work due to the scarcity of training data for HS denoising and the difficulty to train deep learning models for this task. Second, we also show that interpretable architectures are useful: our model architecture can adapt to different noise levels per band and modify the encoding function at test time in a principled manner, making it well suited for solving blind denoising problems that are crucial for processing HS signals.

2.2 Method

Building on Section 1.3 which presents some preliminaries on sparse coding and algorithm unrolling for HS image restoration, we are now in shape to introduce a trainable layer encoding both sparsity and low-rank principles.

2.2.1 A Trainable Low-Rank Sparse Coding Layer

Spatial-Spectral Representation

As shown in [Chakrabarti and Zickler, 2011, Fu et al., 2015], HS patches can be well reconstructed by using only a few basis elements obtained by principal component analysis. The authors further decompose these into a Cartesian product of separate spectral and spatial dictionaries. In this chapter, we adopt a slightly different approach, where we consider a single dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_p]$ in $\mathbb{R}^{m \times p}$ as in Section 1.3 with $m = cs^2$, but each element may be seen as a matrix of size $c \times s^2$ with low-rank structure. More precisely, we enforce the following representation:

$$\forall j \in 1, \dots, p, \quad \mathbf{d}_j = \text{vec}\left(\mathbf{U}_j \times \mathbf{V}_j\right), \tag{2.1}$$

where $\mathbf{U}_j$ is in $\mathbb{R}^{s^2 \times r}$, $\mathbf{V}_j$ is in $\mathbb{R}^{r \times c}$, r is the desired rank of the dictionary elements, and $\text{vec}(.)$ is the operator that flattens a matrix into a vector. The hyperparameter r is typically small with $r = 1, 2$ or 3. When $r = 1$, the dictionary elements are said to be separable in the spectral and spatial domains, which we found to be a too stringent condition to achieve good reconstruction in practice.

The low-rank assumption allows us to build models with a reduced number of parameters, while encoding natural assumption about the data directly in the model architecture. Indeed, whereas a classical full-rank dictionary $\mathbf{D}$ admits cs^2p parameters, the decomposition (2.1) yields dictionaries with $(s^2 + c)rp$ parameters only. Matrices $\mathbf{C}$ and $\mathbf{W}$ are parametrized in a similar manner.

Convolutional variant and implementation tricks

Whereas traditional sparse coding reconstructs local signals (patches) independently according to the iterations (1.6), another variant called convolutional sparse coding (CSC) represents the whole image by a sparse linear combination of dictionary elements placed at every possible location in the image[Simon and Elad, 2019]. From a mathematical point of

view, the reconstruction loss for computing the codes $\boldsymbol{\alpha}_i$ given an input image $\mathbf{y}$ becomes

$$\min_{\{\boldsymbol{\alpha}_i \in \mathbb{R}^p\}_{i=1,\ldots,n}} \frac{1}{2} \left\| \mathbf{y} - \frac{1}{m} \sum_{i=1}^{n} \mathbf{R}_i \mathbf{D} \boldsymbol{\alpha}_i \right\|^2 + \sum_{i=1}^{n} \sum_{j=1}^{p} \boldsymbol{\lambda}_j |\boldsymbol{\alpha}_i[j]|. \tag{2.2}$$

An iterative approach for computing these codes can be obtained by a simple modification of (1.6) consisting of replacing the quantity $\mathbf{D}\boldsymbol{\alpha}_i^{(t)}$ by the i-th patch of the reconstructed image $\frac{1}{m}\sum_{i=1}^{n} \mathbf{R}_i \mathbf{D} \boldsymbol{\alpha}_i^{(t)}$. All of these operations can be efficiently implemented in standard deep learning frameworks, since the corresponding operations corresponds to a transposed convolution with $\mathbf{D}$, followed by convolution with $\mathbf{C}$, see[Simon and Elad, 2019] for more details. In this chapter, we experimented with the CSC variant (2.2) and SC one (1.3), both with low-rank dictionaries, which were previously described. We observed that CSC was providing slightly better results and was thus adopted in our experiments. Following[Lecouat et al., 2020b], another implementation trick we use is to consider a different $\boldsymbol{\lambda}_j$ parameter per dictionary element, which slightly increases the number of parameters, while allowing to learn with a weighted ℓ_1-norm in (2.2).

2.2.2 The Two-Layer Sparse Coding Model with Sensor-Specific Layer

One of the main challenge in hyperspectral imaging is to train a model that can generalize to several types of sensors, which typically admit different number of spectral bands. Whereas learning a model that is tuned to a specific sensor is perfectly acceptable in many contexts, it is often useful to learn a model that is able to generalize across different types of HS signals. To alleviate this issue, several strategies have been adopted such as (i) projecting signals onto a linear subspace of fixed dimension, with no guarantee that representations within this subspace can be comparable between different signals, or (ii) processing input data using a sliding window across the spectral domain.

In this chapter, we address this issue by learning a two-layer model, presented in Figure 2.1, where the first layer is tuned to a specific sensor, whereas the second layer could be generic. Note that the second layer carries most of the model parameters (about $20\times$ more than in the first layer in our experiments). Formally, let us denote by $\boldsymbol{\alpha}$ in $\mathbb{R}^{p\times h\times w}$ the sparse encoding of an input tensor $\mathbf{y}$ in $\mathbb{R}^{c\times h\times w}$ as previously described. A sparse coding layer Φ naturally yields an encoder and a decoder such that:

$$\Phi^{enc} : \mathbf{y} \mapsto \boldsymbol{\alpha}, \quad \text{and} \quad \Phi^{dec} : \boldsymbol{\alpha} \mapsto \frac{1}{n} \sum_{i=1}^{n} \mathbf{R}_i \mathbf{W} \boldsymbol{\alpha}_i. \tag{2.3}$$

Given a noisy image $\mathbf{y}$, the denoising procedure described in the previous section with one layer can be written as

$$\hat{\mathbf{x}}(\mathbf{y}) = \Phi^{dec} \circ \Phi^{enc}(\mathbf{y}).$$

Then, a straightforward multilayer extension of the procedure may consist of stacking several sparse coding layers $\Phi_1, \ldots, \Phi_L$ together to form a multilayer sparse coding denoising model:

$$\hat{\mathbf{x}}(\mathbf{y}) = \Phi_1^{dec} \circ \cdots \circ \Phi_L^{dec} \circ \Phi_L^{enc} \circ \cdots \circ \Phi_1^{enc}(\mathbf{y}).$$

The model we propose is composed of two layers, as shown in Figure 2.1. The first layer encodes spectrally the input HS image, meaning that it operates on 1×1 patches, whereas the second layer encodes both spectrally and spatially the output of the first layer.

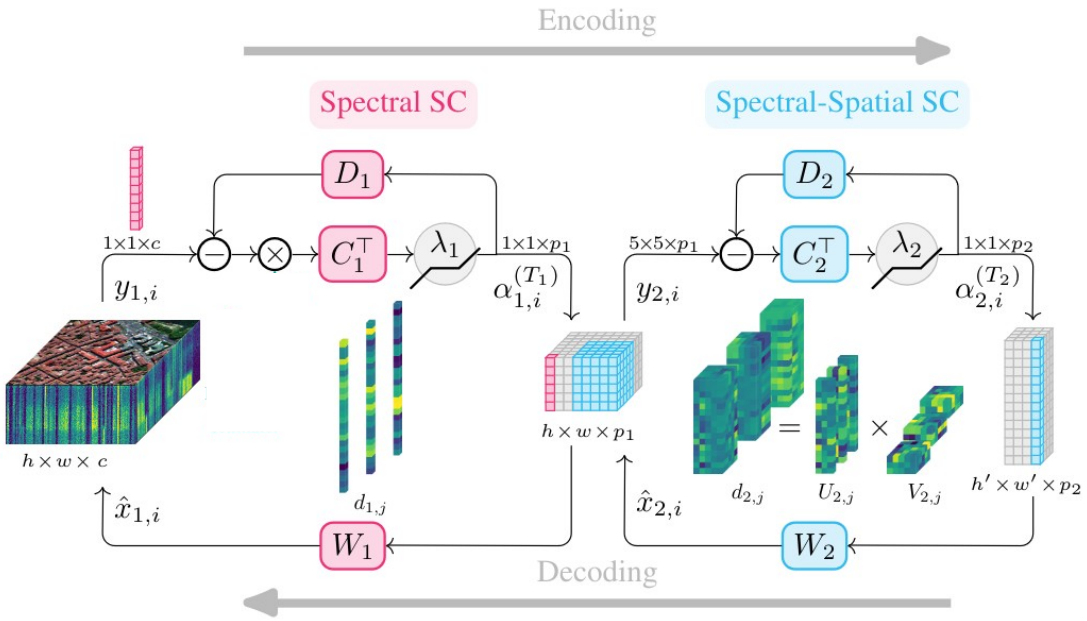


Figure 2.1: Architecture of T3SC: we propose a two-layer sparse coding model which is end-to-end trainable. The first layer performs a sensor-specific spectral decomposition, while the second layer encodes both spectral and spatial information.

Table 2.1: Simplified comparison between learning-free and learning-based approaches.

	Data req.	*training*	*inference*	*adapt. to new data*	*complex noise*
learning-free	no req.	no training	slow	easy	poor
learning-based	clean data	slow	fast	complicated	good perf.

2.3 Experiments

We now present various experiments to demonstrate the effectiveness of our approach for HS image denoising, but first, we discuss the difficulty of defining the state of the art in this field. We believe indeed that it is not always easy to compare learning-free from approaches based on supervised learning. These two classes of approaches have very different requirements/characteristics, making one class more relevant than the other one in some scenarios, and less in others. Table 2.1 summarizes their characteristics, displaying advantages and drawbacks of both approaches.

Benchmarked models

Keeping in mind the previous dichotomy, we choose to compare our method to traditional methods such as bandwise BM3D [Dabov et al., 2007] (implementation based on [Mäkinen et al., 2019, Mäkinen et al., 2020]), BM4D [Maggioni et al., 2013], GLF [Zhuang and Bioucas-Dias, 2017], LLRT [Chang et al., 2017], NGMeet [He et al., 2020]. We also included deep learning models in our benchmark such as HSID-CNN [Yuan et al., 2019], HSI-SDeCNN [Maffei et al., 2020], 3D-ADNet [Shi et al., 2021a], SMDS-Net [Xiong et al., 2020] and QRNN3D [Wei et al., 2020]. Results of HSID-CNN, HSI-SDeCNN and 3D-ADNet on Washington DC Mall (available in the Appendix) are taken directly from the corresponding papers, as the train/test split is the same. Otherwise, the results were obtained by running the code obtained directly from the authors, except for SMDS-Net,

where our implementation turned out to be slightly more effective. Note that the same architecture for our model was used in all our experiments (see Appendix).

Datasets

We evaluate our approach on two datasets with significantly different properties.

- *ICVL* [Arad and Ben-Shahar, 2016] consists of 204 images of size 1392×1300 with 31 bands. We used 100 images for training and 50 for testing as in [Wei et al., 2020] but with a different train/test split ensuring that similar images – *e.g.*, picture from the same scene – are not used twice.

- *Washington DC Mall* is perhaps the most widely used dataset[1] for HSI denoising and consists of a high-quality image of size 1280×307 with 191 bands. Following[Shi et al., 2021a], we split the image into two sub-images of size 600×307 and 480×307 for training and one sub-image of size 200×200 for testing. Even though the test image does not overlap with train images, they nevertheless share common characteristics. Interestingly, the amount of training data is very limited here.

Specific experiments were also conducted with the datasets APEX[Itten et al., 2008], Pavia[2], Urban[Rickard et al., 1993] and CAVE[Yasuma et al., 2010], which appear in the supplementary material.

Normalization

Before denoising, HSI images are normalized to $[0, 1]$. For remote sensing datasets, we pre-compute the 2^{nd} and 98^{th} percentiles for each band, on the whole the training set. Then, normalization is performed on train and test images by clipping each band between those percentiles before applying bandwise min-max normalization, similar to [Audebert et al., 2019, Maffei et al., 2020]. For the close-range dataset ICVL, we simply apply global min-max normalization as in [Xiong et al., 2020, Wei et al., 2020].

Noise patterns

We evaluate our model against different types of synthetic noise:

- *i.i.d Gaussian noise with known variance* σ^2, which is the same on all bands.

- *Gaussian noise with unknown band-dependent variance*: We consider Gaussian noise with different standard deviation σ_j for each band, which is uniformly drawn in a fixed interval. These standard deviations change from an image to the other and are unknown at test time.

- *Noise with spectrally correlated variance*: We consider Gaussian noise with standard deviation σ_j varying continuously across bands, following a Gaussian curve, see details in the appendix.

[1] https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html
[2] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

- *Stripes noise* : similar to [Wei et al., 2020], we applied additive stripes noise to 33% of bands. In those bands, 10-15% of columns are affected, meaning a value uniformly sampled in the interval $[-0.25, 0.25]$ is added to them. Moreover, all bands are disturbed by Gaussian noise with noise intensity $\sigma = 25$.

Metrics

In order to assess the performances the previous methods, we used five different indexes widely used for HSI restoration, namely

- Mean Peak Signal-to-Noise Ratio (MPSNR), which is the classical PSNR metric averaged across bands;
- Mean Structural Similarity Index Measurement (MSSIM), which is based on the SSIM metric [Wang et al., 2004];
- Mean Feature Similarity Index Measurement (MFSIM) introduced in [Zhang et al., 2011];
- Mean ERGAS[Du et al., 2007];
- Mean Spectral Angle Map (MSAM)[Alparone et al., 2007].

We use MPSNR and MSSIM in the main chapter and report the other metrics in the appendix.

Implementation details

We trained our network by minimizing the MSE between the ground truth and restored images. For ICVL, we follow the training procedure described in [Wei et al., 2020]: we first center crop training images to size 1024×1024, then we extract patches of size 64×64 at scales 1:1, 1:2, and 1:4, with stride 64, 32 and 32 respectively. The number of extracted patches for ICVL amounts to 52962. For Washington DC Mall, we do not crop training images and the patches are extracted with stride 16, 8 and 8, for a total of 1650 patches. One epoch in Washington DC Mall corresponds to 10 iterations on the training dataset. Basic data augmentation schemes such as $90°$ rotations and vertical/horizontal flipping are performed. Code and additional details about optimization, implementation, computational resources, are provided in the supplementary material. As reported in Table 2.4, augmenting the number unrolled iterations improves the denoising performances at the expense of inference time. Since the Spectral-Spatial SC layer is the most time-consuming, the number of unrolled iterations chosen for the first and second layers are 12 and 5 respectively.

2.4 Discussion and Conclusion

Quantitative results on synthetic noise

We present in Table 2.2 the results obtained on the ICVL dataset (results on DCMall are presented in the appendix). Our method uses the vanilla model of Section 2.2.2 for the experiments with constant σ or correlated noise.

Table 2.2: Denoising performance on ICVL with various types of noise patterns. The first four rows correspond to i.i.d. Gaussian noise with fixed σ per band. The next three rows corresponds to a noise level that depends on the band, taken uniformly on small interval. This is a blind-noise experiment since at test time, the noise level is unknown. The last two rows correspond to the scenarios with correlated σ across bands, and with stripe noise, respectively. See main text for details.

σ	Metrics	Noisy	BM3D	BM4D	GLF	LLRT	NGMeet	SMDS	QRNN3D	T3SC
5	MPSNR	34.47	46.17	48.85	51.25	51.86	**52.74**	50.91	48.80	52.62
	MSSIM	0.7618	0.9843	0.9916	0.9949	0.9951	**0.9960**	0.9944	0.9918	0.9959
25	MPSNR	21.44	37.86	39.89	43.16	43.43	44.74	42.83	44.20	**45.38**
	MSSIM	0.1548	0.9269	0.9510	0.9695	0.9746	0.9796	0.9700	0.9782	**0.9825**
50	MPSNR	16.03	34.22	34.22	39.26	39.69	41.08	39.25	41.67	**42.16**
	MSSIM	0.0502	0.8654	0.8654	0.9197	0.9504	0.9602	0.9382	0.9655	**0.9677**
100	MPSNR	10.85	30.43	32.47	34.79	36.39	37.55	35.64	37.19	**38.99**
	MSSIM	0.0144	0.7557	0.8155	0.7982	0.9182	0.9311	0.8815	0.9140	**0.9439**
[0-15]	MPSNR	33.89	45.81	45.35	50.57	48.50	41.67	48.23	52.07	**53.31**
	MSSIM	0.6386	0.9767	0.9735	0.9948	0.9899	0.9078	0.9900	0.9957	**0.9967**
[0-55]	MPSNR	23.36	39.06	38.43	44.22	41.13	32.94	41.76	47.13	**48.64**
	MSSIM	0.2601	0.9231	0.9074	0.9818	0.9580	0.7565	0.9620	0.9884	**0.9911**
[0-95]	MPSNR	19.06	36.17	35.55	41.43	38.44	29.40	38.94	43.98	**46.30**
	MSSIM	0.1614	0.8760	0.8540	0.9674	0.9354	0.6609	0.9357	0.9753	**0.9859**
Corr.	MPSNR	28.85	42.73	42.13	47.05	45.76	38.06	45.98	48.90	**49.89**
	MSSIM	0.4740	0.9599	0.9070	0.9881	0.9824	0.8536	0.9835	0.9911	**0.9923**
Strip.	MPSNR	21.20	34.88	37.70	42.06	39.38	39.78	41.98	44.60	**44.74**
	MSSIM	0.1508	0.8641	0.9198	0.9628	0.9258	0.9333	0.9655	**0.9806**	0.9805

Our supervised approach achieves state-of-the-art results (or is close to the best performing baseline) on all settings. GLF performs remarkably well given that this baseline is learning-free.

A visual result on ICVL is shown in Figure 2.2 for stripes noise. Inference times are provided in Table 2.3, showing that our approach is computationally efficient.

Results on real noise

We also conducted a denoising experiment on the Urban dataset, reporting a visual result in Figure 2.3. Deep models were pre-trained on the APEX dataset, which has the same number of channels as Urban (even though the sensors are different), with band-dependent noise with $\sigma \in [0 - 55]$. We show that learning-based models trained on synthetic noise are able to transfer to real data.

Comments on the additional results presented in the appendix

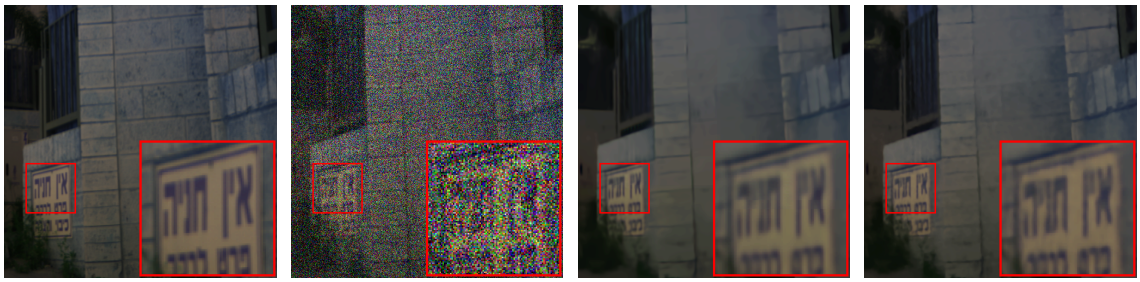
The appendix also contains (i) results on the DCMall dataset including additional baselines mentioned above; (ii) error bars for parts of our experimental results in order to assess their statistical significance; (iii) an experiment when learning simultaneously on several datasets with different types of sensors showing that the second layer can be generic and effective at the same time; (iv) additional visual results; (v) various ablation studies to illustrate the importance of different components of our method.

Table 2.3: Inference time per image on ICVL with $\sigma = 50$; SMDS, QRNN3D and T3SC are using a V100 GPU; BM4D, GLF, LLRT and NGMeet are using an Intel(R) Xeon(R) CPU E5-1630 v4 @ 3.70GHz. Note that unlike GLF, NGMeet, and LRRT, learning-based approaches such as QRNN3D and our approach require a training procedure, which may be conducted offline. The cost of such a training step was about 13.5 hours for our method and 19 hours for QRNN3D on a V100 GPU.

	BM3D	BM4D	GLF	LLRT	NGMeet	SMDS	QRNN3D	T3SC
Inference time (s)	1677	2382	5565	24384	2686	74.3	**3.6**	<u>5.8</u>

Table 2.4: Impact of the number of unrolled iterations per layer on denoising performances and inference time. This experiment was carried out on ICVL with $\sigma = 50$.

Unrolled iterations per layer	1	2	5	12
MPSNR	40.16	41.48	42.15	42.45
Inference time (s)	0.38	1.44	5.27	14.91



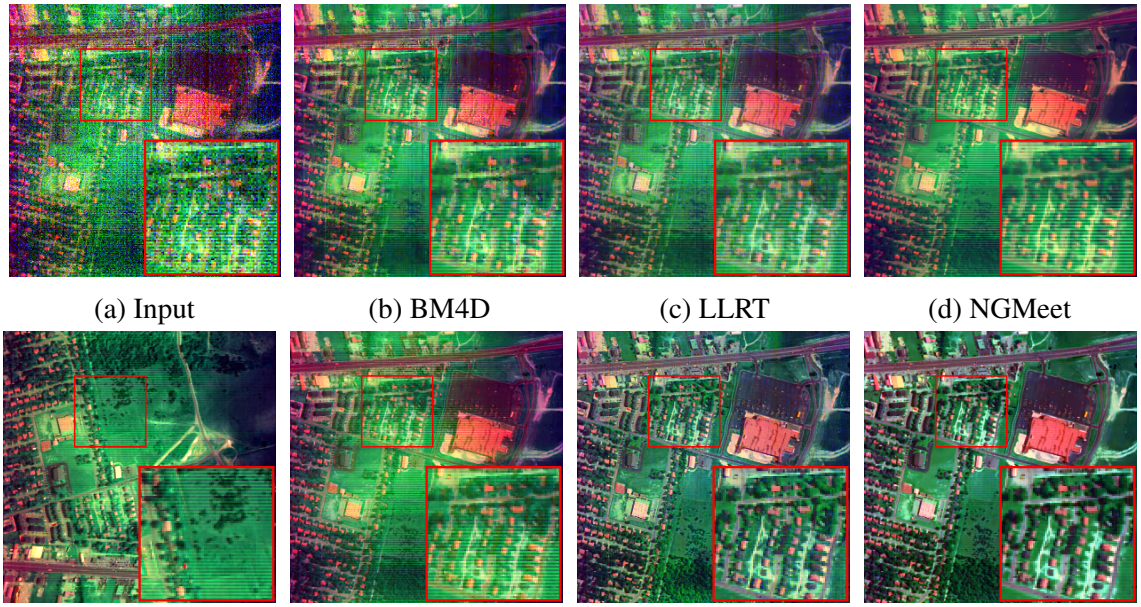
(a) Groundtruth (b) Noisy (c) LLRT (d) NGMeet

(e) GLF (f) SMDS-Net (g) QRNN3D (h) T3SC

Figure 2.2: Denoising results with Gaussian noise $\sigma = 25$ on ICVL with bands 9, 15, 28.

Broader impact

This chapter addresses the problem of denoising the signal, which is a key pre-processing step before using hyperspectral signals in concrete applications. As such, it is necessarily subject to dual use. For instance, HS imaging may be used for environmental monitoring, forestry, yield estimation in agriculture, natural disaster management planning, astronomy, archaeology, and medicine. Yet, HS imaging is also used by the petroleum industry for finding new oil fields, and has obvious military applications for surveillance. We believe the potential benefits of HSI for society are large enough to outweigh the potential harm. Nevertheless, we are planning to implement appropriate dissemination strategies to mitigate the risk of misuse for this work (notably with restrictive software licenses), while targeting a gold standard regarding the scientific reproducibility of our results.



(a) Input (b) BM4D (c) LLRT (d) NGMeet

(e) GLF (f) SMDS-Net (g) QRNN3D (h) T3SC

Figure 2.3: Visual result on a real HSI denoising experiment on Urban dataset with bands 1, 108, 208.

APPENDIX

2.5 Implementation details

In this section, we provide additional implementation details, which are useful to reproduce our experiments (note that the code is also provided).

Noise with spectrally correlated variance. For each band $i \in \{0, \dots, c-1\}$, the standard deviation of the Gaussian noise is defined as :

$$\sigma_i = \beta \; exp\left[-\frac{1}{4\eta^2}\left(\frac{i}{c} - \frac{1}{2}\right)^2\right]$$

with $\beta = 23.08$ and $\eta = 0.157$.

Preprocessing. A basic centering step is used for each input patch of our model. More precisely, for the first layer, each band of the input hyperspectral image is centered independently prior to patches extraction, and means are added back after decoding. For the second layer, patches are centered independently for each band (and similarly, the means are added back after decoding).

Code and patch sizes The hyperparameters of our model are presented in Table 2.5.

Layer	Patches size	Code size	Unrolled iterations	Rank
Spectral SC	1×1	64	12	1
Spectral-Spatial SC	5×5	1024	5	3

Table 2.5: Architecture of our model

Table 2.13 shows that the combination of both layers is more effective than each layer independently.

Initialization All parameters are initialized with He initialization [He et al., 2015].

Blocks inference In order to apply our model to large images, we split them into blocks of size 256×256 with an overlap of 6 pixels. Each block is denoised independently. The output image is obtained by aggregating the denoised blocks. Pixels comprised in several blocks are averaged.

Optimization Our models are trained with batch size of 16 for 60 epochs. We use the Adam optimizer, the initial learning rate is 3×10^{-4}, and is divided by two at epoch 30 and 45.

2.6 Additional quantitative results

Washington DC Mall dataset. Results for this dataset are presented in Table 2.6. Additional baselines are presented in Table 2.7. The conclusions are similar to those already drawn in the main paper.

Table 2.6: Denoising performances on Washington DC Mall.

σ	Metrics	Noisy	BM3D	BM4D	GLF	LLRT	NGMeet	SMDS	QRNN3D	T3SC
5	MPSNR	34.31	35.10	41.13	39.57	41.83	37.57	42.83	43.42	**43.85**
	MSSIM	0.9821	0.9875	0.9962	0.9953	0.9968	0.9928	0.9971	0.9973	**0.9978**
25	MPSNR	20.70	24.51	31.08	35.25	34.95	35.38	35.64	35.04	**36.74**
	MSSIM	0.7688	0.8859	0.9690	0.9883	0.9863	0.9886	0.9889	0.9864	**0.9912**
50	MPSNR	15.25	20.80	26.69	31.77	30.94	31.88	31.76	31.72	**33.12**
	MSSIM	0.5314	0.7508	0.9220	0.9761	0.9704	0.9759	0.9765	0.9741	**0.9819**
100	MPSNR	10.48	17.65	22.51	27.81	26.82	27.86	28.02	27.41	**29.48**
	MSSIM	0.2888	0.5427	0.8141	0.9475	0.9322	0.9460	0.9491	0.9375	**0.9618**
[0-15]	MPSNR	33.32	34.62	37.22	39.89	40.04	37.40	40.77	**43.72**	41.83
	MSSIM	0.9551	0.9746	0.9903	0.9950	0.9951	0.9926	0.9958	**0.9971**	0.9968
[0-55]	MPSNR	22.45	26.11	29.04	38.37	33.36	32.55	34.31	38.44	**39.28**
	MSSIM	0.7450	0.8683	0.9504	0.9934	0.9811	0.9780	0.9859	0.9925	**0.9945**
[0-95]	MPSNR	18.18	23.06	25.77	36.98	30.07	29.21	30.80	35.84	**37.20**
	MSSIM	0.5889	0.7688	0.9033	0.9914	0.9643	0.9589	0.9718	0.9877	**0.9920**
Corr.	MPSNR	28.48	30.50	33.69	37.96	37.77	36.56	38.54	39.84	**40.79**
	MSSIM	0.9085	0.9515	0.9637	0.9928	0.9921	0.9911	0.9934	0.9944	**0.9960**
Strip.	MPSNR	20.47	24.08	29.07	35.27	34.13	34.94	35.24	35.25	**36.34**
	MSSIM	0.7621	0.8672	0.9433	0.9877	0.9833	0.9876	0.9876	0.9874	**0.9906**

Table 2.7: Denoising performances on Washington DC Mall with additional baselines.

σ	Metrics	Noisy	BM3D	BM4D	GLF	LLRT	NGMeet	3D-ADNet	HSID-CNN	HSI-SDeCNN	SMDS-Net	QRNN3D	T3SC
5	MPSNR	34.31	35.10	41.13	39.57	41.83	37.57	42.08	41.68	39.98	42.83	43.42	**43.85**
	MSSIM	0.9821	0.9875	0.9962	0.9953	0.9968	0.9928	0.9968	0.9966	0.9954	0.9971	0.9973	**0.9978**
25	MPSNR	20.70	24.51	31.08	35.25	34.95	35.38	33.78	33.05	33.44	35.64	35.04	**36.74**
	MSSIM	0.7688	0.8859	0.9690	0.9883	0.9863	0.9886	0.9825	0.9813	0.9822	0.9889	0.9864	**0.9912**
50	MPSNR	15.25	20.80	26.69	31.77	30.94	31.88	29.73	28.96	29.61	31.76	31.72	**33.12**
	MSSIM	0.5314	0.7508	0.9220	0.9761	0.9704	0.9759	0.9587	0.9536	0.9608	0.9765	0.9741	**0.9819**
100	MPSNR	10.48	17.65	22.51	27.81	26.82	27.86	24.74	25.29	25.75	28.02	27.41	**29.48**
	MSSIM	0.2888	0.5427	0.8141	0.9475	0.9322	0.9460	0.9064	0.9014	0.9121	0.9491	0.9375	**0.9618**

Study of statistical significance for the ICVL dataset. In order to evaluate the statistical significance of our results, we present some results in Table 2.8 for some of our models and baselines, by running models with five different random seeds. Note that we did not conduct such a study for all results in this paper in order to keep the computational cost of the project reasonable. The conclusions of the paper remain unchanged.

Table 2.8: Denoising performances on ICVL with multiple seeds

σ	Metrics	Noisy	GLF	NGMeet	SMDS	QRNN3D	T3SC
5	MPSNR	34.47 ± 0.01	51.25 ± 0.01	$\mathbf{52.74 \pm 0.01}$	50.78 ± 0.09	49.54 ± 1.28	$\underline{52.62 \pm 0.01}$
	MSSIM	0.7619 ± 0.0001	0.9951 ± 0.0001	$\mathbf{0.9961 \pm 0.0001}$	0.9943 ± 0.0001	0.9924 ± 0.0021	$\underline{0.9960 \pm 0.0001}$
25	MPSNR	21.43 ± 0.01	43.16 ± 0.01	$\underline{44.74 \pm 0.01}$	42.63 ± 0.11	44.20 ± 0.16	$\mathbf{45.37 \pm 0.02}$
	MSSIM	0.1548 ± 0.0002	0.9696 ± 0.0001	$\underline{0.9797 \pm 0.0001}$	0.9687 ± 0.0009	0.9780 ± 0.0009	$\mathbf{0.9825 \pm 0.0001}$
50	MPSNR	16.03 ± 0.01	39.26 ± 0.01	41.09 ± 0.01	39.09 ± 0.08	$\underline{41.47 \pm 0.14}$	$\mathbf{42.16 \pm 0.01}$
	MSSIM	0.0503 ± 0.0001	0.9198 ± 0.0002	0.9603 ± 0.0001	0.9359 ± 0.0012	$\underline{0.9639 \pm 0.0012}$	$\mathbf{0.9677 \pm 0.0001}$
100	MPSNR	10.85 ± 0.01	34.78 ± 0.01	37.55 ± 0.01	35.59 ± 0.04	$\underline{38.38 \pm 0.60}$	$\mathbf{38.99 \pm 0.01}$
	MSSIM	0.0144 ± 0.0001	0.7981 ± 0.0004	0.9312 ± 0.0001	0.8781 ± 0.0017	$\underline{0.9370 \pm 0.0114}$	$\mathbf{0.9439 \pm 0.0002}$
[0-15]	MPSNR	33.94 ± 0.09	50.68 ± 0.11	41.57 ± 0.14	48.00 ± 0.13	$\underline{52.10 \pm 0.12}$	$\mathbf{53.10 \pm 0.12}$
	MSSIM	0.6381 ± 0.0013	0.9950 ± 0.0001	0.9065 ± 0.0022	0.9899 ± 0.0001	$\underline{0.9958 \pm 0.0001}$	$\mathbf{0.9966 \pm 0.0001}$
[0-55]	MPSNR	23.41 ± 0.09	44.41 ± 0.12	32.93 ± 0.09	41.42 ± 0.18	$\underline{47.26 \pm 0.12}$	$\mathbf{48.57 \pm 0.28}$
	MSSIM	0.2621 ± 0.0025	0.9820 ± 0.0004	0.7534 ± 0.0031	0.9593 ± 0.0015	$\underline{0.9889 \pm 0.0004}$	$\mathbf{0.9915 \pm 0.0005}$
[0-95]	MPSNR	19.11 ± 0.09	41.62 ± 0.11	29.40 ± 0.12	38.86 ± 0.06	$\underline{44.07 \pm 0.08}$	$\mathbf{46.24 \pm 0.24}$
	MSSIM	0.1644 ± 0.0031	0.9667 ± 0.0007	0.6601 ± 0.0051	0.9352 ± 0.0004	$\underline{0.9758 \pm 0.0003}$	$\mathbf{0.9863 \pm 0.0005}$

CAVE dataset. We report denoising performances of T3SC on the CAVE Dataset in Table 2.9 To evaluate T3SC, the dataset was divided in four splits : three were used for training and one for testing. The values reported for T3SC are averaged across all rotations of the test split.

Table 2.9: Denoising performances on CAVE dataset with Gaussian noise.

σ	Metrics	Noisy	NGMeet	T3SC
5	MPSNR	35.05	47.96	**49.16**
25	MPSNR	21.99	42.44	**42.77**
50	MPSNR	16.37	38.89	**39.7**
100	MPSNR	10.96	34.99	**36.48**

Joint training across heterogeneous datasets. In Table 2.10, we study the problem of training a single model on three different datasets, APEX, DC Mall, and Pavia, involving a different number of channels. As mentioned in the paper, this model involves a common second layer and a spectral dictionary per dataset. These result show that most of the model parameters (which are present in the second layer) can in fact be shared across datasets without significant loss of accuracy when compared to the training of three different models (thus involving three times more parameters).

Table 2.10: Results for joint training experiment

Training procedure	Model	Metrics	APEX	DC Mall	Pavia Center
Independant trainings	QRNN3D	MPSNR	33.19	31.72	30.56
		MSSIM	0.9619	0.9741	0.9569
	T3SC	MPSNR	**34.91**	**33.12**	**31.32**
		MSSIM	**0.9730**	**0.9819**	**0.9617**
Joint training	QRNN3D	MPSNR	31.95	30.97	29.12
		MSSIM	0.9501	0.9690	0.9428
	T3SC	MPSNR	**34.74**	**33.08**	**31.30**
		MSSIM	**0.9711**	**0.9819**	**0.9616**

Additional metrics. Additional metrics are provided for the ICVL and DCMall datasets, respectively in Tables 2.11 and 2.12. The conclusions of the paper are unchanged.

σ	Metrics	Noisy	BM3D	BM4D	GLF	LLRT	NGMeet	SMDS	QRNN3D	T3SC
5	MFSIM	0.9953	0.9978	0.9986	0.9994	0.9995	**0.9996**	0.9993	0.9987	**0.9996**
	MERGAS	6.18	1.48	1.10	0.84	0.7740	**0.69**	0.87	1.14	0.70
	MSAM	0.2460	0.0518	0.0390	0.0267	0.0229	**0.0211**	0.0307	0.0412	0.0223
25	MFSIM	0.9218	0.9773	0.9829	0.9944	0.9942	0.9954	0.9921	0.9967	**0.9970**
	MERGAS	27.33	3.86	3.21	2.13	2.19	1.77	2.20	1.86	**1.65**
	MSAM	0.5989	0.1286	0.1005	0.0595	0.0459	**0.0384**	0.0717	0.0537	0.0406
50	MFSIM	0.8100	0.9488	0.9488	0.9851	0.9851	0.9863	0.9782	**0.9928**	0.9925
	MERGAS	51.48	5.88	5.88	3.33	3.92	2.71	3.33	2.50	**2.40**
	MSAM	0.7546	0.1964	0.1964	0.1029	0.0682	**0.0505**	0.1033	0.0571	0.0549
100	MFSIM	0.6471	0.8942	0.9008	0.9679	0.9637	0.9661	0.9456	**0.9835**	0.9824
	MERGAS	95.97	9.11	7.96	5.59	6.22	4.08	5.04	4.20	**3.46**
	MSAM	0.8619	0.2984	0.2228	0.1847	0.0919	**0.0679**	0.1441	0.1009	0.0761
[0-15]	MFSIM	0.9876	0.9954	0.9963	0.9991	0.9985	0.9965	0.9984	0.9995	**0.9996**
	MERGAS	10.11	1.91	2.07	0.98	1.17	4.53	1.20	0.79	**0.69**
	MSAM	0.3412	0.0680	0.0672	0.0328	0.0311	0.1772	0.0408	0.0265	**0.0234**
[0-55]	MFSIM	0.9087	0.9743	0.9768	0.9950	0.9900	0.9755	0.9890	0.9984	**0.9985**
	MERGAS	33.34	4.17	4.73	2.07	3.02	14.69	2.50	1.39	**1.20**
	MSAM	0.6478	0.1443	0.1412	0.0687	0.0636	0.4086	0.0784	0.0427	**0.0370**
[0-95]	MFSIM	0.8291	0.9524	0.9560	0.9911	0.9798	0.9536	0.9772	0.9969	**0.9972**
	MERGAS	54.92	5.83	6.73	2.86	4.64	24.82	3.46	2.17	**1.58**
	MSAM	0.7720	0.2001	0.1928	0.0992	0.0813	0.5574	0.1042	0.0622	**0.0471**
Corr.	MFSIM	0.9704	0.9902	0.9923	0.9981	0.9968	0.9919	0.9969	0.9990	**0.9991**
	MERGAS	14.20	2.61	3.74	1.46	1.63	6.37	1.55	1.12	**1.02**
	MSAM	0.4617	0.0934	0.1540	0.0468	0.0416	0.2550	0.0515	0.0316	**0.0291**
Strip.	MFSIM	0.9068	0.9579	0.9736	0.9926	0.9871	0.9880	0.9900	**0.9968**	0.9965
	MERGAS	28.14	7.65	4.65	2.52	4.34	4.32	2.44	1.78	**1.77**
	MSAM	0.6067	0.2197	0.1442	0.0764	0.1272	0.1298	0.0790	**0.0439**	0.0534

Table 2.11: Additional metrics on ICVL

σ	Metrics	Noisy	BM3D	BM4D	GLF	LLRT	NGMeet	SMDS	QRNN3D	T3SC
5	MFSIM	0.9534	0.9578	0.9772	0.9824	0.9817	0.9785	0.9802	**0.9824**	0.9814
	MERGAS	3.12	2.84	1.50	1.96	1.46	2.50	1.38	1.26	**1.19**
	MSAM	0.0862	0.0775	0.0427	0.0495	0.0395	0.0569	0.0373	0.0349	**0.0329**
25	MFSIM	0.8213	0.8676	0.9394	0.9661	0.9629	0.9655	0.9639	0.9614	**0.9673**
	MERGAS	14.96	9.50	4.55	2.91	3.31	2.94	2.87	3.08	**2.50**
	MSAM	0.3087	0.1753	0.1044	0.0684	0.0726	0.0671	0.0676	0.0709	**0.0599**
50	MFSIM	0.7174	0.7861	0.8974	0.9495	0.9439	0.9484	0.9464	0.9487	**0.9542**
	MERGAS	28.00	14.51	7.44	4.24	4.89	4.28	4.45	4.38	**3.68**
	MSAM	0.4785	0.2175	0.1438	0.0890	0.0925	0.0864	0.0944	0.0880	**0.0768**
100	MFSIM	0.6000	0.6821	0.8240	0.9188	0.9065	0.9209	0.9170	0.9100	**0.9329**
	MERGAS	48.42	20.83	11.98	6.54	7.58	6.66	6.52	7.01	**5.51**
	MSAM	0.6566	0.2700	0.1939	0.1183	0.1193	0.1147	0.1205	0.1297	**0.0977**
[0-15]	MFSIM	0.9338	0.9455	0.9690	**0.9831**	0.9774	0.9761	0.9787	0.9828	0.9782
	MERGAS	5.42	4.29	2.29	2.10	1.89	2.53	1.68	**1.36**	1.48
	MSAM	0.1358	0.1052	0.0610	0.0509	0.0487	0.0582	0.0438	**0.0368**	0.0395
[0-55]	MFSIM	0.8196	0.8642	0.9261	**0.9766**	0.9554	0.9523	0.9603	0.9714	0.9748
	MERGAS	18.46	10.41	5.56	2.37	3.86	4.19	3.22	2.37	**2.05**
	MSAM	0.3563	0.1879	0.1171	0.0572	0.0798	0.0961	0.0731	0.0581	**0.0518**
[0-95]	MFSIM	0.7471	0.8057	0.8837	**0.9725**	0.9377	0.9339	0.9473	0.9613	0.9689
	MERGAS	29.42	14.25	8.15	2.68	5.36	6.14	4.60	3.07	**2.50**
	MSAM	0.4899	0.2274	0.1466	0.0632	0.0973	0.1262	0.0962	0.0719	**0.0604**
Corr.	MFSIM	0.9028	0.9229	0.9519	0.9783	0.9713	0.9693	0.9721	**0.9790**	0.9768
	MERGAS	8.25	5.91	4.07	2.29	2.44	2.67	2.10	1.92	**1.65**
	MSAM	0.2049	0.1368	0.1106	0.0559	0.0593	0.0661	0.0540	0.0481	**0.0436**
Strip.	MFSIM	0.8177	0.8621	0.9365	**0.9663**	0.9604	0.9649	0.9639	0.9619	0.9651
	MERGAS	15.38	10.20	4.84	3.00	3.55	3.09	2.99	3.02	**2.62**
	MSAM	0.3152	0.1886	0.1101	0.0698	0.0794	0.0705	0.0700	0.0702	**0.0623**

Table 2.12: Additional metrics on DCMall

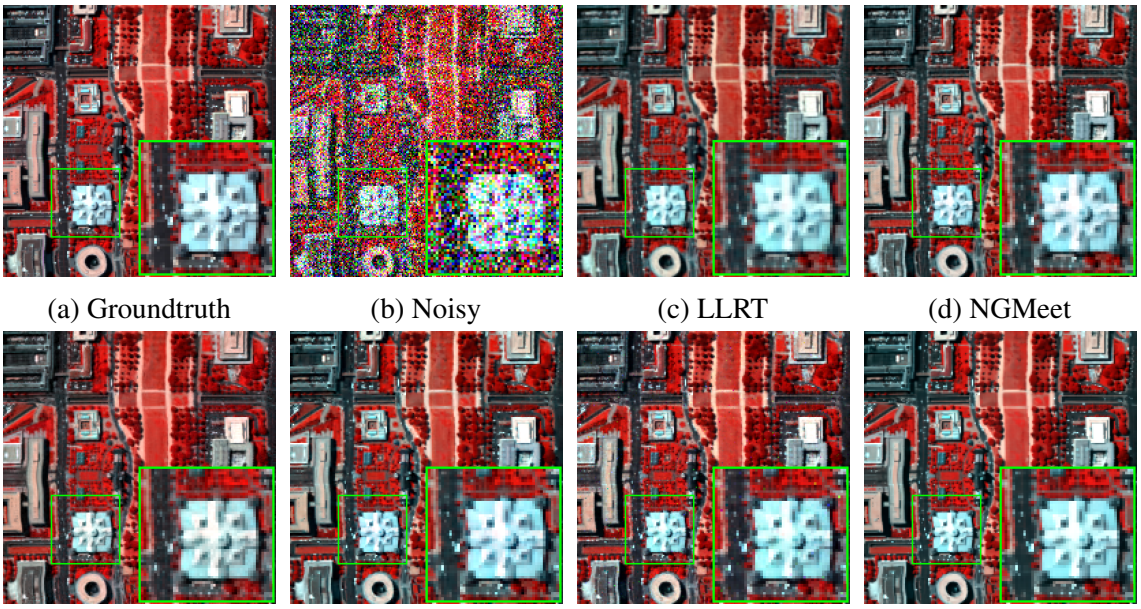
Ablation study. In this paragraph, we present an ablation study, demonstrating in Table 2.13 that our two-layer model outperforms single-layer models.

Metrics	Noisy	Spec	SpecSpat	Spec + SpecSpat
MPSNR	16.03	30.96	40.13	**42.17**
MSSIM	0.0502	0.6884	0.9533	**0.9677**
MFSIM	0.8100	0.9708	0.9849	**0.9925**
MERGAS	51.48	8.84	3.00	**2.39**
MSAM	0.7546	0.1300	0.1021	**0.0547**

Table 2.13: Combination of sparse coding layers: we denote by *Spec* the Spectral Sparse Coding layer and by *SpecSpat* the Spectral-Spatial Sparse Coding layer. This experiment was run on ICVL with $\sigma = 50$.

2.7 Visual examples

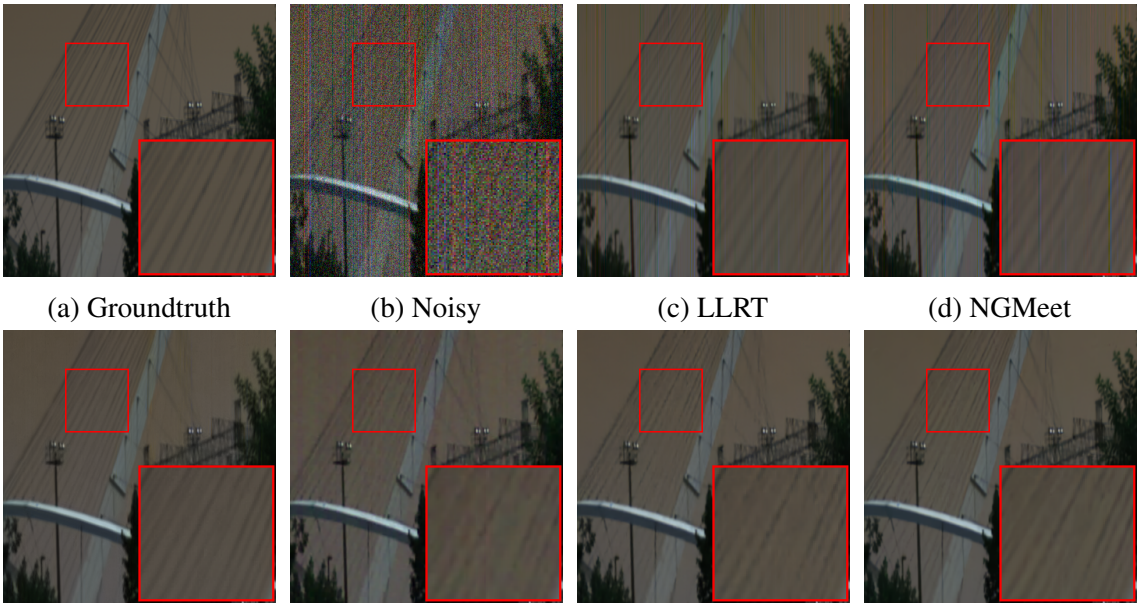
Finally, we show additional visual examples in Figure 2.4 and 2.5.



(a) Groundtruth (b) Noisy (c) LLRT (d) NGMeet

(e) GLF (f) SMDS-Net (g) QRNN3D (h) T3SC

Figure 2.4: Simulated Gaussian noise ($\sigma = 100$) on DCMall



(a) Groundtruth (b) Noisy (c) LLRT (d) NGMeet

(e) GLF (f) SMDS-Net (g) QRNN3D (h) T3SC

Figure 2.5: Visual results for the denoising experiment with stripes noise on ICVL with bands 9, 15, 28.

2.8 GPU resources

The total number of GPU hours involved in this project is around 19k hours on NVIDIA Tesla V100 16Go, including preliminary experiments, model design, final experiments and running baseline methods.

3

Entropic Descent Archetypal Analysis for Blind Hyperspectral Unmixing

Chapter abstract: In this chapter, we introduce a new algorithm based on archetypal analysis for blind hyperspectral unmixing, assuming linear mixing of endmembers. Archetypal analysis is a natural formulation for this task. This method does not require the presence of pure pixels (i.e., pixels containing a single material) but instead represents endmembers as convex combinations of a few pixels present in the original hyperspectral image. Our approach leverages an entropic gradient descent strategy, which (i) provides better solutions for hyperspectral unmixing than traditional archetypal analysis algorithms, and (ii) leads to efficient GPU implementations. Since running a single instance of our algorithm is fast, we also propose an ensembling mechanism along with an appropriate model selection procedure that make our method robust to hyper-parameter choices while keeping the computational complexity reasonable. By using six standard real datasets, we show that our approach outperforms state-of-the-art matrix factorization and recent deep learning methods.

The source code is freely available at https://github.com/inria-thoth/EDAA.

The chapter is based on the following publication:

A. Zouaoui, G. Muhawenayo, B. Rasti, J. Chanussot, and J. Mairal. Entropic descent archetypal analysis for blind hyperspectral unmixing. In *IEEE Transactions on Image Processing*, 2023

3.1 Introduction

Hyperspectral (HS) imaging [Landgrebe, 2002, Plaza et al., 2009, Schaepman et al., 2009, Goetz et al., 1985, Green et al., 1998] consists of measuring the electromagnetic spectrum in a scene by using multiple narrow spectral bands. Thanks to its richer spectral information compared to traditional RGB images, HS images enable more accurate materials identification, leading to a broad range of applications including crop monitoring in agriculture [Adão et al., 2017], waste sorting [Karaca et al., 2013], food safety inspection [Gowen et al., 2007], or mineralogy[Fox et al., 2017].

Remote sensing [Clark et al., 2003, Bioucas-Dias et al., 2013], such as airborne or satellite imagery, yields HS images whose pixels capture several objects or materials. As such, each pixel can include several pure spectral components (called *endmembers*), mixed in different proportions [Ghamisi et al., 2017]. Any further analysis hence requires identifying and disentangling endmembers present in a scene before estimating their respective proportions, or fractional *abundances*, within each pixel of the HSI [Parra et al., 1999]. Since the endmembers spectrum signatures are not known beforehand and must be estimated from data, this operation is named *blind* HS unmixing [Keshava and Mustard, 2002, Bioucas-Dias et al., 2012] owing to its link with blind source separation [Comon and Jutten, 2010].

In this chapter, we adopt a linear mixing model since it is often relevant in remote sensing scenes where mixtures occur between macroscopic materials. Therefore, we assume that each observed pixel can be represented as a linear combination of endmembers and some additive noise. In other words, we are interested in tackling unsupervised linear HS unmixing [Parra et al., 1999].

Further assumptions on the nature of endmembers are generally needed to estimate meaningful spectra. For instance, it can be assumed that there exists at least one pure pixel for each material present in the scene. The problem then requires finding these pure pixels within the original image. The pure pixel assumption is at the core of several geometrical endmember extraction methods including pixel purity index (PPI) [Boardman et al., 1995], N-FINDR [Winter, 1999] and vertex component analysis (VCA) [Nascimento and Dias, 2005]. Once endmembers have been extracted, abundances can be estimated by minimizing the least squares errors between the original input spectra and the linearly reconstructed spectra as long as the abundances fractions satisfy the two physical constraints stating that they should be non-negative and sum to one for each pixel [Heinz and Chein-I-Chang, 2001]. That being said, pure pixels are often missing in real scenarios. In the absence of pure pixels and in the case of linear models, endmembers and abundances can be simultaneously estimated by solving a constrained or penalized non-negative matrix factorization problem (NMF) [Lee and Seung, 2000]. For example, the authors of [Zhuang et al., 2019] have proposed a formulation that involves a data fidelity term and a minimum volume regularization term on endmembers, whose minimization consists in alternating between solving for endmembers and abundances.

In this work, we do not assume the existence of pure pixels as they are often missing in real data, since, for instance, the spectral signatures of endmembers in HS images can be significantly affected by various changes in atmospheric, illumination, and environmental conditions within the scene[Borsoi et al., 2021]. There are multiple strategies that can be employed to tackle spectral variability such as (i) augmenting the linear mixing model[Hong et al., 2018], (ii) using a tensor-based approach, like the Sparsity-Enhanced Convolutional Decomposition (SeCoDe) method[Yao et al., 2021], (iii) performing unmixing in orthogonal subspaces as in[Ren et al., 2023]. Instead, we mitigate the effect

of spectral variability by (i) normalizing each pixel by the ℓ_2-norm of its spectrum as a pre-processing step and (ii) modeling endmembers as convex combinations of pixels present in the scene. Not only HS pixels are linear combinations of the estimated endmembers under the linear mixing model, but the estimated endmembers are also convex combinations of pixels. This corresponds to the archetypal analysis (AA) formulation introduced by Cutler and Breiman in [Cutler and Breiman, 1994]. AA has the advantage to be more interpretable than NMF because the basis elements (*i.e. endmembers*) are directly constructed from the data points (*i.e. pixels*). In addition, since the estimated endmembers spectral signatures generally correspond to averaging the contributions of several pixels, the resulting estimation appears to be more robust to noise and spectral variability than pure pixel methods that only select one pixel per endmember. However, AA usually suffers from a high data fitting error because the basis elements are constrained to be contained in the convex cone of the data points[De Handschutter et al., 2019].

The contributions and innovations of this article are as follows:

1. We propose a new hyperspectral unmixing algorithm relying on entropic gradient descent for archetypal analysis. Our approach (i) provides solutions for hyperspectral unmixing as good as traditional alternating optimization schemes based on projected gradient methods or active set algorithms, and (ii) allows more efficient GPU implementations.

2. The efficiency of our method enables us to make a key practical contribution, consisting of an ensembling mechanism along with an appropriate model selection procedure, which makes our method almost parameter-free and thus easy to use (the only sensitive parameter is the number of endmembers we want to estimate).

3. Our approach, available in an open-source package[1], outperforms state-of-the-art matrix factorization and deep learning methods on six standard real datasets.

The remainder of this chapter is organized as follows. Section 3.2 introduces our method. Section 3.3 presents experimental results highlighting the performance of our proposed approach. Finally, we conclude the article and underline future research directions in Section 3.4.

3.2 Method

In this section, we present our model formulation before describing its optimization. Next, we mention implementation details required to run our approach. Finally, we explain how to leverage our efficient GPU implementation and propose a procedure to make our model robust to hyper-parameter choices and thus easy to use in practice.

3.2.1 Model formulation

Under the linear mixing model (LMM) presented in (1.10), we recall the archetypal analysis formulation introduced in (1.14):

[1]Code is available at https://github.com/inria-thoth/EDAA

$$\begin{aligned} \underset{\mathbf{B},\,\mathbf{A}}{\arg\min} \quad & \frac{1}{2}\left\|\mathbf{Y}-\mathbf{YBA}\right\|_F^2, \\ \text{s.t.} \quad & \mathbf{b}_j \in \Delta_n \text{ for } 1 \leq j \leq r, \\ & \mathbf{a}_i \in \Delta_r \text{ for } 1 \leq i \leq n, \end{aligned} \tag{3.1}$$

where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r] \in \mathbb{R}^{n \times r}$ is the pixel contributions matrix such that the endmembers matrix becomes $\mathbf{YB}$, and the columns of $\mathbf{B}$ are constrained to lie within the simplex Δ_n, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{R}^{r \times n}$ is the abundance matrix, and its columns are constrained to lie within the simplex Δ_r. Finally, $\mathbf{Y} \in \mathbb{R}^{p \times n}$ is the HS image containing n spectra of length p, cast in two dimensions obtained by flattening the spatial dimensions.

3.2.2 Optimization

As explained in Section 1.4.4, solving (3.1) is difficult since the objective function is not jointly convex in $(\mathbf{A}, \mathbf{B})$. However, it is convex with respect to one of the variables when the other one is fixed, as demonstrated in [Mørup and Hansen, 2012]. We have seen in Section 1.4.4 that we essentially need to tackle a quadratic program (QP) under simplicial constraints – *e.g.*, (1.17) , that we recall here:

$$\min_{\mathbf{a} \in \Delta_r} \left[f(\mathbf{a}) = \left\|\mathbf{y}-\mathbf{Za}\right\|_2^2 \right], \tag{3.2}$$

where $\mathbf{y}$ is in $\mathbb{R}^p$, $\mathbf{Z}$ is in $\mathbb{R}^{p \times r}$, $\mathbf{a}$ is in $\mathbb{R}^r$, and f is a convex Lipschitz continuous function with a gradient at $\mathbf{a} \in \Delta_r$ denoted by $\nabla f(\mathbf{a})$.

In this chapter, we adopt entropic gradient descent as our optimization scheme. As introduced in Section 1.4.4, it corresponds to considering the following update: for all j in $\{1, \ldots, r\}$,

$$\mathbf{a}_j^{k+1} = \frac{\mathbf{a}_j^k e^{-\eta^k \nabla f(\mathbf{a}^k)_j}}{\sum_{l=1}^r \mathbf{a}_l^k e^{-\eta^k \nabla f(\mathbf{a}^k)_l}}, \tag{3.3}$$

where $\mathbf{a}_j^k$ is the j-th entry of the vector $\mathbf{a}^k$ and similarly, $\nabla f(\mathbf{a}^k)_j$ is the j-th entry of $\nabla f(\mathbf{a}^k)$, and η^k corresponds to a step size.

We now detail the calculation that leads to (3.3). We start by writing the Lagrangian of (3.2) corresponding to the constraint $\sum_{j=1}^r \mathbf{a}_j = 1$ using the definition of D_h (1.20), the Bregman divergence [Bregman, 1967] endowed by the negative entropy function h (1.19):

$$\mathcal{L}(\mathbf{a}, \nu) = \nabla f(\mathbf{a}^k)^\top \mathbf{a} + \frac{1}{\eta^k} \sum_{j=1}^d \mathbf{a}_j \ln(\mathbf{a}_j) - \frac{1}{\eta^k} \sum_{j=1}^d \mathbf{a}_j \ln(\mathbf{a}_j^k) + \nu \left(\sum_{j=1}^d \mathbf{a}_j - 1 \right). \tag{3.4}$$

Next, for $j \in \{1, \ldots, r\}$, we consider the derivative of $\mathcal{L}$ with respect to the j-th component of $\mathbf{a} \in \Delta_r$:

$$\nabla_{\mathbf{a}_j} \mathcal{L}(\mathbf{a}, \nu) = \nabla f(\mathbf{a}^k)_j + \frac{1}{\eta^k} \ln(\mathbf{a}_j) - \frac{1}{\eta^k} \ln(\mathbf{a}_j^k) + \nu + \frac{1}{\eta^k}, \tag{3.5}$$

By considering the equation $\nabla_{\mathbf{a}_j} \mathcal{L}(\mathbf{a}, \nu) = 0$, we obtain:

$$\ln(\mathbf{a}_j) = \ln(\mathbf{a}_j^k) - \eta^k \nabla f(\mathbf{a}^k)_j - \eta^k \nu - 1, \tag{3.6}$$

and

$$\mathbf{a}_j^{k+1} = \mathbf{a}_j^k e^{-\eta^k \nabla f(\mathbf{a}^k)_j} e^{-\eta^k \nu - 1}. \tag{3.7}$$

As a result, $\mathbf{a}_j^{k+1} \geq 0$ and it remains to choose ν such that $\sum_{j=1}^r \mathbf{a}_j^{k+1} = 1$ as done in (3.3),

Assuming that the entries of $\mathbf{a}^k$ are positive, (3.3) can be efficiently implemented by using the softmax function:

$$\mathbf{a}^{k+1} = \text{softmax}\left(\log(\mathbf{a}^k) - \eta^k \nabla f(\mathbf{a}^k)\right), \tag{3.8}$$

where $\log(\mathbf{a}^k)$ is the vector carrying the logarithm of each entry of $\mathbf{a}^k$.

We are now in shape to describe the alternating optimization scheme, by performing, alternatively, K_1 updates of entropic gradient descent for minimizing $\mathbf{A}$ when $\mathbf{B}$ is fixed, and vice versa using K_2 updates. This strategy is presented in Algorithm 1. Formally, by replacing the generic function f with the functions corresponding to the two optimization sub-problems, we obtain the following updates:

$$\mathbf{A}^{k+1} = \text{softmax}\left(\log(\mathbf{A}^k) + \eta_1^k \mathbf{B}^\top \mathbf{Y}^\top (\mathbf{Y} - \mathbf{Y}\mathbf{B}\mathbf{A}^k)\right), \tag{3.9}$$

$$\mathbf{B}^{k+1} = \text{softmax}\left(\log(\mathbf{B}^k) + \eta_2^k \mathbf{Y}^\top (\mathbf{Y} - \mathbf{Y}\mathbf{B}^k\mathbf{A})\mathbf{A}^\top\right), \tag{3.10}$$

where $\log(\mathbf{A}^k)$ is the matrix carrying the logarithm of each entry of $\mathbf{A}^k$ while softmax is applied in parallel on the columns of $\log(\mathbf{A}^k) + \eta_1^k \mathbf{B}^\top \mathbf{Y}^\top (\mathbf{Y} - \mathbf{Y}\mathbf{B}\mathbf{A}^k)$. Note that when $\mathbf{A}$ and $\mathbf{B}$ are initialized with positive values, these iterates keep them positive. In addition, our optimization strategy does not require inverting any matrix contrarily to ADMM-based approaches[Bioucas-Dias and Figueiredo, 2010].

Algorithm 1 Entropic Descent Archetypal Analysis (EDAA)

Input: ℓ_2-normalized data $\mathbf{Y}$ in $\mathbb{R}^{p\times n}$; r (number of endmembers); T (number of outer iterations); K_1 (number of inner iterations for $\hat{\mathbf{A}}$); K_2 (number of inner iterations for $\hat{\mathbf{B}}$).
Initialize $\hat{\mathbf{A}} \in \mathbb{R}^{r\times n}$ using (3.12).
Initialize $\hat{\mathbf{B}} \in \mathbb{R}^{n\times r}$ using (3.13).
Set η_1 according to (3.14).
Set η_2 according to (3.15).
for $t = 1, \ldots, T$ **do**
    for $k = 1, \ldots, K_1$ **do**
        $\hat{\mathbf{A}} \leftarrow \text{softmax}\left(\log(\hat{\mathbf{A}}) + \eta_1 \hat{\mathbf{B}}^\top \mathbf{Y}^\top (\mathbf{Y} - \mathbf{Y}\hat{\mathbf{B}}\hat{\mathbf{A}})\right)$
            ▷ log is applied element-wise;
            ▷ softmax is applied along the first dimension.
    end for
    for $k = 1, \ldots, K_2$ **do**
        $\hat{\mathbf{B}} \leftarrow \text{softmax}\left(\log(\hat{\mathbf{B}}) + \eta_2 \mathbf{Y}^\top (\mathbf{Y} - \mathbf{Y}\hat{\mathbf{B}}\hat{\mathbf{A}})\hat{\mathbf{A}}^\top\right)$
    end for
end for
$\hat{\mathbf{E}} \leftarrow \mathbf{Y}\hat{\mathbf{B}}$
Return $\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$ ▷ Estimated endmembers, abundances.

3.2.3 Implementation details

Normalization The input image $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]$ is ℓ_2-normalized for each pixel: for all i in $\{1, \dots, n\}$,

$$\mathbf{y}_i \leftarrow \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|_2}, \tag{3.11}$$

where $\mathbf{y}_i$ denotes the i-th pixel. This step is important to gain invariance to illumination changes.

Initialization We initialize the abundance matrix $\mathbf{A}$ uniformly,

$$\hat{\mathbf{A}}^0 = \frac{1}{r}\mathbf{1}_r\mathbf{1}_n^T, \tag{3.12}$$

where $\mathbf{1}_d$ denotes a d-dimensional vector of ones. This corresponds to the maximal entropy configuration for each pixel. The entropy for each pixel will naturally decrease as a result of the optimization, but the high entropy of the initialization will have a regularization effect.

The initialization of the pixel contribution matrix $\mathbf{B}$ is then also close to uniform. Nevertheless, we introduce a small random perturbation which is necessary to break the symmetry between the columns of $\mathbf{B}$ (otherwise, the updates of $\mathbf{A}$ and $\mathbf{B}$ will keep them invariant). Concretely, the entries of $\mathbf{B}$ are randomly sampled according to the uniform distribution on $[0, 1]$, $\mathcal{U}_{[0,1]}$. Next, they are rescaled by a factor 0.1. Finally, we apply the softmax function on each column so that the columns of $\mathbf{B}$ belong to the simplex Δ_n, for j in $\{1, \dots, r\}$,

$$\mathbf{b}_j^0 = \text{softmax}(0.1\ \mathbf{u}), \tag{3.13}$$

where $\mathbf{u} \sim \mathcal{U}_{[0,1]^n}$. In practice, we observe that such an initialization leads to a matrix $\hat{\mathbf{B}}^0$ that is very close to a uniform initialization $\frac{1}{n}\mathbf{1}_n\mathbf{1}_r^T$.

Step sizes We use constant step sizes η_1 and η_2, for $\mathbf{A}$ and $\mathbf{B}$ respectively.

$$\eta_1 = \frac{\gamma}{\sigma_{\max}^2}, \tag{3.14}$$

where γ is a value in $S = \{0.125, 0.25, 0.5, 1, 2, 4, 8\}$ and $\sigma_{\max}$ is the largest singular value of the matrix $\mathbf{Y}\mathbf{B}^0$. We recover the classical convergence of gradient descent with fixed step size [Nesterov, 2003] up to the factor γ, since $\sigma_{\max}^2$ corresponds to the Lipschitz constant of the sub-problem related to (3.1) when minimizing w.r.t $\mathbf{A}$, $\mathbf{B}$ being fixed. Having γ in S allows us to use slightly different step sizes and yields better performance in practice. Note that our model selection procedure, presented later, will automatically choose the right parameter γ, thus removing the burden for the user of having to deal with an extra hyper-parameter. Finally, η_2 is simply a rescaled version of η_1 to account for the matrices being of transposed dimensions:

$$\eta_2 = \sqrt{\frac{r}{n}}\ \eta_1. \tag{3.15}$$

Hyperparameters For all experiments, if not stated otherwise, we set $T = 100$ and $K_1 = K_2 = 5$ as it provides a good trade-off between convergence speed and unmixing accuracy. Note also that these hyper-parameters are robust to different real datasets as detailed in section 3.3.

3.2.4 Model selection procedure

As stated above, the initialization of the matrix $\mathbf{B}$ is random, leading to different solutions for each run of the algorithm since the overall optimization problem is non convex. Besides, we allow for different step-sizes γ, which we draw randomly from the set S in practice. Since the convergence of the algorithm is very fast (see experimental section for concrete benchmarks), we are able to provide a large diversity of solutions given a dataset by running M times our method with different random seeds, while keeping the global computational complexity reasonable. A major question we address next is then *how to select optimally the best solution in terms of unmixing accuracy*.

For this, we take inspiration from classical model selection and sparse estimation theory[Hastie et al., 2009]. First, we measure the fit of each solution in terms of residual error $\|\mathbf{Y} - \mathbf{YBA}\|_1$, choosing the ℓ_1-norm which is known to be more robust to outliers than the mean squared error. Second, we *select* the set of solutions that are in the same ball park as the best solution we have obtained in terms of ℓ_1 fit. This selection process is illustrated by the red dotted line in Figure 3.1, while the precise criterion is described in Algorithm 2.

From the subset of solutions with good fit, we then choose the one whose endmembers have the best incoherence, a desired property, which is classical in the theory of sparse estimation[Elad and Bruckstein, 2002, Gribonval and Nielsen, 2003, Mairal et al., 2014]. Indeed, dictionaries (here endmembers) with more incoherence will benefit from better theoretical guarantees in terms of estimation of abundances, making it a natural criterion for model selection in the context of unmixing. Formally, the coherence is simply defined as the maximal pairwise spectral correlation between the estimated endmembers. More precisely, for $\hat{\mathbf{E}} = [\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_p]$ the endmembers matrix, the coherence μ is defined as:

$$\mu = \max_{k \neq k'} \langle \hat{\mathbf{e}}_k, \hat{\mathbf{e}}_{k'} \rangle, \tag{3.16}$$

where $\langle . \rangle$ denotes the inner product.

To the best of our knowledge, this is the first time the coherence μ is used as a model selection criterion for archetypal analysis. Our experiments, see next section, show that it is highly effective.

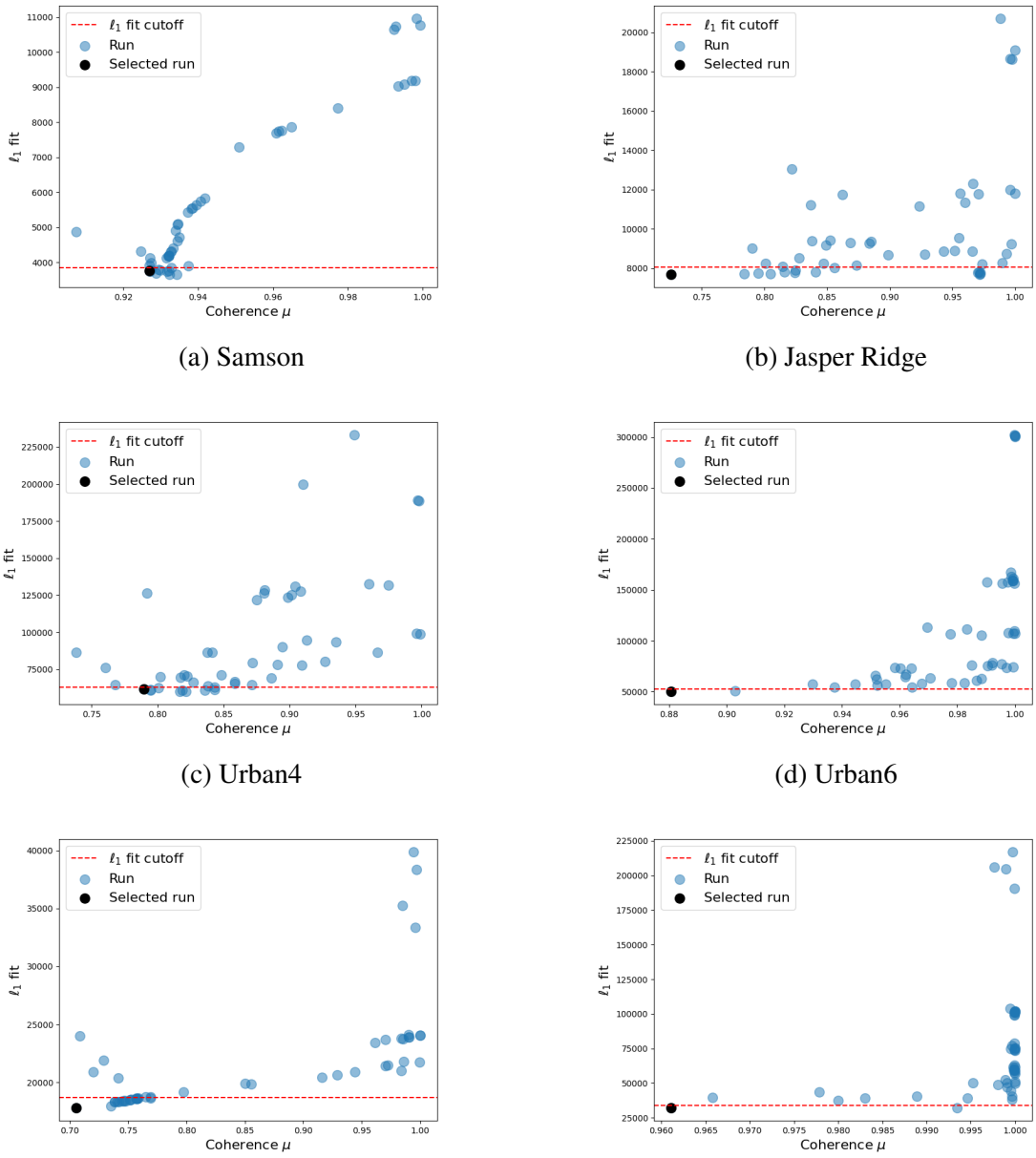
In summary, we automatically select the model whose endmembers have the lowest maximal pairwise spectral correlation among the ones that have a good ℓ_1 fit. This strategy is illustrated in figure 3.1 and described in Algorithm 2. In the experiments, the number of runs M was set to 50.

3.3 Experiments

We have performed experiments on one simulated dataset with different noise and purity levels as well as six standard real datasets whose descriptions are given below.

3.3.1 Data description

1. Simulated dataset: For our study, we chose six endmembers from the USGS library and generated a 1000-pixel data cube using the methodology outlined in[Ambikapathi et al., 2011]. This approach allowed us to vary the purity level of the pixels by adjusting the parameter ρ, which is used in Table 3.1. Specifically, lower values of ρ correspond to pixels that are less pure, while higher values indicate



(a) Samson

(b) Jasper Ridge

(c) Urban4

(d) Urban6

(e) APEX

(f) WDC

Figure 3.1: Illustration of the model selection procedure on six datasets using $M = 50$ runs. Runs are illustrated by blue dots and the selected one is in black. The selected run is the one with lowest coherence μ under the dashed red line representing the ℓ_1 fit threshold, see Alg. 2.

greater purity. By manipulating this parameter, we were able to simulate a range of real-world scenarios and evaluate the robustness of our algorithm under different conditions.

2. Samson: The Samson[2] hyperspectral image is a 95x95 pixels sub-region of a larger image captured using 156 bands spanning from 401 to 889 nm. Three main materials have been identified: Tree, Soil and Water. Note that we use a different ground truth from [Rasti et al., 2022] that we selected for its sharper details on the abundances.

[2]downloaded at https://rslab.ut.ac.ir/data

Algorithm 2 Model Selection Procedure

Input: M (number of runs); ℓ_2-normalized data $\mathbf{Y}$ in $\mathbb{R}^{p\times n}$; r (number of endmembers); T (number of outer iterations); K_1 (number of inner iterations for $\hat{\mathbf{A}}$); K_2 (number of inner iterations for $\hat{\mathbf{B}}$).
for $m = 1, \dots, M$ **do**
    Set random seed s_m.
    $\hat{\mathbf{E}}_m, \hat{\mathbf{A}}_m \leftarrow \text{EDAA}(\mathbf{Y}, r, T, K_1, K_2, s_m)$ ▷ See (1)
    $\text{fit}_m \leftarrow ||\mathbf{Y} - \hat{\mathbf{E}}_m\hat{\mathbf{A}}_m||_1$
    Compute coherence μ_m on $\hat{\mathbf{E}}_m$. ▷ See (3.16)
end for
$\text{fit}_{\min} \leftarrow \min\{\text{fit}_1, \dots, \text{fit}_M\}$
$\mathcal{I} \leftarrow \{m \mid \text{fit}_m \leq 1.05 \times \text{fit}_{\min}\}$ ▷ Subset of models.
$\text{best} \leftarrow \arg\min\{\mu_i, i \in \mathcal{I}\}$
Return: $\hat{\mathbf{E}}_{\text{best}}$, $\hat{\mathbf{A}}_{\text{best}}$.

3. Jasper Ridge: The Jasper Ridge[2] hyperspectral image is a 100x100 pixels sub-region of a larger image initially captured using 224 bands spanning from 380 to 2500 nm. In total, 198 bands remain as 26 were removed as a pre-processing step due to dense water vapor and atmospheric effects. Four main materials have been identified: Tree, Dirt, Water and Road.

4. Urban4 and Urban6: The Urban[2] hyperspectral image is a 307x307 pixels image collected by the Hyperspectral Digital Image Collection Experiment (HYDICE) [Rickard et al., 1993] sensor using 210 bands spanning from 400 to 2500 nm. In total, 162 bands remain as 48 were removed as a pre-processing step due to dense water vapor and atmospheric effects. There exists three versions of this dataset w.r.t. the number of endmembers. In this study, we focus on the two extremes: Urban4 contains 4 endmembers (Asphalt Road, Grass, Tree and Roof) and Urban6 contains two additional materials: Dirt and Metal, making it more challenging.

5. APEX: The APEX [Schaepman et al., 2015] hyperspectral image that we consider in this paper is a 111x122 pixels cropped region[3] of a larger image captured over 285 bands spanning from 413 to 2420nm. Four main materials were identified: Road, Tree, Roof and Water. Note that the ground truth abundance map for water appears to contain shadows due to the sunlight direction (see figure 3.3). This is a common issue when dealing with real remote sensing data and we cannot expect the semi-automated ground truth abundances map generation to be perfect.

6. Washington DC Mall: The Washington DC Mall (WDC) hyperspectral dataset[3] consists in a 319x292 pixels image captured by the HYDICE [Rickard et al., 1993] sensor over 191 bands spanning from 400 to 2400 nm. Six main materials were identified: Grass, Tree, Roof, Road, Water and Trail.

According to [Zhu, 2017] (Samson, Jasper Ridge and Urban) and [Rasti et al., 2022] (APEX and WDC), the endmembers spectra were manually selected from the images and the ground truth abundances were set by the fully constrained least squares (FCLS) unmixing algorithm. Illustrations of the datasets and their ground truth endmembers are available in the supplementary material.

[3] downloaded at https://github.com/BehnoodRasti/MiSiCNet

3.3.2 Experimental setup

We compare our approach to nine competitive methods from different unmixing categories:

- Geometrical unmixing baseline: FCLS [Heinz and Chein-I-Chang, 2001] using VCA [Nascimento and Dias, 2005] to extract endmembers. Our implementation of the FCLS algorithm uses the *DecompSimplex* routine implemented in SPAMS[4]. This method relies on the active-set algorithm [Nocedal and Wright, 1999] that enables significantly faster convergence than generic quadratic programming solvers by leveraging the underlying sparsity of the abundances as noted by [Chen et al., 2014].

- Deep learning unmixing: Endnet[5] [Ozkan et al., 2018] using VCA [Nascimento and Dias, 2005] to initialize the endmembers, MiSiCNet[6] [Rasti et al., 2022] and the deep unrolling network ADMMNet[7] [Zhou and Rodrigues, 2021].

- NMF-based blind unmixing: minimum-volume-constraint non-negative matrix factorization (MVCNMF)[8][Miao and Qi, 2007], non-negative matrix factorization quadratic minimum volume (NMF-QMV)[9] [Zhuang et al., 2019] using the *boundary* term as the quadratic minimum volume penalty, near-convex archetypal analysis (NCAA)[10][De Handschutter et al., 2019] and AA [Cutler and Breiman, 1994] using the implementation from[Chen et al., 2014] developed in SPAMS.

  The approach denoted as AA involves solving (3.1) using an active-set method to optimize the convex sub-problems. This method is a conventional alternating approach that uses a fixed initialization ($\mathbf{A}$ and $\mathbf{B}$ entries are set to 0) and a fixed number of iterations ($T = 100$). While AA is a well-known method, it has not been thoroughly evaluated on various real-world unmixing datasets. Therefore, we include it as a competing method in our study to compare its performance with other state-of-the-art algorithms. By using AA, we can also assess the improvements in optimization achieved by EDAA since the underlying model (3.1) is the same for both methods. This comparison enables us to gain insights into the benefits of EDAA for hyperspectral unmixing.

- Finally, we include a recent technique addressing spectral variability: SeCoDe[11][Yao et al., 2021].

To quantitatively evaluate the performance of the selected methods, we consider two metrics that are computed both globally and individually for each endmember. On one hand, we measure the quality of the generated abundances by means of the abundances root mean square error (RMSE) in percent between the ground truth and the estimated abundances:

[4] http://thoth.inrialpes.fr/people/mairal/spams/
[5] implementation at https://github.com/burknipalsson/hu_autoencoders
[6] implementation at https://github.com/BehnoodRasti/MiSiCNet
[7] no implementation available online
[8] implementation found here
[9] implementation at https://github.com/LinaZhuang/NMF-QMV_demo
[10] implementation found here
[11] implementation at https://github.com/danfenghong/IEEE_TGRS_SeCoDe

$$\text{RMSE}(\mathbf{A}, \hat{\mathbf{A}}) = 100 \times \sqrt{\frac{1}{rn} \sum_{i=1}^{r} \sum_{j=1}^{n} \left(\mathbf{A}_{i,j} - \hat{\mathbf{A}}_{i,j} \right)^2}. \tag{3.17}$$

On the other hand, we assess the quality of the estimated endmembers spectra by using the spectral angle distance (SAD) in degrees between the ground truth and the generated endmembers:

$$\text{SAD}(\mathbf{E}, \hat{\mathbf{E}}) = \frac{180}{\pi} \times \frac{1}{r} \sum_{i=1}^{r} \arccos \left(\frac{\langle \mathbf{e}_i, \hat{\mathbf{e}}_i \rangle}{||\mathbf{e}_i||_2 ||\hat{\mathbf{e}}_i||_2} \right), \tag{3.18}$$

where $\mathbf{e}_i$ denotes the i-th column of $\mathbf{E}$, *i.e.,*the spectrum of the i-th endmember.

3.3.3 Unmixing experiments

Table 3.1: Abundances RMSE and endmembers SAD for the simulated dataset with different noise and purity levels. The best results are shown in bold. The second best results are underlined.

Methods	ρ	RMSE			SAD		
		SNR (dB)			SNR (dB)		
		20	30	40	20	30	40
FCLS [Heinz and Chein-I-Chang, 2001]	0.7	16.53	17.15	17.97	7.20	7.18	7.38
	0.85	12.96	11.44	10.15	3.86	3.05	2.45
	1.0	8.03	4.43	3.31	1.84	0.58	**0.17**
MiSiCNet [Rasti et al., 2022] ($\lambda = 0.1$)	0.7	10.60	8.64	8.31	**3.94**	**1.88**	**1.58**
	0.85	13.76	8.62	7.68	4.70	1.20	**0.75**
	1.0	15.08	6.42	3.73	5.11	1.12	0.39
ADMMNet [Zhou and Rodrigues, 2021]	0.7	15.49	16.92	16.68	11.44	13.76	13.48
	0.85	17.18	19.54	19.58	8.52	10.23	9.93
	1.0	21.13	16.57	16.60	9.03	6.26	6.55
MVCNMF [Miao and Qi, 2007]	0.7	15.80	9.75	15.53	6.85	2.52	6.85
	0.85	9.88	9.80	9.54	**1.85**	**1.16**	0.76
	1.0	6.84	7.03	3.55	1.12	0.78	**0.17**
NMFQMV [Zhuang et al., 2019]	0.7	**10.11**	**8.32**	**7.36**	4.96	4.76	3.32
	0.85	10.58	**7.11**	**6.79**	2.91	1.18	0.98
	1.0	11.24	6.62	3.80	3.25	1.54	0.41
SeCoDe [Yao et al., 2021]	0.7	13.49	13.36	13.62	16.06	16.38	16.54
	0.85	13.56	14.76	14.69	14.62	15.79	15.73
	1.0	7.88	4.74	4.35	7.76	5.07	4.76
NCAA [De Handschutter et al., 2019]	0.7	14.62	13.43	14.64	8.32	8.00	7.44
	0.85	12.15	7.66	7.19	5.72	2.56	1.10
	1.0	11.61	3.20	3.25	5.37	1.70	0.71
AA ($T = 100$)	0.7	17.67	17.23	17.18	7.61	6.71	6.43
	0.85	9.44	8.23	8.04	2.55	2.09	1.86
	1.0	**5.13**	**2.98**	**2.75**	**0.88**	**0.45**	0.34
EDAA ($K_1 = K_2 = 30$)	0.7	14.35	13.75	14.91	5.50	5.01	6.52
	0.85	**8.84**	8.37	8.37	2.63	2.18	1.92
	1.0	5.82	4.26	2.89	0.96	0.51	0.33

Table 3.1 presents the results of our unmixing accuracy evaluation on the simulated dataset. Our analysis indicates that the performance of the AA variants (active-set based

and EDAA) is lower in scenarios where the pixels are highly mixed ($\rho = 0.7$) when compared to geometrically motivated methods, such as MiSiCNet, MVCNMF and NMF-QMV, which do not rely directly on the pixel values to estimate the endmembers. However, it is worth noting that the EDAA method performs better than the plain AA method in this setting, thanks to its advanced model selection. In the case where pure pixels are present ($\rho = 1.0$), the AA model formulation (3.1) yields the best results, as it estimates the endmembers as convex combinations of the pixels. This approach is more robust to noise than the FCLS method. For the medium case ($\rho = 0.85$), both AA and EDAA exhibit better performance than MiSiCNet and NMF-QMV for low Signal-to-Noise Ratio (SNR) due to their greater robustness to noise. However, they are outperformed by MiSiCNet and NMF-QMV for high SNR, as there are no pure pixels available for these methods to leverage.

Table 3.2: Abundances RMSE on six real datasets. The best results are shown in bold. The second best results are underlined.

		FCLS	Endnet	MiSiCNet	ADMMNet	MVCNMF	NMF-QMV	SeCoDe	NCAA	AA ($T = 100$)	EDAA
Samson	Soil	11.28	11.61	6.47	18.41	6.62	13.67	7.67	12.37	6.16	**5.74**
	Tree	9.13	7.72	5.38	14.32	**3.47**	8.40	6.68	4.87	4.00	3.77
	Water	5.05	6.86	3.47	6.67	4.66	11.61	2.57	7.80	**2.30**	2.59
	Overall	8.88	8.97	5.25	14.00	5.08	11.44	6.05	8.89	4.44	**4.24**
Jasper Ridge	Dirt	21.23	18.26	21.68	21.58	20.19	19.97	20.62	18.85	9.29	**7.32**
	Road	24.72	29.40	24.94	21.37	24.28	26.13	21.42	23.95	8.04	**7.61**
	Tree	11.20	4.00	3.41	13.19	9.80	14.55	9.57	**3.34**	7.43	6.63
	Water	13.61	22.38	7.07	13.92	17.83	19.81	9.38	10.36	6.10	**5.69**
	Overall	18.52	20.70	16.98	17.96	18.79	20.53	16.31	16.18	7.80	**6.85**
Urban4	Road	30.54	12.04	10.30	19.45	12.89	20.25	12.84	11.58	11.13	**8.62**
	Grass	32.99	17.94	12.35	22.17	14.00	20.22	24.34	**7.57**	11.47	9.28
	Roof	15.40	11.28	8.01	13.31	10.98	13.29	13.20	10.46	7.30	**6.37**
	Tree	20.02	11.69	8.78	16.45	13.63	21.56	26.36	7.63	8.51	**6.27**
	Overall	25.78	13.52	10.00	18.15	12.93	19.11	20.17	9.48	9.76	**7.75**
Urban6	Road	31.61	17.44	19.18	24.43	21.86	24.81	18.56	17.51	13.84	**11.39**
	Grass	23.62	31.47	18.84	32.86	28.19	27.97	20.97	**11.25**	11.88	18.61
	Roof	13.00	9.35	7.41	14.07	11.40	16.39	20.82	13.31	11.20	**6.01**
	Tree	16.14	15.50	11.72	22.71	16.12	19.95	26.67	10.52	**9.71**	9.85
	Dirt	25.06	30.67	23.95	22.33	20.42	24.31	20.18	26.71	**10.99**	15.94
	Metal	12.99	25.21	33.67	11.20	17.35	13.40	12.05	**8.93**	17.73	15.86
	Overall	21.54	23.09	16.27	22.41	19.92	21.74	20.34	15.88	**12.83**	13.63
APEX	Road	33.31	29.32	32.66	35.81	32.14	31.43	29.11	34.83	20.11	**16.54**
	Tree	20.97	18.18	19.97	27.25	23.96	24.62	22.25	21.98	14.58	**14.48**
	Roof	14.15	15.88	18.42	15.88	15.24	14.73	14.11	14.58	13.06	**11.27**
	Water	18.03	17.47	16.88	17.16	17.86	17.99	**16.77**	17.62	16.80	16.83
	Overall	22.77	20.90	22.86	25.35	23.23	23.10	21.35	23.55	16.36	**14.94**
WDC	Grass	30.90	**27.35**	31.61	39.87	37.79	34.69	30.55	34.78	35.88	32.53
	Tree	22.42	35.98	23.64	26.66	28.55	19.90	34.44	27.19	17.17	**11.46**
	Road	27.90	38.49	34.91	21.00	20.77	22.49	22.15	20.06	39.86	**13.97**
	Roof	8.71	27.04	11.98	9.95	**6.96**	19.81	10.17	18.21	20.52	29.31
	Water	17.76	12.94	14.72	20.44	12.93	20.34	21.23	24.83	24.03	**9.63**
	Trail	12.80	12.63	12.07	14.09	**11.26**	12.36	15.35	11.83	16.83	13.19
	Overall	21.57	27.64	23.39	24.00	22.42	22.60	23.81	24.89	27.24	**20.46**

Tables 3.2 and 3.3 report the unmixing accuracy in terms of abundances RMSE (3.17) and endmembers SAD (3.18) on six standard real datasets. The datasets are arbitrarily ranked based on their difficulty. For a fair comparison, all methods were evaluated on the ℓ_2-normalized data (3.11) which induces slight changes compared to the results reported in [Rasti et al., 2022].

Overall EDAA obtains the best abundances RMSE on five out of six real datasets. The results on Urban6 fall in favor of plain AA but the gap is moderate (12.83 vs 13.63) given that EDAA is more than seven times faster (594 versus 75 seconds) than plain AA on this dataset (see table 3.4).

Note however that plain AA performs poorly on WDC where the ground truth endmembers are highly correlated. We argue that our model selection technique is instrumental

Table 3.3: Endmembers SAD on six real datasets. The best results are shown in bold. The second best results are underlined.

		FCLS	Endnet	MiSiCNet	ADMMNet	MVCNMF	NMF-QMV	SeCoDe	NCAA	AA ($T = 100$)	EDAA
Samson	Soil	2.76	**0.61**	1.21	7.48	1.08	4.90	0.79	4.57	0.78	1.64
	Tree	3.05	1.93	3.38	2.48	2.54	5.34	1.96	2.42	**1.80**	1.98
	Water	7.15	1.48	5.36	6.68	4.82	11.14	2.56	5.15	1.38	**1.31**
	Overall	4.32	1.34	3.32	5.55	2.81	7.13	1.77	4.05	**1.32**	1.64
Jasper Ridge	Dirt	13.03	**1.63**	4.26	21.90	9.41	12.40	9.94	4.08	2.54	2.74
	Road	40.39	32.85	20.04	46.06	36.38	45.66	45.87	41.84	5.26	**3.10**
	Tree	11.16	1.39	**1.27**	10.78	8.28	14.46	12.42	2.25	4.68	4.23
	Water	13.24	3.21	4.18	13.14	10.40	14.53	6.44	5.00	3.03	**2.80**
	Overall	19.46	9.77	7.44	22.97	16.12	21.77	18.67	13.29	3.88	**3.22**
Urban4	Road	15.40	6.40	5.73	27.68	9.00	14.51	17.28	9.44	**3.70**	6.01
	Grass	24.18	3.09	5.84	18.58	7.32	16.39	7.25	4.81	**1.80**	2.14
	Roof	47.56	**3.76**	16.10	33.08	27.80	36.31	30.08	23.98	15.64	10.49
	Tree	19.82	2.32	4.60	11.34	6.39	22.48	16.17	**1.44**	3.47	2.81
	Overall	26.74	**3.89**	8.07	22.67	12.63	22.42	17.69	9.92	6.15	5.36
Urban6	Road	13.43	**3.26**	7.47	35.54	23.96	26.60	32.28	18.95	4.26	4.85
	Grass	22.30	4.13	10.97	29.20	12.60	21.63	13.34	7.36	3.44	**2.17**
	Roof	15.65	17.76	13.97	41.57	28.09	15.55	42.86	27.14	21.18	**13.70**
	Tree	20.70	7.72	9.99	16.67	12.93	23.31	25.85	**5.62**	8.76	8.92
	Dirt	69.81	17.42	19.57	36.46	19.59	23.60	27.66	37.21	**9.87**	13.33
	Metal	39.35	7.04	9.75	39.91	5.78	68.73	57.78	11.13	7.59	**4.52**
	Overall	30.21	9.56	11.95	29.20	17.16	29.90	33.30	17.90	9.18	**8.64**
APEX	Road	40.23	14.46	33.02	49.02	44.02	54.53	48.91	37.26	**4.80**	6.83
	Tree	14.13	7.53	**3.16**	17.57	7.49	16.06	16.37	6.50	7.56	7.68
	Roof	8.25	**4.36**	11.31	11.41	5.34	7.98	8.13	4.41	7.44	7.50
	Water	7.15	2.83	6.02	4.01	3.16	9.71	4.06	9.06	2.32	**2.21**
	Overall	17.44	7.30	13.38	20.50	15.00	22.07	19.37	14.31	**5.53**	6.06
WDC	Grass	17.40	**3.54**	17.46	43.51	11.01	34.36	25.43	15.33	17.10	8.11
	Tree	23.73	12.85	12.36	23.58	20.73	17.70	50.57	25.36	8.17	**1.81**
	Road	32.56	26.76	33.20	39.15	16.95	17.28	21.96	16.06	35.29	**7.67**
	Roof	34.84	13.70	28.87	31.47	**4.77**	44.87	46.46	33.94	16.19	50.97
	Water	4.78	1.75	1.57	11.60	5.43	19.74	30.22	4.64	3.62	**1.08**
	Trail	9.94	**1.49**	3.24	24.85	13.57	9.60	28.89	11.41	4.28	4.75
	Overall	20.54	**10.02**	16.11	29.03	12.08	23.93	33.92	17.79	14.11	12.40

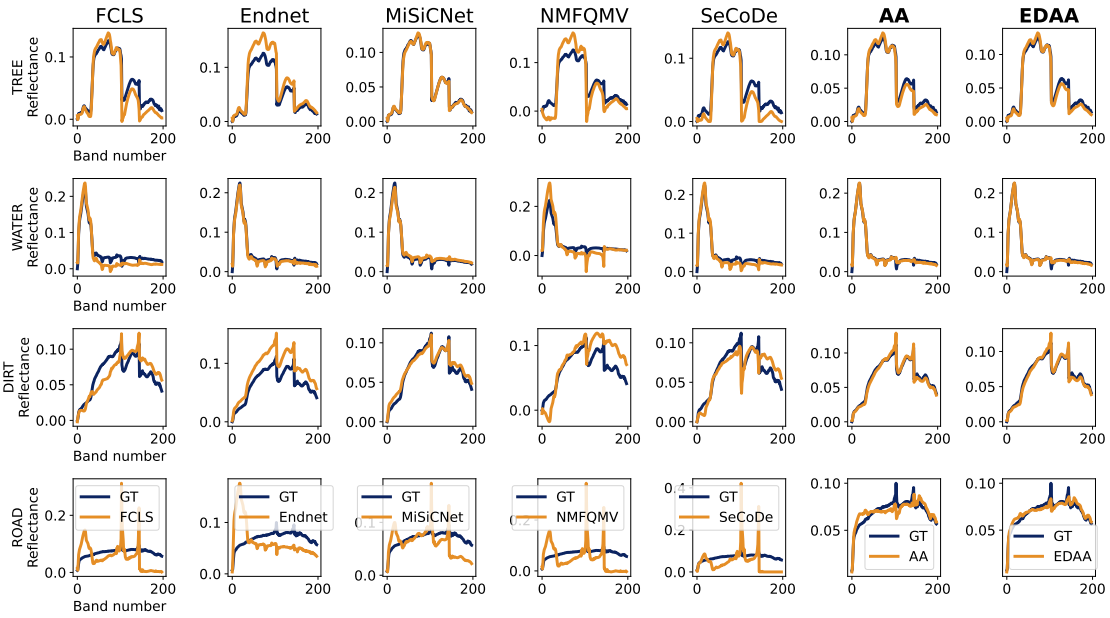
in avoiding collapsing runs in which endmembers spectra are highly correlated. This is underlined by the overall competitive SAD results obtained by EDAA across datasets. It should be noted that the SAD metric alone is not sufficient to assess the unmixing performance as a good SAD score does not necessarily lead to better abundance maps. Thus it is not contradictory to have slightly worse SAD scores yet better looking abundance maps.

The FCLS baseline based on VCA obtains rather poor results except for WDC. This is likely due to the pure pixel assumption. Indeed, VCA selects a single pixel to represent the endmembers spectra, which is too stringent in real scenarios where spectral variability is ubiquitous.

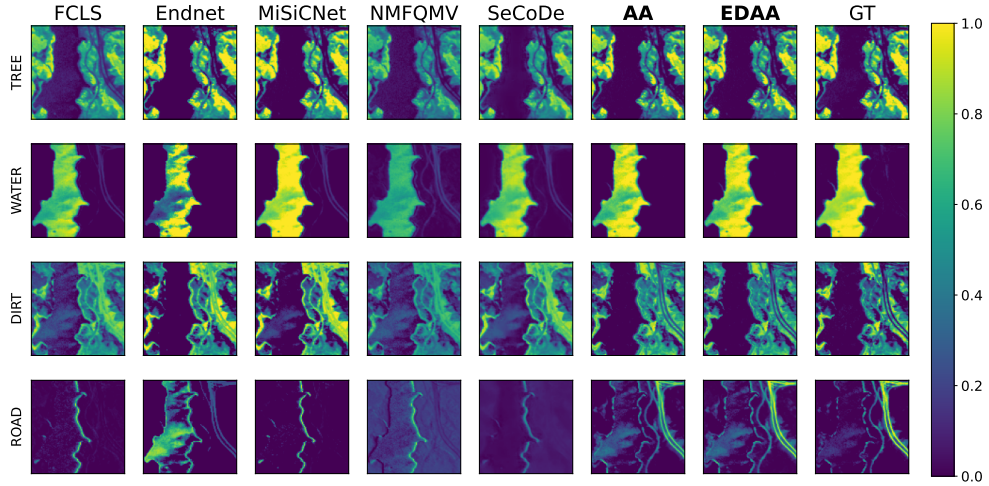
Despite its quadratic minimum volume *boundary* term, NMF-QMV generally obtains worse results than the FCLS baseline. Since it operates the unmixing in a subspace, NMF-QMV cannot prevent the endmembers spectra from having negative values, which breaks the physical interpretability of the estimates and subsequently harms the unmixing performance. This phenomenon can be observed in figures 3.2, 3.3 and 3.4 for several endmembers. The associated abundances show that NMF-QMV produces maps that are too uniform and lack sparsity.

In contrast, MVCNMF is a strong baseline as it does not operate in a subspace.

When it comes to deep learning based methods, Endnet achieves very good results in terms of SAD but tends to create overly sparse abundances which hinders its performance in terms of abundances RMSE. For instance, as can be seen in figure 3.2, the Road endmember is overlooked by Endnet even though EDAA recovers it neatly. Likewise, in figure 3.3, the Road endmember spreads too much compared to EDAA which appears



(a)



(b)

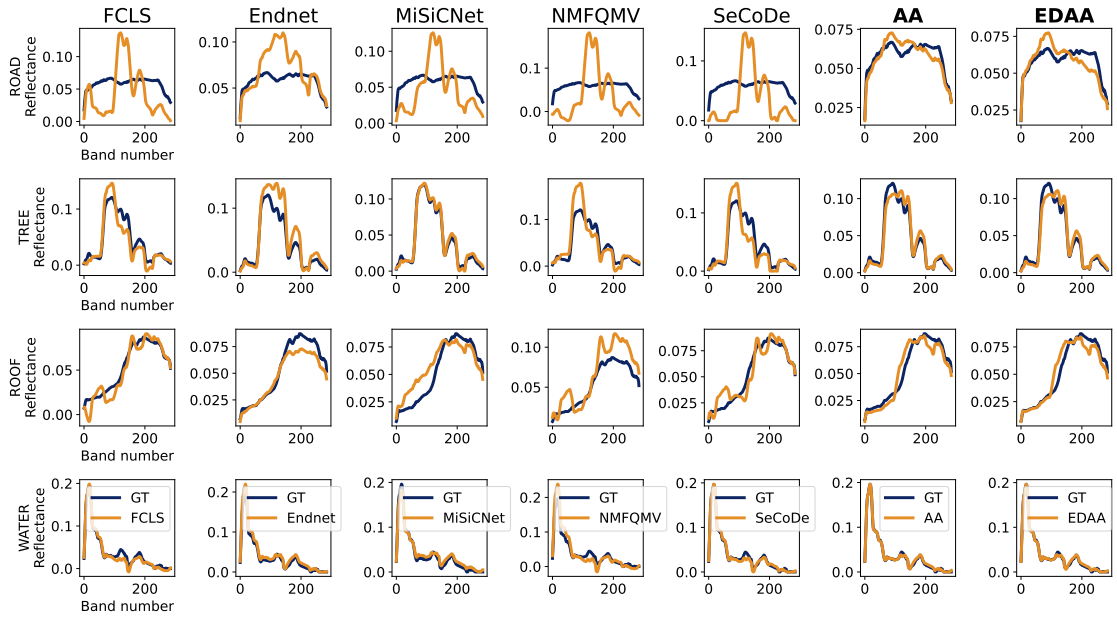
Figure 3.2: Estimated endmembers (a) and abundances (b) on the Jasper Ridge dataset. Ground truth abundances are displayed on the right-most column.

closer to the ground truth.

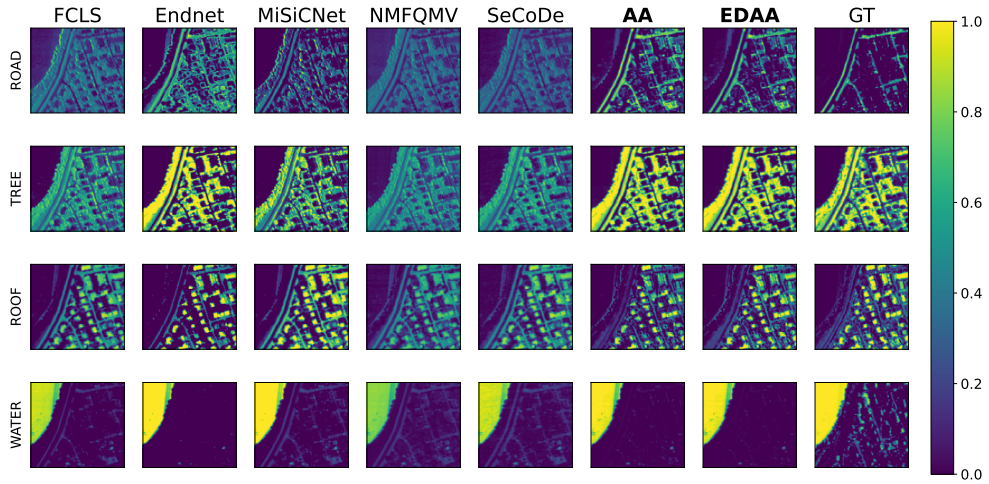
MiSiCNet gives better unmixing results than Endnet in terms of abundances RMSE except for APEX although the SAD results falls in favor of Endnet except for Jasper Ridge. This is likely due to Endnet using the spectral angle distance on the input data in its loss which helps in achieving better SAD accuracy. However a good SAD is not sufficient to obtain good abundance maps, an area where MiSiCNet tends to shine as it incorporates spatial information by using convolutional filters and implicitly applying a regularizer on abundances.

ADMMNet obtains rather poor results on all datasets, which is likely due to using ADMM to blindly solve the linear mixing model rather than leveraging a known library.

SeCoDe works reasonably well on the Samson dataset due to its implementation



(a)



(b)

Figure 3.3: Estimated endmembers (a) and abundances (b) on the APEX dataset. Ground truth abundances are displayed on the right-most column.

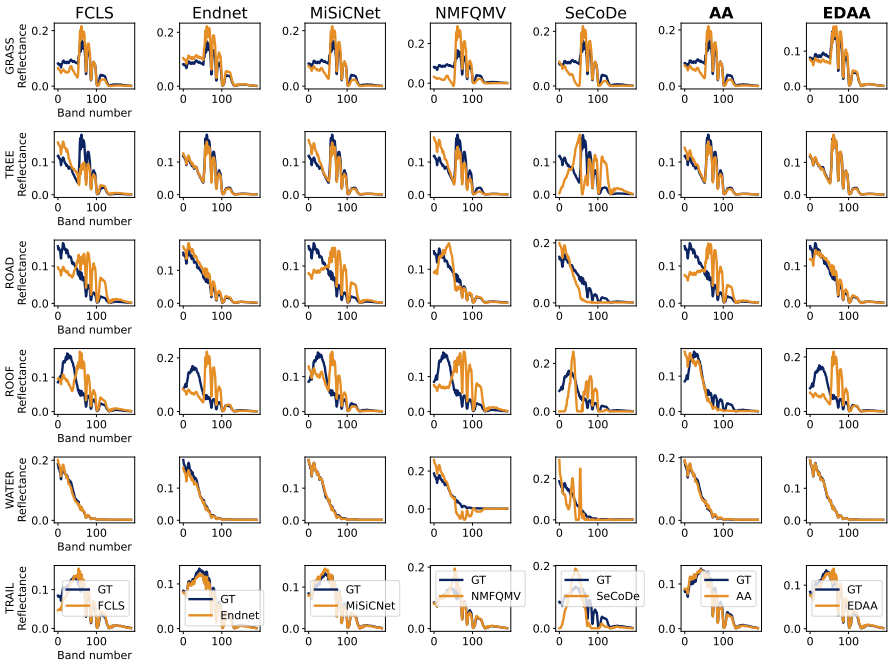
available online that has been tailored to this dataset. However it does not perform as well on other datasets, which is likely due to requiring precise hyperparameters tuning depending on the data at hand.

NCAA obtains reasonable results due to its fine-tuning procedure but its computational cost is prohibitive as we will be discussing next.

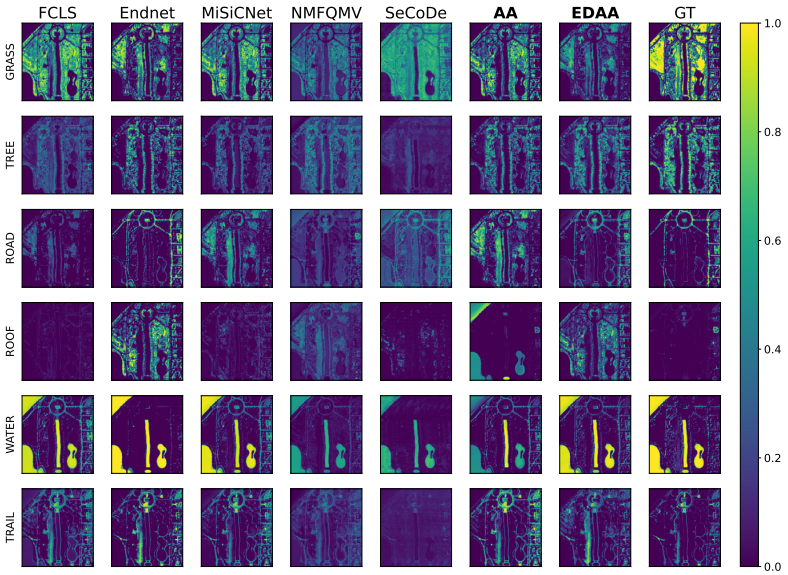
Finally, the plain AA method leveraging an active-set algorithm is a very competitive method for five out of six datasets, yet its performance drops significantly when dealing with the hardest mixing scenario that is WDC.

For example, in figure 3.2 only AA and EDAA are able to uncover the Road endmember in Jasper Ridge whereas all the other techniques fail.

Unlike EDAA, AA does not rely on a random initialization of the estimates as they



(a)



(b)

Figure 3.4: Estimated endmembers (a) and abundances (b) on the WDC dataset. Ground truth abundances are displayed on the right-most column.

are set to zeros in practice. Instead, EDAA requires starting from feasible points w.r.t. the optimization sub-problems. In addition, the initial estimates in EDAA should not contain any zeros due to the presence of the logarithm in (3.8). Thanks to the entropic gradient descent speed, it is possible to fit several models that have been randomly initialized as described in 3.2.3 and use the model selection procedure presented in Algorithm 2 which selects the model that exhibits the lowest coherence among the pool of candidates that are well fitted. Notably, this approach prevents the estimated endmembers from *collapsing*

into degenerate solutions that would end up being perfectly correlated. This likely explains why the performance of plain AA on WDC drops as the endmembers are highly correlated.

Additional qualitative results for the Samson, Urban4 and Urban6 datasets can be found in the supplementary material. Moreover, we have also included the results on a 250×191 pixels subset of the Cuprite dataset, which is more challenging and does not come with ground truth abundance maps.

3.3.4 Computational cost

Table 3.4 reports the processing times for the different unmixing algorithms on the six real datasets. MVCNMF, NMF-QMV, SeCoDe and NCAA were implemented in Matlab (2020b) while FCLS, Endnet, MiSiCNet, ADMMNet and the AA variants were implemented in Python (3.8). NMF-QMV, MVCNMF, SeCoDe, NCAA, FCLS and AA run on CPU whereas Endnet, MiSiCNet, ADMMNet and EDAA run on GPU. The processing times were obtained using a computer with an Intel(R) Xeon(R) Silver 4110 processor (2.10 GHz), 32 cores, 64GB of memory, and a NVIDIA GeForce RTX (2080 Ti) graphical processing unit. The table shows that FCLS is clearly faster than the other unmixing techniques, however it is a supervised method that relies on an endmembers extraction algorithm. In this case, VCA is used which is also fast. The deep learning methods are the slowest techniques despite running on GPU. Interestingly, EDAA requires a lower computational cost than NMF-QMV and AA although our approach consists in aggregating 50 runs obtained iteratively. For example, it takes on average 1.5 seconds for EDAA to perform a single unmixing task on the Urban6 dataset, which is three times faster than FCLS. This demonstrates the efficiency of EDAA which allows us to use an adequate model selection procedure over several runs. Note that the processing time of NCAA is prohibitive due to its fine-tuning component despite using a fast projected gradient method.

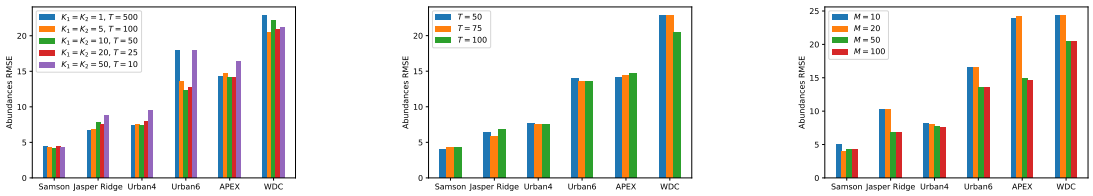
Table 3.4: Processing times in seconds on six real datasets. The best results are in bold and the second best underlined. EDAA includes the model selection procedure over $M = 50$ runs.

	FCLS	Endnet	MiSiCNet	ADMMNet	MVCNMF	NMF-QMV	SeCoDe	NCAA	AA ($T = 100$)	EDAA
Samson	**0.3**	≈ 560	≈ 80	55.2	<u>3.8</u>	20.1	24.2	113.1	33.0	11.2
JasperRidge	**0.4**	≈ 680	≈ 90	54.1	<u>8.0</u>	22.3	28.6	501.4	43.6	9.6
Urban4	**3.6**	≈ 720	≈ 411	504.6	<u>25.4</u>	112.5	475.1	3529.9	415.7	66.3
Urban6	**4.4**	≈ 1000	≈ 417	507.5	<u>45.0</u>	158.4	707.7	4880.8	593.7	75.2
APEX	**0.6**	≈ 720	≈ 92	80.0	<u>11.3</u>	27.2	56.1	441.5	75.4	15.4
WDC	**4.0**	≈ 1000	≈ 409	535.8	<u>55.1</u>	174.4	524.7	4179.0	493.1	81.0

3.3.5 Ablation study

Finally, we study the sensitivity to hyper-parameters for Algorithm 1 and 2 in figure 3.5 where the Y-axis corresponds to the overall abundances RMSE. Given a fixed computational budget of 1000 updates, figure 3.5 (a) shows that the hyper-parameters of EDAA are robust provided that the number of runs M in the model selection is large enough (here 100). Only the two extremes configurations ($K_1 = K_2 = 1, T = 500$ and $K_1 = K_2 = 50$, $T = 10$) are slightly worse, especially on Urban6. For the remaining experiments, we use $K_1 = K_2 = 5$. In figure 3.5 (b), we see that the number of outer iterations is quite stable except for WDC which requires more updates (1000, *i.e.*,$T = 100$). Finally, we study the importance of the number of runs M from which to select the best candidate in

figure 3.5 (c). We observe that the model selection procedure requires at least 50 runs to obtain very good performances, hence we use $M = 50$ in our unmixing experiments. On unknown datasets where real-time unmixing is not required, it is advised to use a large number of runs (at least 100) to ensure that the model selection procedure yields a good candidate. Detailed results for both abundances RMSE and SAD metrics are available in the supplementary material.



(a) (b) (c)

Figure 3.5: Sensitivity analysis to the hyperparameters in Algorithms 1 and 2 measured in global abundances RMSE: (a) Varying inner and outer iterations K_1, K_2 and T for a constant number of updates (1000) and runs $M = 100$, (b) Varying outer iterations T using $K_1 = K_2 = 5$ and (c) Varying number of runs M using $K_1 = K_2 = 5$ and $T = 100$.

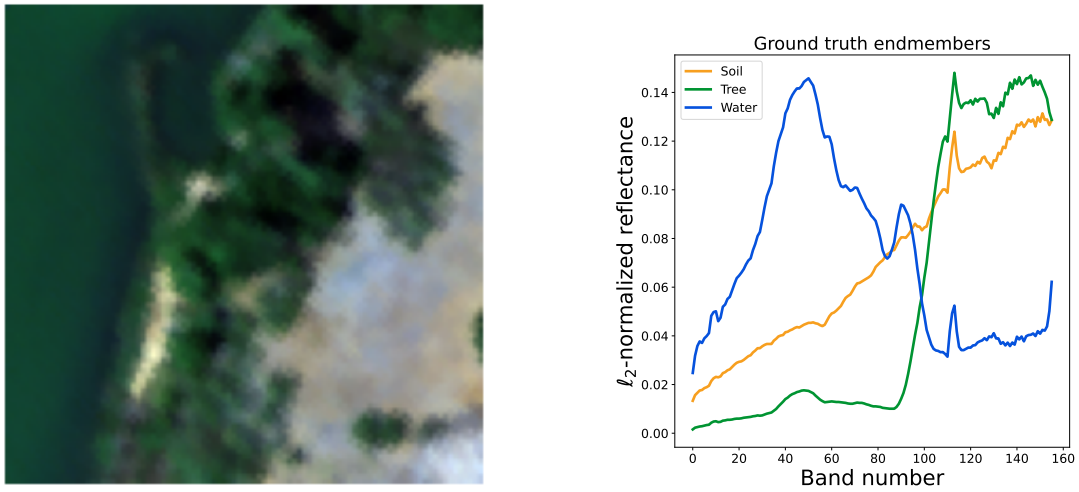
3.4 Conclusion

We have proposed a new algorithm based on archetypal analysis for blind hyperspectral unmixing. We have shown how to take advantage of its efficient GPU implementation in order to develop an adequate model selection procedure to obtain state-of-the-art performances. Remarkably, our simple and easy-to-use approach considerably improves the unmixing results on a comprehensive collection of standard real datasets. In addition, we have made our results reproducible by releasing an open source codebase which also includes the plain archetypal analysis variant presented in this study. While this paper was under review, we also investigated in[Rasti et al., 2023b] the problem of semi-supervised unmixing by using a variant of archetypal analysis, showing that such a framework may be useful beyond the problem of blind unmixing that we address here. Finally, it is worth noting that our approach does not consider the spatial structure of the data. This feature suggests that a natural extension to our approach would be to incorporate missing spatial information, which could potentially improve the accuracy of our results.

APPENDIX

3.5 Datasets description

In this section, we provide illustrations of the unmixing datasets used in the experiments. Each dataset is described with a false-color RGB image alongside the ℓ_2-normalized ground truth endmembers.



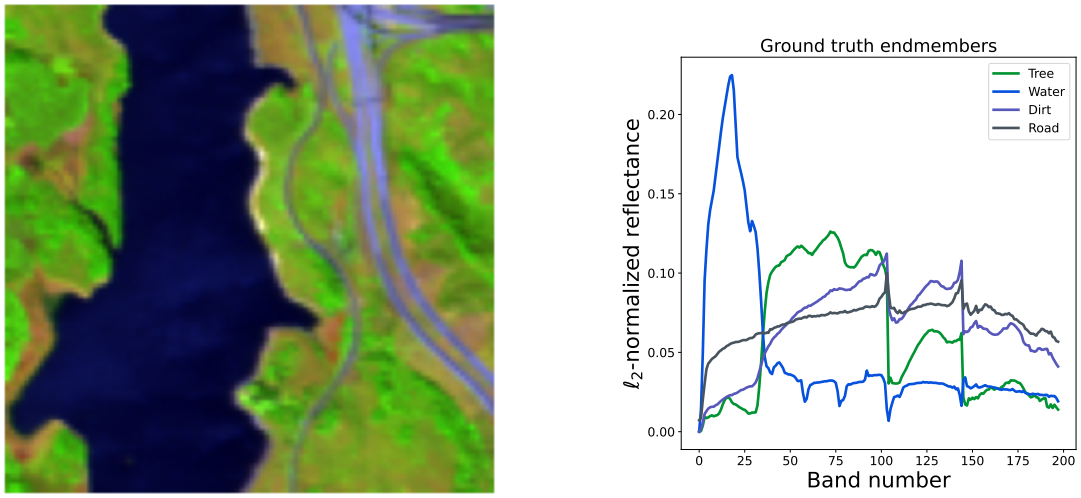
(a) (b)

Figure 3.6: Samson dataset: (a) False colors RGB image (Red: 83rd band, Green: 43, Blue: 9) (b) ℓ_2-normalized ground truth endmembers.

3.6 Additional results

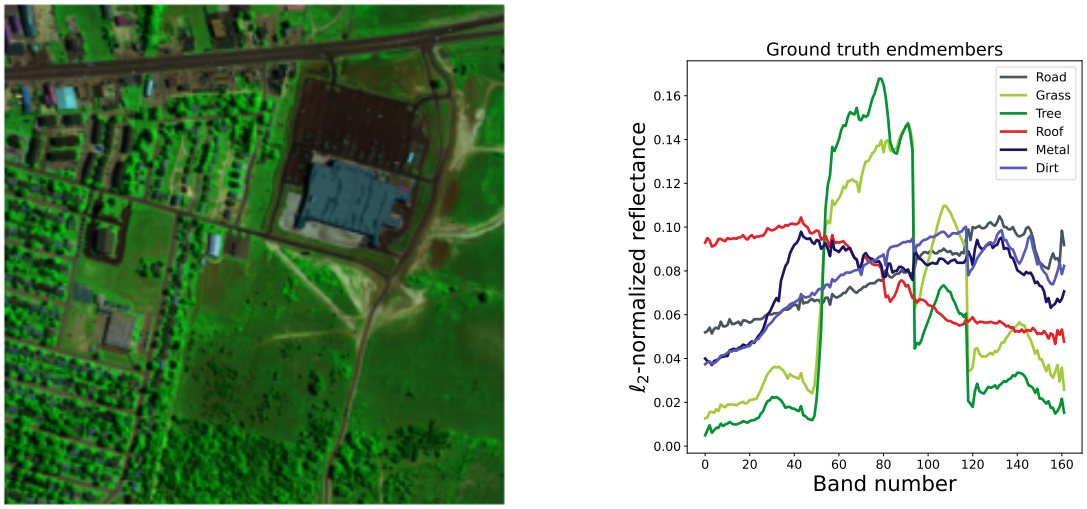
We provide qualitative results on the Samson, Urban4 and Urban6 datasets in figures 3.11, 3.12 and 3.13.

Cuprite dataset In order to evaluate the performance of the selected methods, we provide qualitative results on a 191 $\times$ 250 pixels subset of the Cuprite dataset, which is only accompanied by a geological map of the main materials present in the scene (see Fig.3.14(a)). Recovering the spectral signatures of interest in this dataset is a challenging



(a) (b)

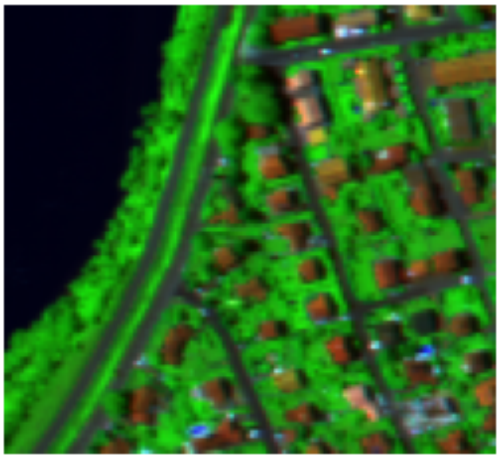
Figure 3.7: JasperRidge dataset: (a) False colors RGB image (Red: 130th band, Green: 50, Blue: 5) (b) ℓ_2-normalized ground truth endmembers.



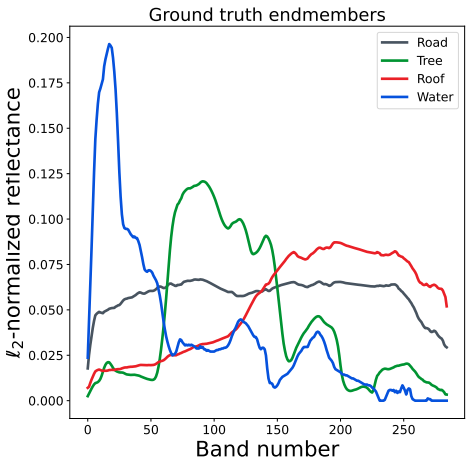
(a) (b)

Figure 3.8: Urban dataset: (a) False colors RGB image (Red: 130th band, Green: 70, Blue: 30) (b) ℓ_2-normalized ground truth endmembers for Urban6. Urban4 corresponds to the first 4 materials.

task due to the similarity of their spectral signatures and the presence of significant spectral variability. All methods were asked to output abundance maps for $r = 5$ unknown endmembers. As shown in Fig.3.14(b), all methods successfully recovered the three dominant materials of interest, namely Chalcedony, Alunite, and Kaolinite. However, we observed that MiSiCNet, AA, and EDAA produced sharper abundance maps than FCLS and NMFQMV. It is worth noting that we could not provide results for Endnet due to its outdated code base. Moreover, we had to increase the number of outer iterations for EDAA to 1000 to obtain meaningful results, while AA iterations could be kept at 100. Finally, we found that the ℓ_1 fitting criterion was not satisfactory and had to be replaced with ℓ_2 for this particular dataset.

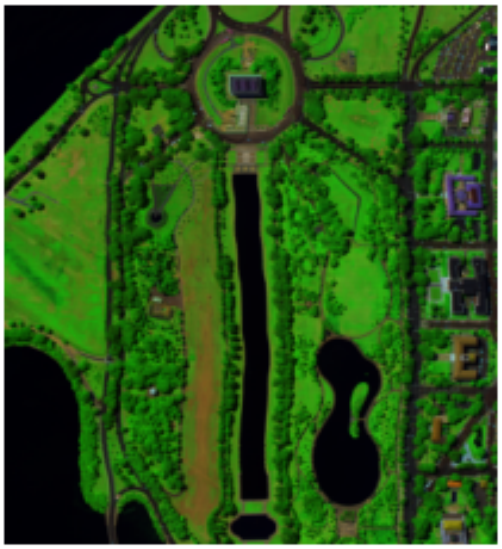


(a)

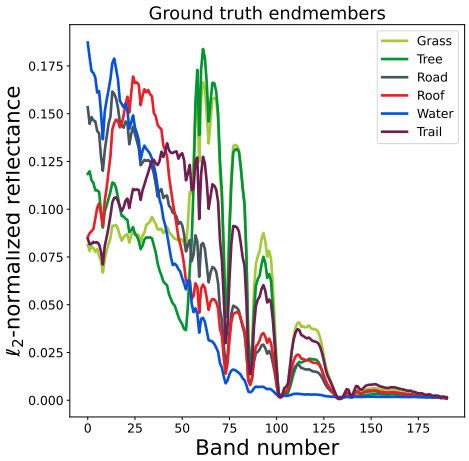


(b)

Figure 3.9: APEX dataset: (a) False colors RGB image (Red: 200th band, Green: 100, Blue: 10) (b) ℓ_2-normalized ground truth endmembers.



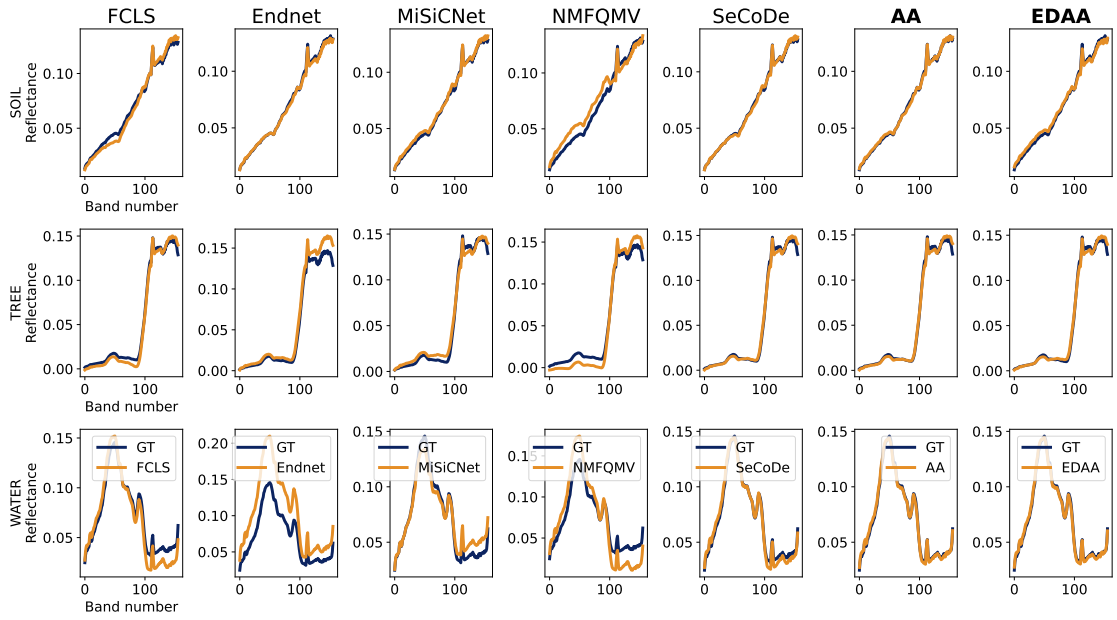
(a)



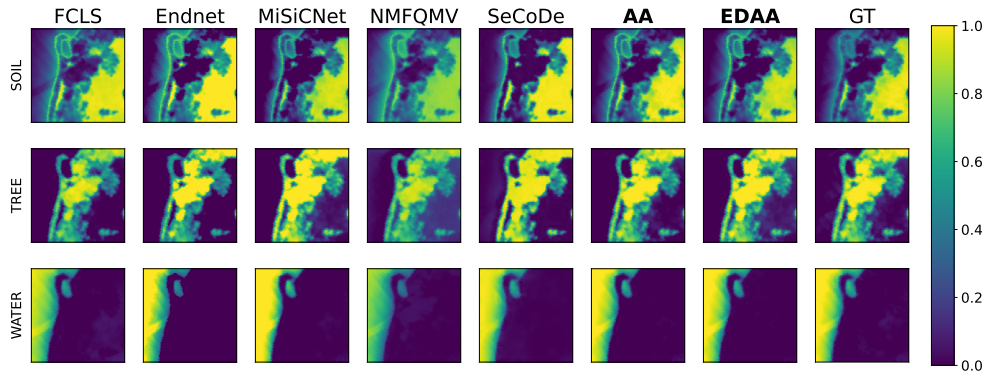
(b)

Figure 3.10: Washington DC Mall dataset: (a) False colors RGB image (Red: 150th band, Green: 75, Blue: 20) (b) ℓ_2-normalized ground truth endmembers.

Ablation study We report the detailed results obtained in the ablation study. For each dataset, the overall abundances RMSE and SAD are computed for all configurations. Table 3.6 underlines the importance of the number of runs M in the model selection procedure. Table 3.5 studies the sensitivity of the outer iterations T in EDAA when we decrease the computational budget. Finally, table 3.7 studies the sensitivity of the inner and outer iterations K_1, K_2 and T in EDAA given a fixed computational budget.

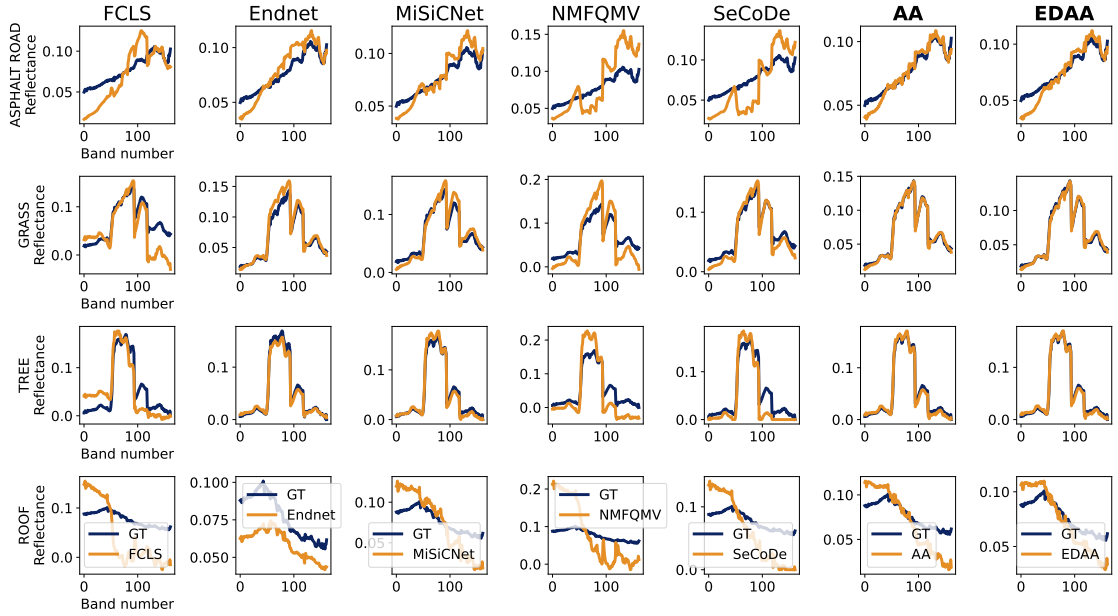


(a)

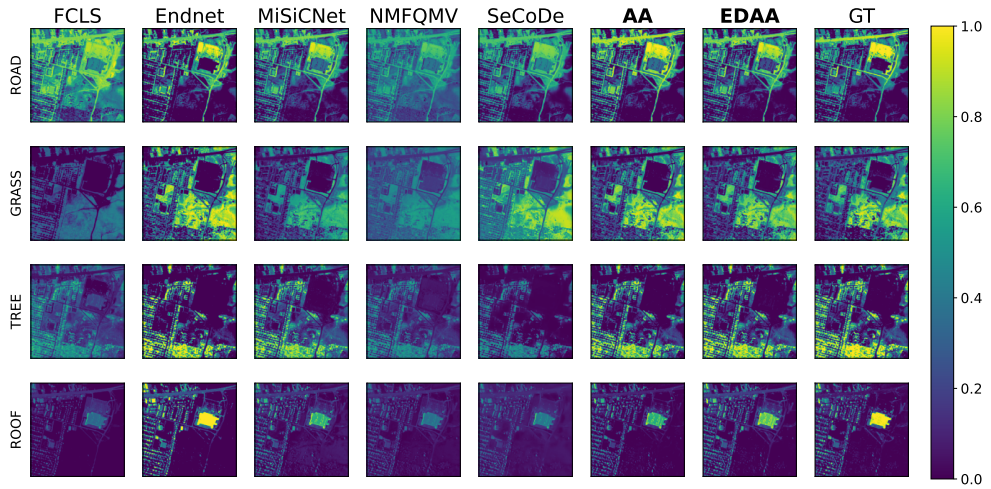


(b)

Figure 3.11: Estimated endmembers (a) and abundances (b) on the Samson dataset. Ground truth abundances are displayed on the right-most column.

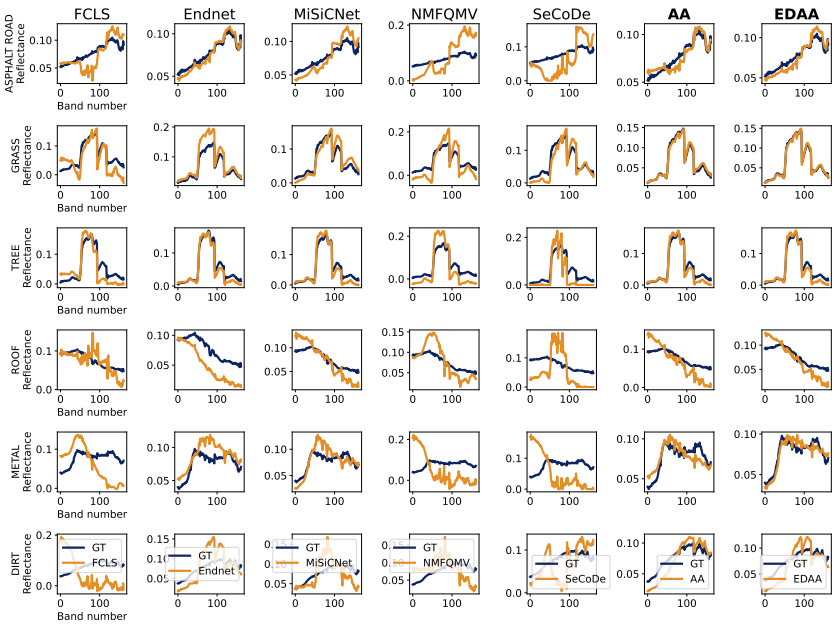


(a)

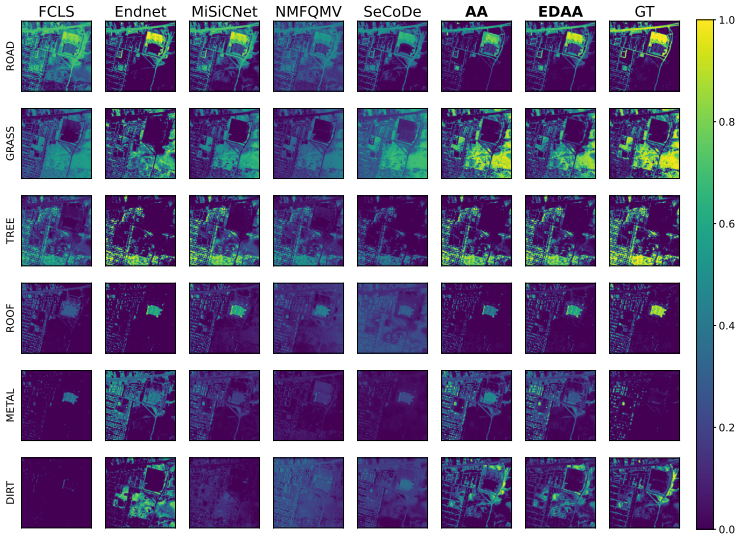


(b)

Figure 3.12: Estimated endmembers (a) and abundances (b) on the Urban4 dataset. Ground truth abundances are displayed on the right-most column.

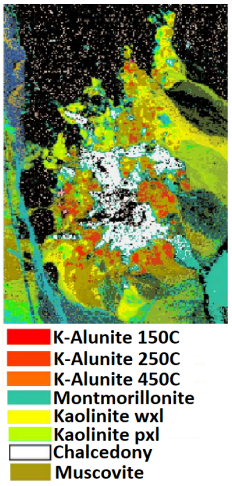


(a)

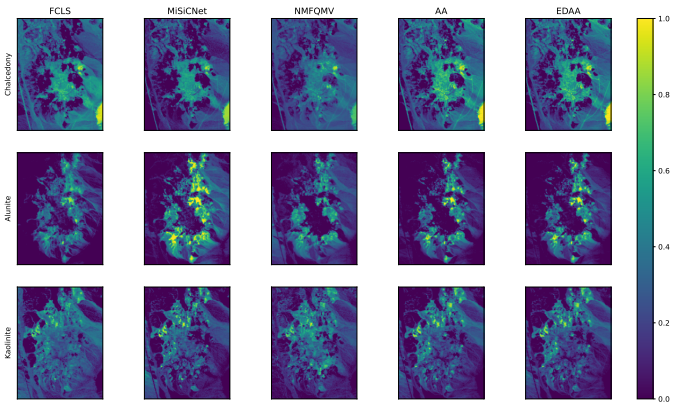


(b)

Figure 3.13: Estimated endmembers (a) and abundances (b) on the Urban6 dataset. Ground truth abundances are displayed on the right-most column.



(a) Geological Map



(b) Estimated abundance maps

Figure 3.14: Abundance maps of three dominant minerals estimated using different blind unmixing techniques applied to the Cuprite dataset.

Table 3.5: Sensitivity to the number of outer iterations T of Algorithm EDAA with $K_1 = K_2 = 5$. The abundances RMSE and SAD metrics are computed globally. The best results are in bold and the second best are underlined.

		$T = 100$	$T = 75$	$T = 50$
Samson	RMSE	<u>4.24</u>	4.34	**3.97**
	SAD	<u>1.64</u>	1.69	**1.46**
Jasper Ridge	RMSE	6.85	**5.90**	<u>6.44</u>
	SAD	3.22	**3.06**	<u>3.19</u>
Urban4	RMSE	<u>7.51</u>	**7.46**	7.72
	SAD	5.87	<u>5.01</u>	**4.67**
Urban6	RMSE	**13.52**	<u>13.54</u>	13.99
	SAD	7.95	**7.85**	<u>7.93</u>
APEX	RMSE	14.66	<u>14.38</u>	**14.12**
	SAD	**6.29**	<u>6.58</u>	6.99
WDC	RMSE	**20.47**	<u>22.81</u>	22.89
	SAD	**12.33**	16.97	<u>16.87</u>

Table 3.6: Sensitivity to the number of runs M of the model selection procedure with $K_1 = K_2 = 5$ and $T = 100$. The abundances RMSE and SAD metrics are computed globally. The best results are in bold and the second best are underlined.

		$M = 10$	$M = 20$	$M = 50$	$M = 100$
Samson	RMSE	5.05	**3.90**	4.24	4.24
	SAD	1.94	**1.43**	1.64	1.64
Jasper Ridge	RMSE	10.27	10.27	**6.85**	**6.85**
	SAD	3.73	3.73	**3.22**	**3.22**
Urban4	RMSE	8.17	8.06	7.75	**7.51**
	SAD	5.43	5.41	**5.36**	5.87
Urban6	RMSE	16.50	16.50	13.63	**13.52**
	SAD	8.16	8.16	**7.92**	7.95
APEX	RMSE	23.84	24.17	14.94	**14.66**
	SAD	12.57	12.88	**6.06**	6.29
WDC	RMSE	24.39	24.39	**20.46**	20.47
	SAD	**11.52**	**11.52**	12.40	12.33

Table 3.7: Sensitivity to hyperparameters of EDAA for a constant number of updates (1000). The abundances RMSE and SAD metrics are computed globally. The best results are in bold and the second best are underlined.

		$K_1 = K_2 = 1, T = 500$	$K_1 = K_2 = 5, T = 100$	$K_1 = K_2 = 10, T = 50$	$K_1 = K_2 = 20, T = 25$	$K_1 = K_2 = 50, T = 10$
Samson	RMSE	4.42	4.24	**4.15**	4.48	4.34
	SAD	1.65	1.64	**1.61**	1.78	1.68
Jasper Ridge	RMSE	**6.70**	6.85	7.80	7.59	8.79
	SAD	3.48	3.22	4.28	**3.12**	4.79
Urban4	RMSE	**7.40**	7.51	7.43	7.90	9.49
	SAD	5.72	5.87	5.53	6.05	**5.46**
Urban6	RMSE	17.92	13.52	**12.35**	12.71	17.92
	SAD	11.71	**7.95**	8.74	8.52	8.79
APEX	RMSE	14.25	14.66	14.20	**14.15**	16.46
	SAD	8.17	6.29	7.73	8.06	**5.26**
WDC	RMSE	22.91	**20.47**	22.16	20.92	21.20
	SAD	12.59	**12.33**	16.05	13.46	13.51

4

Sparse Unmixing using Archetypal Analysis

Chapter abstract: This chapter introduces a new sparse unmixing technique using archetypal analysis (SUnAA). First, we design a new model based on archetypal analysis. We assume that the endmembers of interest are a convex combination of endmembers provided by a spectral library and that the number of endmembers of interest is known. Then, we propose a minimization problem. Unlike most conventional sparse unmixing methods, here the minimization problem is non-convex. We minimize the optimization objective iteratively using an active set algorithm. Our method is robust to the initialization and only requires the number of endmembers of interest. SUnAA is evaluated using two simulated datasets for which results confirm its better performance over other conventional and advanced techniques in terms of signal-to-reconstruction error. SUnAA is also applied to the Cuprite dataset and the results are compared visually with the available geological map provided for this dataset. The qualitative assessment demonstrates the successful estimation of the minerals abundances and significantly improves the detection of dominant minerals compared to the conventional regression-based sparse unmixing methods.

The source code is freely available at https://github.com/inria-thoth/SUnAA.

The chapter is based on the following publication:

B. Rasti*, A. Zouaoui*, J. Mairal, and J. Chanussot. SUnAA: Sparse Unmixing using Archetypal Analysis. In *IEEE Geosciences and Remote Sensing Letters*, 2023

*equal contributions

4.1 Introduction

Spectral unmixing estimates the abundances of pure spectra of materials called endmembers. Depending on the prior knowledge available about endmembers, the unmixing problem can be divided into three main categories: (1) Supervised Unmixing, (2) Blind Unmixing and (3) Semi-supervised or Sparse Unmixing. In supervised unmixing, abundances are estimated by relying on known endmembers whereas blind unmixing estimates both the endmembers and the abundances simultaneously. Semi-supervised unmixing relies on a library of endmembers that ideally contains the endmembers present in the scene and is often formulated as a sparse regression problem, thus it is known as sparse unmixing. Abundances can typically be estimated by enforcing sparsity-promoting penalties. J. M. Bioucas-Dias originally proposed this idea in [Bioucas-Dias and Figueiredo, 2010] where sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL), and the constrained SUnSAL (C-SUnSAL) were introduced. Both SUnSAL and C-SUnSAL use the ℓ_1 penalty to promote sparsity on the abundances. SUnSAL assumes the ℓ_2 norm for the fidelity term augmented to the ℓ_1, while C-SUnSAL assumes the ℓ_2 norm as a constraint to minimize ℓ_1. SUnSAL solves the minimization problems using the alternating direction method of multipliers (ADMM) [Eckstein and Bertsekas, 1992].

SUnSAL was later improved in [Iordache et al., 2012] by adding the total variation (TV) penalty (SUnSAL-TV) to the minimization problem in order to incorporate spatial information. We should note that SUnSAL-TV does not hold the abundances sum-to-one constraint (ASC) due to the conflict with the ℓ_1 penalty. Collaborative sparse [Iordache et al., 2014a] unmixing enforces the sum of ℓ_2 norms on the abundances to promote sparsity. Double reweighted sparse unmixing [Wang et al., 2017] and spectral–spatial weighted sparse unmixing (S^2WSU) [Zhang et al., 2018a] exploit the weighted ℓ_1 norm to promote sparsity. The former also uses the TV penalty to capture spatial information. Multiscale sparse unmixing algorithm (MUA) [Borsoi et al., 2019] captures spatial correlations by performing sparse regression on segmented pixels using either a binary partition tree (BPT), the simple linear iterative clustering (SLIC), or the K-means algorithm. In [Ince, 2020], SLIC was chosen for the segmentation, and sparse unmixing was performed using superpixel-based graph Laplacian regularization. A library pruning-based sparse unmixing called multiple signal classification collaborative sparse unmixing (MUSIC-CSR) was proposed in [Iordache et al., 2014b]. The library was pruned using an orthogonal projection where HySime [Bioucas-Dias and Nascimento, 2008] was used to obtain the subspace bases to reduce the noise effect. Then collaborative sparse regression was used for abundance estimation.

A common drawback of the sparse unmixing techniques mentioned above is that the estimated fractional abundances do not necessarily describe the aerial fraction of each pure material on the ground due to the absence of ASC. Indeed, the ℓ_1 penalties cannot be applied to the abundances while holding the ASC. This issue was addressed in sparse unmixing using convolutional neural network (SUnCNN) [Rasti and Koirala, 2022]. In [Rasti and Koirala, 2022], it was shown that the problem of selecting a suitable prior for a sparse regression could be moved to the optimization on the parameters of a deep encoder-decoder network while the ASC can be enforced using a softmax layer. However, selecting suitable hyperparameters for deep networks is often challenging. In [Lin and Gader, 2022], an asymmetric encoder-decoder architecture is used for sparse unmixing. Instead of softmax, a sparse variation of softmax is used to avoid the full support of softmax while enforcing ASC.

In conventional sparse unmixing, the endmembers library is fixed, and the abundances

estimation is of interest. However, even a pruned and well-selected spectral library cannot flawlessly represent the endmembers of materials in a real-world dataset. There are several factors, such as noise, atmospheric effects, illumination variations, and the intrinsic variation of materials which may affect the endmembers and induce scaling factors for the endmembers present in the scene compared to the ones from the library. To address this issue, we assume that endmembers of interest can be modeled by a convex combination of the library endmembers. This corresponds to the formulation of archetypal analysis (AA) [Cutler and Breiman, 1994]. Recently, archetypal analysis has been successfully harnessed for blind unmixing in [Zouaoui et al., 2023]. In [Xu et al., 2022], ℓ_1 sparsity-constrained archetypal analysis was proposed for blind unmixing where the sparsity was enforced on the abundances. In this chapter, we propose solving sparse unmixing using archetypal analysis (SUnAA). In the proposed model, an additional matrix is introduced, which defines the contributions of the endmembers from the library to the estimated spectra of endmembers present in the scene. Here the ASC can be enforced but the resulting proposed minimization is jointly non-convex. The optimization problem is solved using an active set algorithm, leading to a parameter-free technique, besides the number of endmembers of interest that is required. The experimental results confirm that SUnAA outperforms conventional and deep learning-based sparse unmixing techniques in terms of signal-to-reconstruction error (SRE) for two simulated datasets and visually for the Cuprite dataset. The major contributions of this chapter are summarized as follows: 1) we propose a new model based on archetypal analysis for sparse unmixing: we assume that the unknown endmembers are a convex combination of the library endmembers, 2) we propose a non-convex optimization for sparse unmixing: unlike the conventional sparse unmixing which is based on sparse regression and convex optimization, we show that the proposed non-convex optimization leads to accurate abundances estimation, and 3) we adopt a parameter-free active set algorithm to minimize the proposed optimization problem.

4.2 Method

4.2.1 Conventional sparse unmixing

As introduced in Section 1.4, in semi-supervised unmixing, the linear mixing model states that the observed spectra are modeled as a linear combination of the library endmembers:

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{N}, \tag{4.1}$$

where $\mathbf{Y} \in \mathbb{R}^{p\times n}$ denotes the n observed spectra over p channels and $\mathbf{N} \in \mathbb{R}^{p\times n}$ is the model error and noise. $\mathbf{D} \in \mathbb{R}^{p\times m}$ denotes the spectral library containing m endmembers. $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the unknown, potentially redundant, fractional abundances to estimate. This setup is often referred to as sparse unmixing since the (redundant) fractional abundances $\mathbf{X}$ are estimated by applying sparsity-enforcing penalties/constraints, using the following sparse regression formulation:

$$\begin{aligned} \underset{\mathbf{X}}{\arg\min} \quad & \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \sum_{i=1}^{n} ||\mathbf{x}_i||_q, \\ \text{s.t.} \quad & \mathbf{X} \geq 0, \\ & \mathbf{1}_m^T \mathbf{X} = \mathbf{1}_n^T \end{aligned} \tag{4.2}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$. The ℓ_q norm is often selected to be a (weighted) sparsity promoting norm. For instance, SUnSAL [Bioucas-Dias and Figueiredo, 2010] solves problem (4.2) for $q = 1$ using ADMM [Eckstein and Bertsekas, 1992]. However, [Iordache et al., 2012] suggests using SUnSAL without ASC due to the conflict with ℓ_1.

4.2.2 SUnAA

Inspired by archetypal analysis [Cutler and Breiman, 1994], we propose a new model formulation for sparse unmixing, provided that the number of endmembers of interest, r, is known:

$$
\begin{aligned}
(\hat{\mathbf{B}}, \hat{\mathbf{A}}) = \underset{\mathbf{B},\, \mathbf{A}}{\arg\min} \quad & \|\mathbf{Y} - \mathbf{DBA}\|_F^2, && \text{(4.3a)}\\
\text{s.t.} \quad & \mathbf{B} \geq 0, && \text{(4.3b)}\\
& \mathbf{1}_m^T \mathbf{B} = \mathbf{1}_r^T, && \text{(4.3c)}\\
& \mathbf{A} \geq 0, && \text{(4.3d)}\\
& \mathbf{1}_r^T \mathbf{A} = \mathbf{1}_n^T, && \text{(4.3e)}
\end{aligned}
$$

where $\mathbf{B} \in \mathbb{R}^{m \times r}$ corresponds to the contributions of the endmembers from the library $\mathbf{D}$ and $\mathbf{A} \in \mathbb{R}^{r \times n}$ is the (low-rank) abundance matrix. We should note that AA uses $\mathbf{Y}$ instead of $\mathbf{D}$ in (4.3b). It is worth mentioning that $\mathbf{D}$ turns the blind scenario into a semi-supervised scenario, which could be more efficient in the highly mixed scenario without pure pixels.

In (4.3), we assume that the unknown endmembers of interest are a convex combination of the library endmembers. This is our primary assumption to compensate for the library mismatch. In addition, we assume that the number of endmembers is known. Therefore, we enforce non-negativity and sum-to-one constraints on $\mathbf{B}$. Equivalently to abundance non-negativity constraint (ANC 4.3d) and abundance sum-to-one constraint (ASC 4.3e), we call the constraints on $\mathbf{B}$ endmember non-negativity constraint (ENC 4.3b) and endmember sum-to-one constraint (ESC 4.3c).

It is important to note that the minimization problem (4.3) is not jointly convex in $(\mathbf{B}, \mathbf{A})$. However it is convex with respect to one of the variables when the other is fixed, hence (4.3) can be solved by alternating between two steps inside a cyclic descent scheme. First, the $\mathbf{A}$-step when $\mathbf{B}$ is fixed. Second, the $\mathbf{B}$-step, by fixing $\mathbf{A}$. In this paper, we adopt the algorithm proposed in [Chen et al., 2014]. Here, we explain briefly solutions to the sub-problems proposed. $\mathbf{A}$-step: Assuming $\mathbf{B}$ is fixed and $\mathbf{E} = \mathbf{DB}$, the sub-problem corresponds to

$$
\begin{aligned}
\hat{\mathbf{A}} = \underset{\mathbf{A}}{\arg\min} \quad & \|\mathbf{Y} - \mathbf{EA}\|_F^2.\\
\text{s.t.} \quad & \mathbf{A} \geq 0,\\
& \mathbf{1}_r^T \mathbf{A} = \mathbf{1}_n^T
\end{aligned}
\qquad \text{(4.4)}
$$

(4.4) is a smooth least-squares optimization problem with a simplicial constraint. As noted by [Chen et al., 2014], generic quadratic programming solvers could be used but significantly faster convergence can be obtained by designing a dedicated algorithm that can leverage the underlying sparsity of the solution. Following [Chen et al., 2014] we use an active set algorithm to solve (4.4). $\mathbf{B}$-step: Assuming $\mathbf{A}$ is fixed, problem (4.3) writes

as follows:

$$\begin{aligned} \hat{\mathbf{B}} = \underset{\mathbf{B}}{\arg\min} \quad & \|\mathbf{Y} - \mathbf{DBA}\|_F^2. \\ \text{s.t.} \quad & \mathbf{B} \geq 0, \\ & \mathbf{1}_m^T \mathbf{B} = \mathbf{1}_r^T \end{aligned} \tag{4.5}$$

Solving (4.5) is not as straightforward as (4.4) since it does not correspond to the standard quadratic form. However, following [Chen et al., 2014], we consider solving (4.5) separately for every column $\mathbf{b}_j$ of $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r]$ by fixing all other variables in order to obtain a quadratic program:

$$\begin{aligned} \underset{\mathbf{b}_j}{\arg\min} \quad & \left|\left| \frac{1}{\|\mathbf{a}^j\|_2^2} (\mathbf{Y} - \mathbf{DB}_{\text{old}}\mathbf{A})\mathbf{a}^{j\top} + \mathbf{Db}_{j,\text{old}} - \mathbf{Db}_j \right|\right|_F^2, \\ \text{s.t.} \quad & \mathbf{b}_j \geq 0, \\ & \mathbf{1}_m^T \mathbf{b}_j = 1 \end{aligned} \tag{4.6}$$

where $\mathbf{b}_{j,\text{old}}$ is the current value of $\mathbf{b}_j$ before the update, and $\mathbf{a}^j$ in $\mathbb{R}^{1\times n}$ is the j-th row of $\mathbf{A}$.

Note that the first and second terms are fixed, therefore the same active set algorithm can be used to solve (4.6). The pseudo-code for the algorithm used to solve (4.3) is given in Algorithm 3. As for initialization, we uniformly initialize matrices $\mathbf{B}$ and $\mathbf{A}$, *i.e.*,$\mathbf{B}^{(0)} = (\mathbf{1}_m \mathbf{1}_r^T)/m$ and $\mathbf{A}^{(0)} = (\mathbf{1}_r \mathbf{1}_n^T)/r$.

Algorithm 3 SUnAA

Input: $\mathbf{Y}$ (HS data); $\mathbf{D}$ (Endmember library); r (number of endmembers); T (number of iterations).
Initialization: $\mathbf{B}^{(0)} \leftarrow (\mathbf{1}_m \mathbf{1}_r^T)/m$ and $\mathbf{A}^{(0)} \leftarrow (\mathbf{1}_r \mathbf{1}_n^T)/r$
for $t = 1, \ldots, T$ **do**
  $\mathbf{E} \leftarrow \mathbf{DB}$
  $\hat{\mathbf{A}} = \text{ActSet}(\mathbf{Y}, \mathbf{E})$ ▷ **A-step**
  for $j = 1, \ldots, r$ **do** ▷ **B-step**
    $\tilde{\mathbf{Y}} \leftarrow \frac{1}{\|\mathbf{a}^j\|_2^2} (\mathbf{Y} - \mathbf{DB}_{\text{old}}\mathbf{A})\mathbf{a}^{j\top} + \mathbf{Db}_{j,\text{old}}$
    $\hat{\mathbf{b}}_j \leftarrow \text{ActSet}(\tilde{\mathbf{Y}}, \mathbf{D})$
  end for
end for
$\hat{\mathbf{E}} \leftarrow \mathbf{D}\hat{\mathbf{B}}$
Return: $\hat{\mathbf{A}}$: Abundances, $\hat{\mathbf{E}}$: Endmembers, $\hat{\mathbf{B}}$: Endmembers contributions.

4.3 Experiments

We compare SUnAA with seven sparse unmixing techniques: SUnSAL [Bioucas-Dias and Figueiredo, 2010], SUnSAL-S (SUnSAL with ASC and without sparsity penalty, *i.e.*,$\lambda = 0$), SUnSAL-TV [Iordache et al., 2012], S^{2}WSU [Zhang et al., 2018b], MUA (using BPT segmentation) [Borsoi et al., 2019], MUSIC-CSR [Iordache et al., 2014b], and SUnCNN [Rasti and Koirala, 2022] applied to two simulated datasets and Cuprite. All the parameters are set as default for the competing methods. The results are mean values over ten experiments.

4.3.1 Simulated datasets

Two synthetic datacubes (DC1 and DC2) were used for simulated experiments. For DC1, a synthetic library composed of 240 spectral signatures selected from the USGS library with a minimum pair-spectra angle of 4.44°is used. DC1 was simulated using a linear mixing model with 5 endmembers selected from the library and 75×75 pixels. The abundance maps are composed of five rows of square regions uniformly distributed over the spatial dimension. DC2 is a challenging simulated dataset with no pure pixels and two mixed pixels on the facet of the data simplex (more details in [Rasti et al., 2022]). It contains 105×105 pixels, simulated by the linear combination of six endmembers from the USGS library. Two endmembers show no absorption features throughout the wavelength and are similar but scaled versions of each other, making the scenario even more challenging.

For the quantitative evaluation, we use the SRE in dB

$$\mathrm{SRE}(\mathbf{X}, \hat{\mathbf{X}}) = 20 \log_{10} \frac{\|\mathbf{X}\|_F}{\|\mathbf{X} - \hat{\mathbf{X}}\|_F}. \tag{4.7}$$

Three levels of additive noise, *i.e.,*20, 30, and 40 dB are considered in the simulated experiments.

Tables 4.1 and 4.2 compare the results obtained by applying different sparse unmixing techniques to DC1 and DC2, respectively, in terms of SRE. For DC1, SUnAA outperforms the other techniques for SNR=30 and 40 dB, and for SNR=20 dB gives the second-best results after MUA. SUnCNN shows the second-best performance in terms of SRE. MUA shows the best performance for low SNR, *i.e.,*20 dB. However, it performs poorly for high SNRs. SUNSAL and SUnSAL-S yield very low SREs. S^2WSU, MUSIC-CSR and SUnSAL-TV perform moderately, however, S^2WSU provides better performances for higher SNRs.

The results for DC2 show that SUnAA outperforms the other techniques for all SNRs considerably. All the other techniques give low SRE. This could be attributed to the complexity of the dataset. Since the library may have several scaled versions of one endmember, the conventional sparse regression might not lead to a sparse solution. A remedy to this problem is to prune the library by removing the spectra with small spectral angle distances. However, in the case of complex datasets such as DC2, the endmembers may be removed from the library, which also leads to poor estimation.

Overall, comparing the SRE reveals that SUnAA outperforms the other techniques. SUnAA also show consistent performances with respect to the noise level and datasets. All the other methods fail for the complex dataset. For the simple datasets, SUnCNN gives the second-best performance. MUA outperforms the other techniques for very low SNR, *i.e.,*20 dB, which could be attributed to the segmentation applied before the sparse regression in MUA. However, this might cause oversmoothing in the abundances. S^2W performs moderately and better than SUnSAL models. SUnSAL-TV outperforms SUnSAL and SUnSAL-S due to the TV penalty, which exploits spatial information.

Fig. 4.1 and 4.2 provide a visual comparison of the abundance maps estimated for endmember 1 of DC1 and DC2, respectively. As can be seen, both MUA and SUnSAL-TV oversmooth the abundances, particularly for low-SNRs. MUA also introduces artifacts, which is not desirable. This is due to the segmentation step in MUA and the absence of parameter selection for the TV regularizer in SUnSAL-TV.

Table 4.1: Sparse unmixing experiments applied to DC1 in terms of SRE. The best performances are shown in bold. The second best are underlined.

SNR	SUnSAL	SUnSAL-S	SUnSAL-TV	S^2WSU	MUA	MUSIC-CSR	SUnCNN	SUnAA
20 dB	4.86	4.31	9.76	7.99	**13.19**	6.07	11.15	<u>11.52</u>
30 dB	8.94	8.47	14.39	15.52	18.28	13.36	<u>20.63</u>	**21.27**
40 dB	13.83	13.15	20.84	28.16	21.12	24.39	<u>30.62</u>	**31.23**

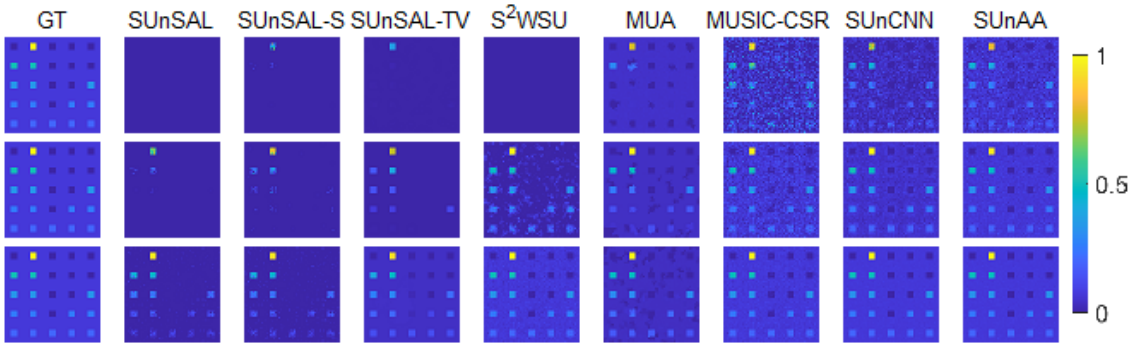


Figure 4.1: The fractional abundance of endmember 1 of DC1. From top to bottom SNR= 20, 30, and 40 dB.

Table 4.2: Sparse unmixing experiments applied to DC2 in terms of SRE. The best performances are shown in bold. The second best are underlined.

SNR	SUnSAL	SUnSAL-S	SUnSAL-TV	S^2WSU	MUA	MUSIC-CSR	SUnCNN	SUnAA
20 dB	3.04	1.41	2.49	2.76	<u>6.95</u>	2.51	4.45	**9.54**
30 dB	3.72	2.42	<u>7.42</u>	6.57	6.90	4.43	5.08	**10.76**
40 dB	5.96	3.74	6.76	<u>7.43</u>	7.31	5.45	5.96	**11.74**

4.3.2 Cuprite dataset

We used a subset of 250×191 pixels of the Cuprite dataset for real-world experiments. The minerals in that region are well-studied and are therefore suitable for evaluating the abundance maps qualitatively. Fig. 3.14 (a) depicts the geological ground reference for the dominant minerals. We use a library $\mathbf{D} \in \mathbb{R}^{188\times498}$, which is composed of 498 spectral pixels from the USGS library. The water absorption and noisy bands were removed, hence the final pixels are of dimension $p = 188$.

Fig. 4.3 (b) demonstrates the abundance maps estimated by using different unmixing techniques applied to Cuprite. We showed three dominant minerals in the scene, *i.e.,*Chalcedony, Alunite, and Kaolinite, corresponding to library endmembers 297, 420 and 465. For SUnAA, those abundances appear as 15, 13, and 11th. Note that we select $r = 16$.

Fig. 4.3 (b) shows that all conventional sparse unmixing techniques and SUnCNN perform similarly. It can be observed that SUnSAL-TV and MUA oversmooth the mineral abundances, which could be attributed to the total variation penalty in SUnSAL-TV and the segmentation-based framework in MUA, which cannot preserve the textures. On the other hand, SUnAA provides sharper maps. Compared to the geological map of USGS (Fig. 4.3 (a)), SUnAA considerably outperforms the other techniques for the detection of Chalcedony and Alunite. In the case of Kaolinite, all techniques perform similarly,

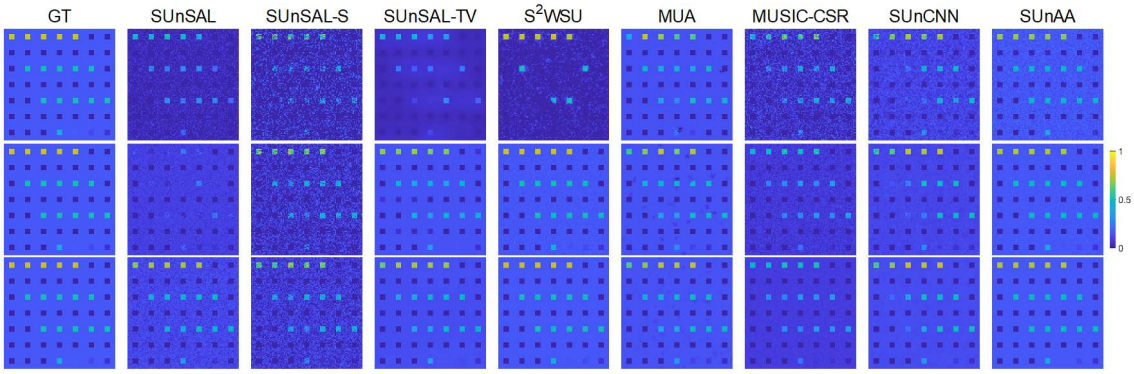
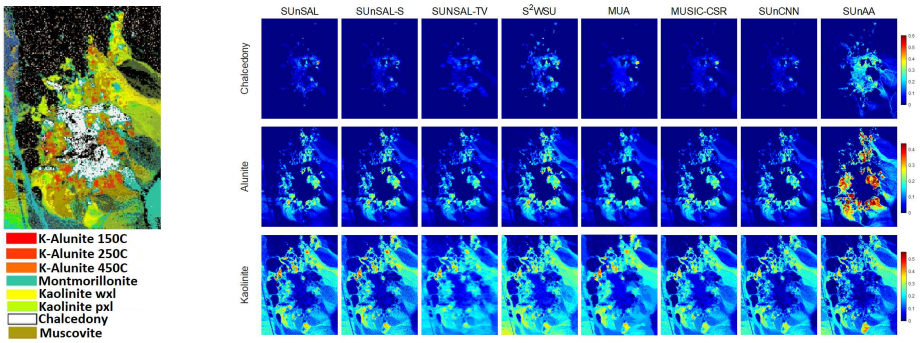


Figure 4.2: The fractional abundance of endmember 1 of DC2. From top to bottom SNR= 20, 30, and 40 dB.

however, SUnAA shows slightly better performance compared to the other techniques, particularly for the southern region.

The substantial improvements of SUnAA can be attributed to the proposed model formulation leveraging archetypal analysis. Fig. 4.4 depicts the estimated endmembers corresponding to the three minerals (Chalcedony, Alunite, and Kaolinite). Comparing the estimated endmembers with the corresponding ones from the library, *i.e.,*297, 420 and 465, reveals that they are scaled versions. In real-world applications, the captured datasets are affected by noise, atmospheric effects, illumination variations, and the intrinsic variation of materials [Borsoi et al., 2021]. Therefore, expecting the measured endmembers from a library to represent the materials in a real-world dataset is unrealistic. On the other hand, using archetypal analysis, we can achieve a linear combination of the endmembers from the dictionary to better represent the endmembers present in the scene.

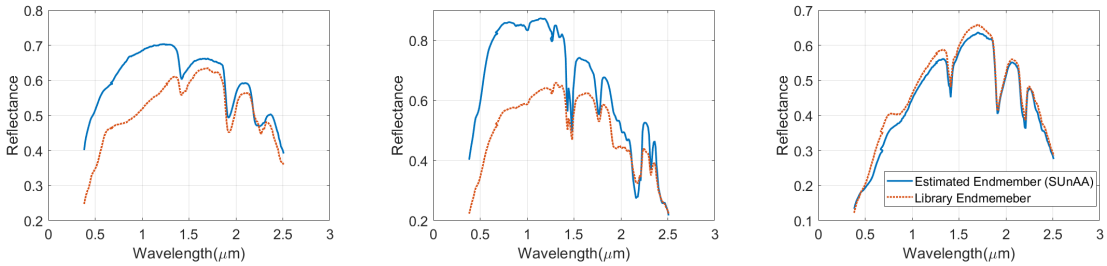


(a) Geological Map (b) Estimated abundance maps

Figure 4.3: Abundance maps of three dominant minerals estimated using different sparse unmixing techniques applied to the Cuprite dataset.

4.4 Conclusion

We proposed a sparse unmixing technique using archetypal analysis called SUnAA. SUnAA models the endmembers of interest as a convex combination of endmembers from a library. We proposed a non-convex optimization to simultaneously estimate the endmembers contributions and abundances. The proposed iterative algorithm is based



(a) Chalcedony (b) Alunite (c) Kaolinite

Figure 4.4: Endmembers of three dominant minerals estimated using SUnAA applied to the Cuprite dataset compared with the ones from USGS library.

on an active set method and is parameter-free. We evaluated SUnAA on two simulated and Cuprite datasets. The experimental results confirm that SUnAA leads to accurate abundances estimation and significantly outperforms the conventional sparse unmixing techniques. Additionally, experiments on a real-world dataset suggest that SUnAA can better detect existing minerals in a given scene.

APPENDIX

4.5 Urban dataset

The Urban hyperspectral image contains 307x307 pixels captured by the Hyperspectral Digital Image Collection Experiment (HYDICE) [Rickard et al., 1993] sensor in 210 specrtral bands ranging from 400 to 2500 nm. After removing 48 noisy and water absorption bands 162 bands remain. The dataset have 4 endmembers, Asphalt Road, Grass, Tree, and Roof. The library $\mathbf{D} \in \mathbb{R}^{162\times402}$ is composed of 402 spectral pixels obtained from the dataset.

Table 4.3 compares the sparse unmixing experiments applied to Urban in terms of SRE. As can be seen, SUnAA and SUnCNN perform similarly and outperform the other techniques. SUnSAL-S performs better than the other methods for this dataset. Fig. 4.5 visually compares the estimated abundances using different sparse unmixing techniques with the ground reference map.

We should note that, in this experiment, we use $r = 5$ for SUnAA. We notice that the library includes endmembers close to the zero endmember, often used for shadow removal, and SUnAA picks this endmember for $r = 4$. Therefore, we select $r = 5$, and since the ground truth abundances do not include the shadow abundance, we remove this from the abundances and normalize the abundances to be sum to one.

Table 4.3: Sparse unmixing experiments applied to Urban in terms of SRE. The best performances are shown in bold. The second best are underlined.

SNR	SUnSAL	SUnSAL-S	SUnSAL-TV	S^2WSU	MUA	MUSIC-CSR	SUnCNN	SUnAA
Urban	1.73	7.10	2.07	1.58	2.65	2.24	<u>7.74</u>	**7.77**

4.6 Processing time

Table 4.4 gives the processing times for different unmixing techniques applied to Urban and Cuprite datasets. SUnSAL, SUnSAL-S, SUnSAL-TV, S^2WSU, and MUA were implemented in Matlab (2020b) and MUSIC-CSR, SUnCNN, and SUnAA were implemented in Python (3.8). The reported processing times were obtained using a computer with Intel(R) Xeon(R) Silver 4112 CPU 2.60GHz, eight cores, 64GB of memory, and an NVIDIA TITAN RTX with 24GB of memory. The table shows that MUA is the most and SUnSAL-TV is the least efficient method for both datasets. SUnAA is competitive to MUA in the case

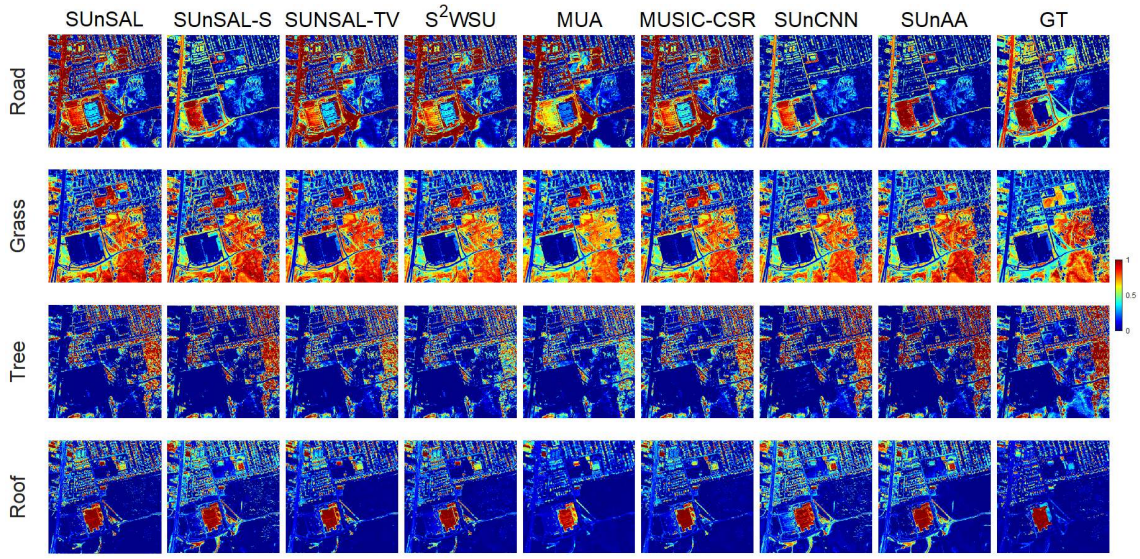


Figure 4.5: The fractional abundance of Urban datasets using different sparse unmixing techniques.

of Urban and outperforms the other techniques. However, in the case of Cuprite, SUnAA is not as efficient. We should note that the increase in the processing time of SUnAA is due to the column-wise update of matrix $\mathbf{B}$. Therefore, the higher number of endmembers will decrease the algorithm's speed.

Table 4.4: Processing times of different sparse unmixing techniques applied to real datasets.

HS	SUnSAL	SUnSAL-S	SUnSAL-TV	S^2WSU	MUA	MUSIC-CSR	SUnCNN	SUnAA
Urban	672.5	1550.4	2107.9	606.5	**216.0**	1975.1	320.8	<u>306.2</u>
Cuprite	224.0	<u>203.4</u>	1109.2	389.8	**160.8**	318.4	904.8	703.5

5

HYPERSPECTRAL UNMIXING PYTHON PACKAGE

Chapter abstract: Spectral pixels are often a mixture of the pure spectra of the materials, called endmembers, due to the low spatial resolution of hyperspectral sensors, double scattering, and intimate mixtures of materials in the scenes. Unmixing estimates the fractional abundances of the endmembers within the pixel. Depending on the prior knowledge of endmembers, linear unmixing can be divided into three main groups: supervised, semi-supervised, and unsupervised (blind) linear unmixing. Advances in image processing and machine learning substantially affected unmixing. This chapter provides a critical comparison between advanced and conventional techniques from the three categories. We compare the performance of the unmixing techniques on three simulated and two real datasets. The experimental results reveal the advantages of different unmixing categories for different unmixing scenarios. Moreover, we provide an open-source Python-based package, HySUPP, to reproduce the results.

The source code is freely available at https://github.com/inria-thoth/HySUPP.

The chapter is based on the following publication:

B. Rasti, A. Zouaoui, J. Mairal and J. Chanussot. Image Processing and Machine Learning for Hyperspectral Unmixing: An Overview and the HySUPP Python Package. *arXiv preprint arXiv:2308.09375*, 2023

5.1 Introduction

5.1.1 Key trends in hyperspectral unmixing

Hyperspectral unmixing has undergone various developmental stages, as illustrated in figure 5.1. In its early years, from 1992 to 2008, this research topic received minimal attention, with an annual publication count of no more than 18. However, starting from 2009 and continuing until 2016, there was a steady increase in interest, with the number of publications even surpassing 200 in 2016. Since 2017, hyperspectral unmixing has remained a prominent and dynamic area of research, driven by the emergence of deep learning techniques for unmixing. In 2022, this field reached its fourth most prolific year on record, with 164 publications.

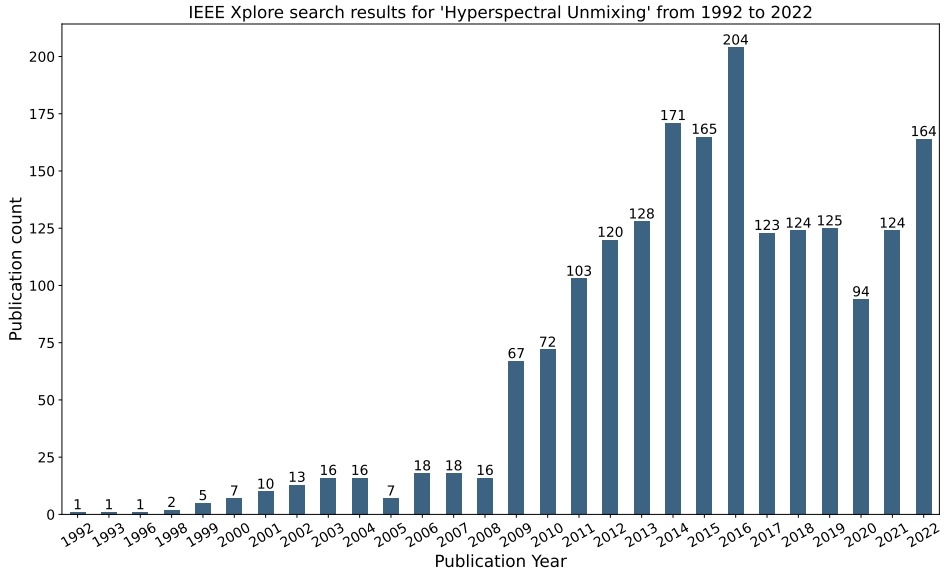


Figure 5.1: Publications over time based on IEEE Xplore keyword search tool using “Hyperspectral Unmixing" as input.

During the period when hyperspectral unmixing gained popularity, the primary scientific programming language of choice was MATLAB, as depicted in figure 5.2. However, in recent times, there has been a steady increase in interest and adoption of Python, while MATLAB’s popularity has waned. This shift can be attributed, in part, to the growing trend of open science, where open-sourcing scientific code has become widely embraced. Consequently, the present scenario suggests that developing and releasing an open-source unmixing package in Python is more advantageous than doing so in MATLAB, even though it may involve the translation of existing unmixing code from MATLAB to Python.

5.1.2 Existing overview publications and unmixing packages

An early survey on hyperspectral unmixing was given in [Keshava and Mustard, 2002], which discusses basic geometrical and statistical methods. In [Bioucas-Dias et al., 2012], linear model-based unmixing techniques were divided into three categories: geometrical, statistical, and sparse regression-based approaches. A Matlab toolbox is available at[1].

[1] https://openremotesensing.net/

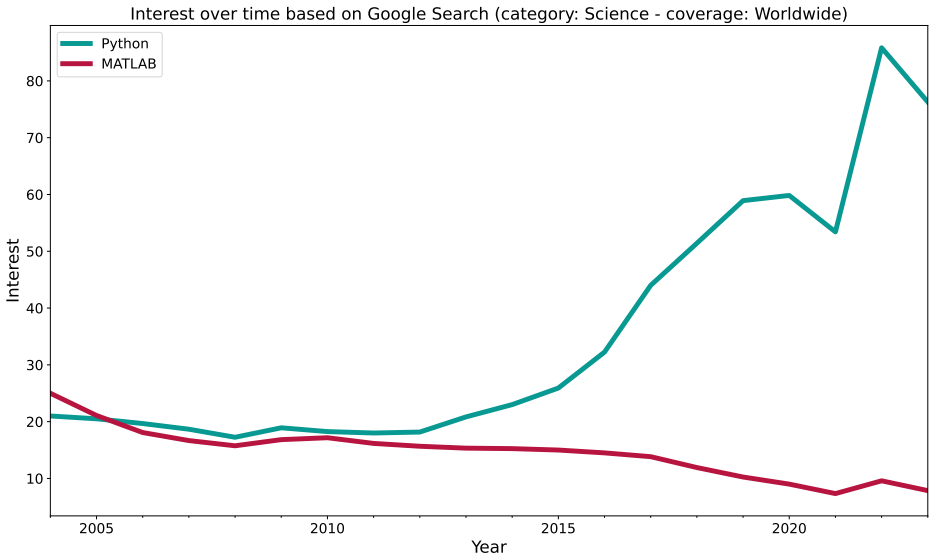


Figure 5.2: Worldwide interest in scientific programming languages over time according to Google Trends in the “Science" category.

However, the toolbox is incomplete, and some methods, such as dependent component analysis (DECA), are missing. In recent years, deep learning and neural networks have become state-of-the-art for many tasks in machine learning and image processing. Consequently, many unmixing approaches were proposed based on shallow and deep neural networks. A comparison of autoencoder-based networks was drawn in [Palsson et al., 2022]. The authors discussed autoencoder-based architectures divided into five categories, i.e., Sparse Nonnegative Autoencoders, Variational Autoencoders, Adversarial Autoencoders, Denoising Autoencoders, and Convolutional Autoencoders. They further discuss the choice of different modules, such as different activation functions or loss functions, and they compared shallow networks to deep ones. They also provide a TensorFlow-based Python package that is available on GitHub. However, the package is limited to blind unmixing approaches based on autoencoders. It does not discuss or compare the supervised, semi-supervised, and more conventional blind unmixing methods.

In [Chen et al., 2023], some model-based and neural network-based unmixing approaches were explained but without experimental comparisons. A list of resources for the approaches was given.

A survey on endmember variability in Spectral Mixture Analysis (SMA) was given in [Somers et al., 2011]. In [Zare and Ho, 2013], an overview of unmixing methods that address endmember variability was given. A comprehensive overview of the unmixing methods that address spectral variability was recently provided in [Borsoi et al., 2021], and a list of Matlab codes was also given. In [Plaza et al., 2011], an overview of endmember extraction approaches was given. Review papers on hyperspectral remote sensing data analysis briefly discussed the unmixing methods [Bioucas-Dias et al., 2013, Ghamisi et al., 2017]. In [Heylen et al., 2014], a survey of nonlinear unmixing methods is given.

There are other open-source tools such as HyperMix [Jimenez and Plaza, 2015], Spec-

tral Python (SPy) [2], Spectral Library Tool [3], PySptools[4], that include basic algorithms for estimation of the number of endmembers, endmember extraction, abundance estimation, and some library tools, and library-based methods. Thus, there is a need for a comprehensive package that covers the methodologies across different unmixing categories and contains state-of-the-art image processing and machine learning techniques.

5.1.3 Contributions

This chapter covers the following:

- Categorizing the unmixing approaches depending on the prior knowledge available about endmembers. Linear unmixing can be divided into three main categories: supervised, semi-supervised (library-based), and unsupervised (blind) unmixing.
- Comparing the unmixing methods in terms of prior knowledge of the endmembers and draw conclusions that can help researchers to select an appropriate unmixing method to tackle real-world challenges. We compare conventional and deep learning-based unmixing approaches in those categories for three simulated and two real-world datasets. For the simulated datasets, we consider three scenarios: a simple, pure pixel dataset, a dataset with spectral variations, and a challenging dataset with no pure pixel. Such comparisons provide insight to researchers into which category to use for their application. Additionally, the comparisons reveal the drawbacks of the categories, which motivate the developers to investigate new ideas to address them.
- Providing an open-source HyperSpectral Unmixing Python Package (HySUPP). HySUPP is the first open-source python-based hyperspectral unmixing package to include supervised, semi-supervised, and blind unmixing methods. The package will benefit the geoscience and remote sensing community, including researchers, developers, lecturers, and students. The package installation is straightforward since HySUPP relies on a few dependencies. In addition, all the methods can be run using a single command line instruction.

5.2 HySUPP Toolbox

HySUPP exhibits a list of highly desirable properties summarized as follows: i) completeness, ii) reproducibility, iii) extensibility and iv) homogeneity. It implements common best practices and enables simple benchmarking of unmixing techniques thanks to user-friendly command line instructions.

5.2.1 Features

Completeness As a practitioner, the ability to explore and experiment with different unmixing techniques is crucial since no single approach can consistently outperform others in all unmixing scenarios. Thus, ensuring the completeness of our toolbox becomes essential. Our toolbox is designed to cover all three types of unmixing - supervised,

[2] https://www.spectralpython.net/
[3] https://spectral-libraries.readthedocs.io/en/latest/
[4] https://pypi.org/project/pysptools/

semi-supervised, and unsupervised - while striving to be as representative as possible of the various unmixing approaches, although aiming for exhaustive inclusion would be impractical. Currently, HySUPP provides access to a diverse set of 20 different unmixing methods, including 6 supervised, 6 semi-supervised, and 8 unsupervised techniques.

Reproducibility Ensuring experimental reproducibility is crucial when exploring various unmixing techniques, as it guarantees the robustness of the conclusions drawn by users. In line with this principle, our toolbox offers the ability to seed experiments, facilitating reproducibility through repeatable noise generation. Additionally, HySUPP automatically saves estimates outputs, providing users with a convenient way to review and compare results, thus enhancing the reliability of their research findings.

Extensibility HySUPP's architecture is designed to support the effortless integration of new methods, providing a platform for future advancements in hyperspectral unmixing. Leveraging configuration files, we empower researchers to conduct experiments with ease and flexibility, enabling them to effortlessly incorporate their own models into the toolbox.

Homogeneity The uniformity of HySUPP's codebase ensures consistency across different components, enhancing its overall usability and reliability. More specifically, we use common methods for models and common attributes for datasets.

Best practices We offer a straightforward yet potent method to monitor the objective function of each approach using Python's `tqdm` library. Furthermore, our pipeline incorporates precise endmember auto-alignment through the utilization of the `munkres` algorithm, provided ground truth abundance maps are accessible. This enhancement ensures the accurate computation of unmixing performance. By establishing these best practices, we streamline the implementation process and foster a collaborative environment where researchers can easily build upon existing work and share their contributions effectively.

Benchmarking Owing to its rich set of features, HySUPP enables users to easily benchmark all implemented methods on their dataset of choice. Our toolbox currently provides 3 synthetic datasets corresponding to different unmixing scenarios. Moreover, we incorporate 4 distinct metrics to thoroughly evaluate unmixing accuracy across methods. Finally, leveraging `mlxp`[Arbel, 2023] results query tool, we empower users to analyze and visualize their results in an appealing and informative manner.

5.2.2 Example

The following command line instruction provides an example on how to run a semi-supervised unmixing technique, `SUnCNN`[Rasti and Koirala, 2022], on the `DC1` dataset using an optional custom value for the signal-to-noise ratio (SNR):

```
$ python unmixing.py mode=semi data=DC1 model=SUnCNN noise.SNR=30
```

Table 5.1 lists unmixing methods included in HySUPP with their corresponding dependencies. We highlighted our main contributions and the link to the original implementations.

Table 5.1: The list of unmixing methods included in HySUPP with their corresponding dependencies. We highlighted our main contributions and the link to the original implementations.

Method	Source	Python	Dependencies	GPU	Contributions
FCLSU[Heinz and Chein-I-Chang, 2001]	pysptools	✓	`numpy`, `cvxopt`		Refactor into a `SupervisedUnmixingModel` sub-class
SiVM[Heylen et al., 2011]	github	✓	`numpy`		Refactor into a `BaseExtractor` sub-class
SISAL[Bioucas-Dias, 2009]	github	✓	`numpy`		Refactor into a `BaseExtractor` sub-class
UnDIP[Rasti et al., 2021]	github	✓	`torch`	✓	Refactor separate scripts into a single model
VCA[Nascimento and Bioucas-Dias, 2003]	github	✓	`numpy`		Replace `scipy` dependency by `numpy`. Add random seed
CLSUnSAL[Iordache et al., 2014a]	github	✓	`numpy`		Implementation based on SUnSAL
MUA_SLIC[Borsoi et al., 2019]	github	✓	`numpy`, `skimage`		MATLAB code translation using `skimage`'s SLIC
S^{2}WSU[Zhang et al., 2018b]	github	✓	`numpy`, `scipy`		MATLAB code translation
SUnAA[Rasti et al., 2023b]	github	✓	`numpy`, `spams`		Refactor to match `SemiUnmixingModel` sub-class
SUnCNN[Rasti and Koirala, 2022]	github	✓	`torch`	✓	Refactor separate scripts into a single model
SUnSAL[Bioucas-Dias and Figueiredo, 2010]	github	✓	`numpy`		Refactor into `SemiUnmixingModel` sub-class
ADMMNet[Zhou and Rodrigues, 2021]	-	✓	`torch`	✓	Implemented from scratch
BayesianSMA[Dobigeon et al., 2009]	webpage	`matlab.engine`	`numpy`		Python wrapper around existing MATLAB code
CNNAEU[Palsson et al., 2020]	github	✓	`torch`	✓	Convert existing `keras` implementation into `torch`
EDAA[Zouaoui et al., 2023]	github	✓	`torch`	✓	Refactor to match `BlindUnmixingModel` sub-class
MiSiCNet[Rasti et al., 2022]	github	✓	`torch`	✓	Refactor separate scripts into a single model
MSNet[Yu et al., 2022]	github	✓	`torch`	✓	Refactor to match `BlindUnmixingModel` sub-class
NMFQMV[Zhuang et al., 2019]	github	`matlab.engine`	`numpy`		Python wrapper around existing MATLAB code
PGMSU[Shi et al., 2021b]	github	✓	`torch`	✓	Refactor to match `BlindUnmixingModel` sub-class

5.3 Experiments

We use three simulated and one real datasets. The simulated datasets were designed according to different mixing scenarios briefly explained in Table 5.2. We avoid using the widely used benchmark datasets such as Samson and Jasper since their abundances are generated synthetically. For real-world experiments, we use the Cuprite dataset, a well-studied site with geological reference maps. The simulated experiments were carried out for 30 dB SNR. The tuning parameters were fine-tuned for the methods up to some levels. We should note that some methods have several parameters to be tuned; therefore, searching for the optimum is cumbersome. All the results are averaged over 10 experiments, and the standard deviations are shown by error bars. In experiments, we compare 20 unmixing methods from different categories as follows. For supervised methods, we used three endmember extraction/estimation techniques, i.e., VCA, SiVM, and SISAL with FCLSU and UnDIP. All six combinations of them were considered. For blind unmixing, we use PGMSU, MSNET, CNNAEU, ADMMNet, BayesianSMA, NMFQMV, MiSiCNet, and EDAA. For sparse unmixing, we used SUnSAL, CLSUnSAL, MUA_SLIC, S^{2}WSU, SUnCNN, and SUnAA. The codes for all those methods were provided in HySUPP for reproducibility.

For the quantitative evaluation, we use the SRE in dB for estimated abundances given by

$$\mathrm{SRE}(\mathbf{X}, \hat{\mathbf{X}}) = 20 \log_{10} \frac{\|\mathbf{X}\|_F}{\|\mathbf{X} - \hat{\mathbf{X}}\|_F}. \tag{5.1}$$

5.3.1 Data description

Synthetic Datasets with Spatial Structure

We simulated three data cubes (DC1, DC2, and DC3). DC1 was simulated using a linear mixing model with 5 endmembers selected from the USGS library and 75×75 pixels. The abundance maps are composed of five rows of square regions uniformly distributed over the spatial dimension. This dataset contains pure pixels for all endmembers. DC2 has 100×100 pixels and was simulated using a linear mixing model with 9 endmembers. The abundance maps were sampled from a Dirichlet distribution centered at a Gaussian random field to have piece-wise smooth maps with steep transitions. Therefore, DC2 contains

spectral variations. For DC1 and DC2, an endmember library $\mathbf{D} \in \mathbb{R}^{188\times240}$, composed of 240 spectral signatures were selected from the USGS library with a minimum pair-spectra angle of 4.44°. DC3 was simulated with no pure pixels, and it has two mixed pixels on the facet of the data simplex. DC3 is composed of 105×105 pixels using the linear combination of six endmembers. For DC3, we use a library $\mathbf{D} \in \mathbb{R}^{188\times498}$ composed of 498 spectral pixels from the USGS library. Note that we remove the water absorption and noisy bands, and therefore, the final pixels are of dimension $p = 188$.

Cuprite Dataset

The Cuprite dataset used in this paper contains 250× 191 pixels. Cuprite is a well-studied mineral site, and the dominant minerals are demonstrated using a geological ground reference therefore, the estimated abundance maps by different techniques can be compared visually. We use the same library as explained for DC3.

Table 5.2: Specifications of the synthetic datasets used in the experiments.

	# endmembers (r)	# atoms in $\mathbf{D}$ (m)	# pixels (n)	Main features
DC1	5	240	5625 (75 × 75)	Pure pixels
DC2	9	240	10000 (100 × 100)	Pure pixels, Spectral variability
DC3	6	498	11025 (105 × 105)	No pure pixels, 2 points on the data simplex facets

5.3.2 Synthetic datasets

Figure 5.3 demonstrates the unmixing results in terms of abundance SRE in DB for different techniques applied to DC1, DC2, and DC3. The outcomes of the results can be summarized as follows:

Presence of pure pixels For the pure pixel scenario, supervised techniques perform very well (Fig. 5.3, DC1). Overall, they perform better than the semisupervised and blind techniques. This confirms the importance of geometrical information for endmember extraction/ estimation techniques when there are pure pixels in the dataset. This is further confirmed in blind methods where MiSiCNet and NMFQMV, which exploit geometrical information, outperform the other blind techniques and provide competitive results compared to supervised ones. Sparse techniques show moderate results except SUnAA, which outperforms all the other techniques. The results confirm that the sparse unmixing techniques are not suitable when there are pure pixels for the endmembers in the dataset. We should mention that SUnAA does not match the characteristics of conventional sparse techniques. Even though SUnAA relies on a library, it uses a nonconvex optimization to estimate endmembers.

Spectral variability On the other hand, for DC2, which contains spectral variations, as can be seen from Fig. 5.3, sparse unmixing techniques outperform the supervised and blind techniques. The results confirm that sparse unmixing techniques are more suitable for capturing the spectral variability. SUnAA outperforms the other technique. Note that SUnAA is a parameter-free technique. Supervised techniques outperform blind techniques due to the presence of pure pixels.

Absence of pure pixels In the case of missing pure pixels (Fig. 5.3, DC3), blind unmixing techniques that exploit geometrical information outperform the other techniques. Sparse unmixing provides very poor results. Although SUnAA considerably significantly outperforms the performance of sparse unmixing techniques, it is very far from the best performance which is obtained by MiSiCNet. Among supervised techniques, SISAL (+ FCLSU/UnDIP) shows competitive results because it uses geometrical information to estimate the endmembers.

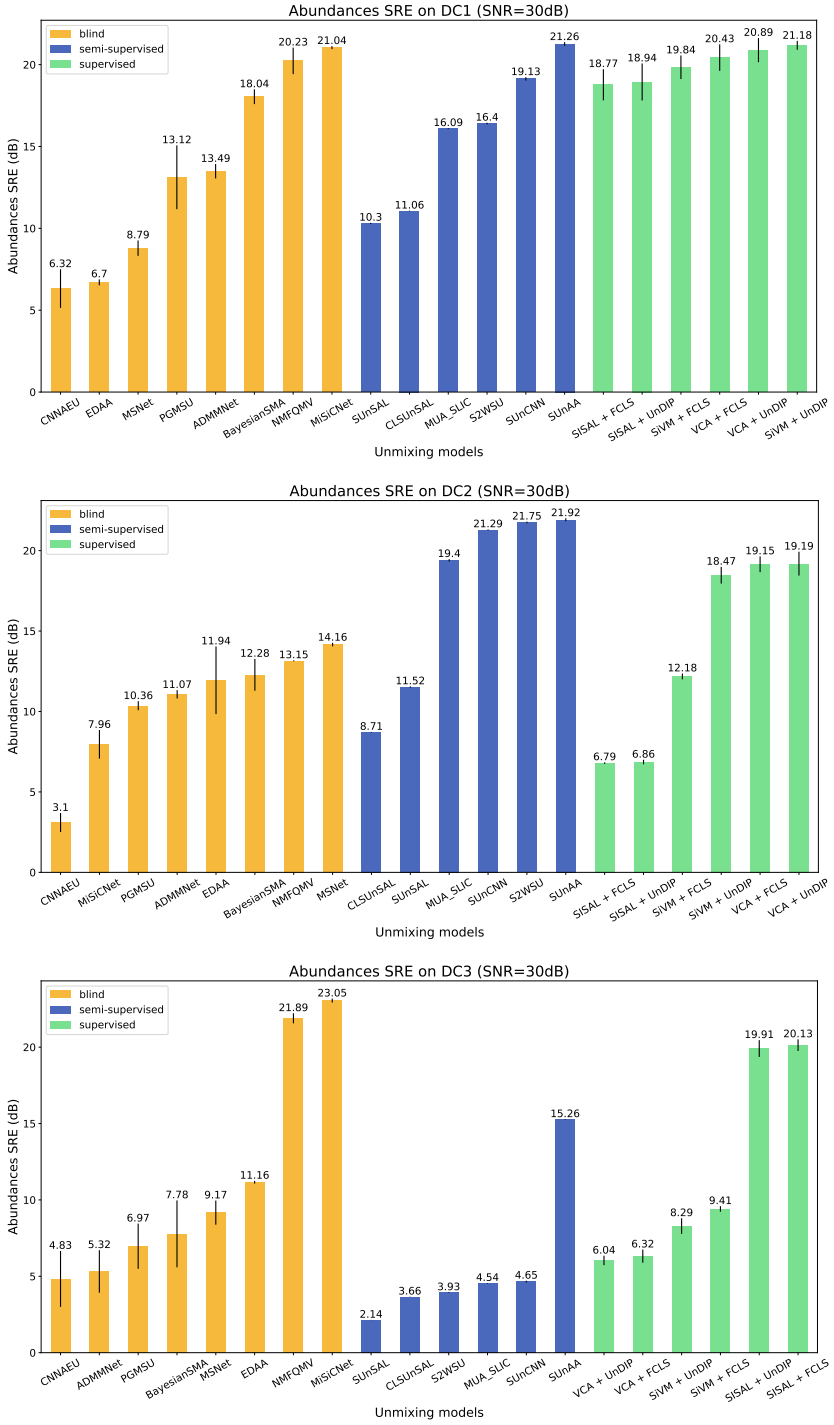


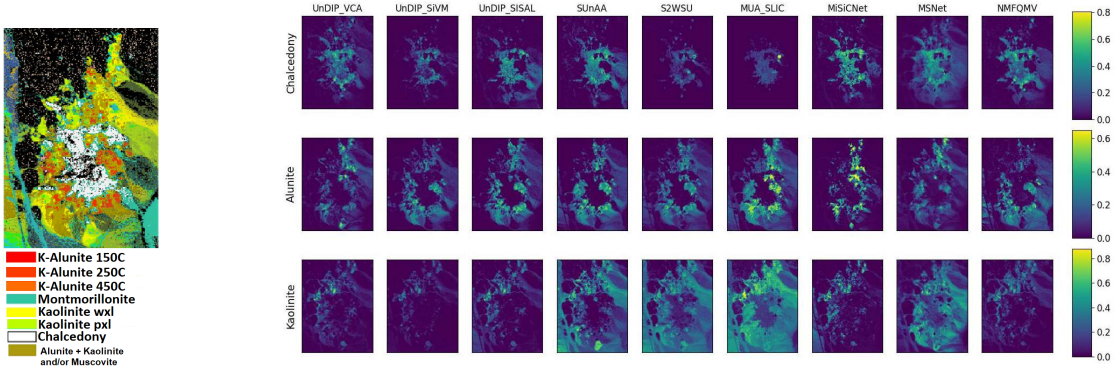
Figure 5.3: Comparing abundance SRE (↑) in dB using different unmixing techniques applied to (from top to bottom) synthetic DC1, DC2, and DC3.

5.3.3 Real data

We selected three methods per category to conduct unmixing on the Cuprite dataset. Blind unmixing: MiSiCNet, MSNet and NMFQMV. Semi-supervised: SUnAA, MUA_SLIC, S^{2}WSU. Supervised: UnDIP combined with SISAL, SiVM and VCA. We describe the hyperparameters that were fine-tuned for the following techniques. The hyperparameters set as default are omitted.

- MiSiCNet: $\lambda = 100$, `projection=True`.
- MSNet: $\alpha = 0.1$, $\beta = 0.1$.
- MUA_SLIC: $\beta = 30$, $\lambda_1 = 0.001$, $\lambda_2 = 0.001$, slic_size $= 200$.
- S^{2}WSU: $\lambda = 0.001$.
- SISAL: $\tau =$ 1e-6.

Fig. 5.4 (b) visually compares the estimated abundances for three dominant minerals, i.e., Chalcedony, Alunite, and Kaolinite. The comparison with the geological reference map Fig. 5.4 (a) reveals that the estimated abundances obtained by semisupervised methods show more resemblance to the reference map for all three minerals. SUnAA visually outperforms the other techniques, particularly in the case of Chalcedony. The blind unmixing methods can better estimate Chalcedony compared to MUA_SLIC and S^{2}WSU. This could be attributed to the mismatch of the endmember with the library's endmembers for this mineral. It is worth mentioning that SUnAA does not entirely rely on the library, and it estimates the endmember. Therefore, it can compensate for such a mismatch. More importantly, SUnAA is a parameter-free technique. We should note that selecting optimum parameters for the unmixing techniques is not a trivial task in real-world applications since the abundance RMSE cannot be computed.



(a) Geological Ref. Map (b) Estimate abundances

Figure 5.4: Estimate abundances obtained by applying different unmixing techniques to Cuprite compared with the geological reference map.

5.4 Discussion and Conclusion

Assuming that we have captured a spectral dataset and now have an unmixing problem in hand, we need to estimate the abundances of materials. The main question is which method

to choose and which group of methods to select to tackle the problem. Indeed, the first step is to evaluate our problem and see if the linear mixing model or its variations are suitable for our problem. This decision needs prior knowledge of the physics of the problem. For instance, if you are dealing with intimate mixtures or close-range and microscopic scenarios, you should use nonlinear models. If you are dealing with macroscopic Earth observation problems, then linear models or their variations will be suitable. In some research, nonlinear models perform better than linear ones, however, one may pay attention to the selection of the model in a real-world application. Usually, linear models are more general.

Here, we clarify the keys to the success of each category. The success of supervised (or sequential) unmixing lies in the confidence of the endmember measured, extracted, or estimated (pure/no pure scenario). Therefore, we should not use the supervised method if we are not confident about the endmembers. In other words, supervised methods perhaps are the best choice for endmembers with high confidence. When we have prior information on the material in the scene and a well-designed endmember library, semi-supervised unmixing could be successful. Semi-supervised are also suitable to capture the spectral variability. The success of semi-supervised unmixing lies in the quality of the endmember library. Blind unmixing methods should be selected when there is no library, no pure pixels in the data set (including highly mixed scenarios), or the confidence of the measure, selected, or estimated endmembers is low. They should be used with caution, and the estimated endmembers should always go through physical interpretation.

6

CONCLUSION AND PERSPECTIVES

In this thesis, we have developed novel algorithms based on sparse and archetypal signal decomposition aimed at addressing essential challenges in hyperspectral data analysis, specifically focusing on image restoration and spectral unmixing. First, we introduced a novel sparse coding-based unfolding algorithm designed for the representation of hyperspectral images. Furthermore, we leveraged an established model-based framework known as archetypal analysis to tackle various unmixing scenarios, offering valuable insights and solutions for spectral unmixing tasks. To facilitate the practical application of these methods, we have also developed a Python-based toolbox tailored for hyperspectral unmixing. This toolbox empowers practitioners to readily evaluate and compare multiple unmixing techniques on their own datasets, enhancing accessibility and usability in the hyperspectral research community.

6.1 Contributions summary

Here we summarize the contributions presented in this thesis.

Unfolding sparse coding to restore hyperspectral images Our first contribution investigated data-efficient and interpretable models for hyperspectral image restoration, by encoding prior knowledge used in the dictionary learning literature into an end-to-end trainable network architecture. In Chapter 2, we introduced a novel two-layered model-based deep learning method called T3SC. The primary objective of T3SC is to provide effective denoising of hyperspectral images while maintaining interpretability. The efficiency of our approach proved to be key to handle the limited availability of training data for hyperspectral denoising. In addition, our principled architecture, namely the sensor specific first spectral layer followed by a sensor agnostic spectral-spatial layer, is tailored to the specificity of hyperspectral images that are typically acquired by various sensors with different characteristics.

Modeling spectral unmixing using archetypal analysis Our second and third contributions draw significant inspiration from the interpretability offered by the archetypal analysis framework. In Chapter 3 we demonstrated the viability of archetypal analysis for blind hyperspectral unmixing within the linear mixing model. To efficiently tackle this challenge, we introduced a novel optimization scheme known as entropic gradient descent, specifically tailored for solving the archetypal analysis formulation. Notably, our approach is highly compatible with GPU computation, and its computational efficiency enables

us to develop a robust model selection procedure that mitigates the challenges posed by hyperparameter choices. In Chapter 4 we extended our approach to semi-supervised unmixing, emphasizing its advantages in addressing discrepancies between endmembers extracted from a library and those encountered in the actual scene. To facilitate the optimization process, we employed an active-set algorithm that leverages the inherent sparsity of solutions within the quadratic program, while enforcing simplicial constraints.

Enabling simple benchmarks for hyperspectral unmixing Finally, in Chapter 5, we draw a critical comparison of different unmixing techniques, classified into supervised, semi-supervised and blind unmixing categories. To enhance accessibility and usability, we introduced an open-source Python toolbox that offers users a wide array of unmixing techniques suitable for their specific datasets. This toolbox facilitates benchmarking across a range of popular approaches using both simulated and real datasets, enabling a comprehensive assessment of their respective merits and limitations.

6.2 Future research and perspectives

Based on the contributions of this thesis, several questions and research directions arise and would be interesting to investigate in the future.

Remaining challenges in spectral unmixing Despite significant advancements, spectral unmixing remains one of the most challenging tasks in hyperspectral analysis. Below, we briefly outline some of the primary challenges encountered in spectral unmixing:

- Linear models, while generally versatile, may experience a significant drop in performance when applied to different datasets.
- The selection of appropriate parameters significantly impacts the effectiveness of unmixing methods, and optimizing them for real-world datasets is particularly challenging.
- Linear unmixing methods often exhibit degraded performance as the number of endmembers increases, potentially failing on datasets with numerous endmembers.
- Spectral variability poses a major challenge, as it can significantly reduce the performance of linear unmixing techniques.
- The absence of real datasets with ground truth is a significant impediment.
- Multi-temporal and multi-source spectral unmixing tasks present additional complexities.
- Given the large volume of HS data, scalable unmixing approaches are crucial for global monitoring.

These challenges highlight the ongoing efforts and research needed to enhance the robustness and adaptability of spectral unmixing methods in various applications.

BIBLIOGRAPHY

[Adão et al., 2017] Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., and Sousa, J. J. (2017). Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote sensing*, 9(11):1110.

[Akbari et al., 2012] Akbari, H., Halig, L., Schuster, D. M., Fei, B., Osunkoya, A., Master, V., Nieh, P., and Chen, G. (2012). Hyperspectral Imaging and Quantitative Analysis for Prostate Cancer Detection. *Journal of Biomedical Optics*, 17(7):1–11.

[Alparone et al., 2007] Alparone, L., Wald, L., Chanussot, J., Thomas, C., Gamba, P., and Bruce, L. M. (2007). Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data-fusion contest. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3012–3021.

[Ambikapathi et al., 2011] Ambikapathi, A., Chan, T.-H., Ma, W.-K., and Chi, C.-Y. (2011). Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4194–4209.

[Arad and Ben-Shahar, 2016] Arad, B. and Ben-Shahar, O. (2016). Sparse Recovery of Hyperspectral Signal from Natural RGB Images. In *Proc. European Conference on Computer Vision (ECCV)*.

[Arbel, 2023] Arbel, M. (2023). MLXP: A framework for conducting machine learning experiments in python.

[Audebert et al., 2019] Audebert, N., Saux, B., and Lefèvre, S. (2019). Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2):159–173.

[Bach et al., 2012] Bach, F., Jenatton, R., Mairal, J., Obozinski, G., and others (2012). Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106.

[Baydin et al., 2018] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research (JMLR)*, 18:1–43.

[Beck and Teboulle, 2003] Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175.

[Bertsekas, 1997] Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334.

[Bioucas-Dias and Nascimento, 2008] Bioucas-Dias, J. and Nascimento, J. (2008). Hyperspectral Subspace Identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435 –2445.

[Bioucas-Dias, 2009] Bioucas-Dias, J. M. (2009). A variable splitting augmented Lagrangian approach to linear spectral unmixing. In *2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4.

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

[Bioucas-Dias et al., 2013] Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., and Chanussot, J. (2013). Hyperspectral Remote Sensing Data Analysis and Future Challenges. *IEEE Geoscience and Remote Sensing Magazine*, 1(2):6–36.

[Bioucas-Dias et al., 2012] Bioucas-Dias, J. M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., and Chanussot, J. (2012). Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379.

[Boardman et al., 1995] Boardman, J., Kruse, F. A., and Green, R. (1995). Mapping target signatures via partial unmixing of AVIRIS data: in Summaries. In *JPL Airborne Earth Sci. Workshop*, pages 23–26.

[Borsoi et al., 2019] Borsoi, R. A., Imbiriba, T., Bermudez, J. C. M., and Richard, C. (2019). A Fast Multiscale Spatial Regularization for Sparse Hyperspectral Unmixing. *IEEE Geoscience and Remote Sensing Letters*, 16(4):598–602.

[Borsoi et al., 2021] Borsoi, R. A., Imbiriba, T., Bermudez, J. C. M., Richard, C., Chanussot, J., Drumetz, L., Tourneret, J.-Y., Zare, A., and Jutten, C. (2021). Spectral Variability in Hyperspectral Data Unmixing: A comprehensive review. *IEEE Geoscience and Remote Sensing Magazine*, 9(4):223–270.

[Bregman, 1967] Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217.

[Buades et al., 2005] Buades, A., Coll, B., and Morel, J.-M. (2005). A Non-Local Algorithm for Image Denoising. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Chakrabarti and Zickler, 2011] Chakrabarti, A. and Zickler, T. (2011). Statistics of Real-World Hyperspectral Images. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Chang et al., 2017] Chang, Y., Yan, L., and Zhong, S. (2017). Hyper-Laplacian Regularized Unidirectional Low-Rank Tensor Recovery for Multispectral Image Denoising. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5901–5909.

[Chen et al., 2023] Chen, J., Zhao, M., Wang, X., Richard, C., and Rahardja, S. (2023). Integration of physics-based and data-driven models for hyperspectral image unmixing: A summary of current methods. *IEEE Signal Processing Magazine*, 40(2):61–74.

[Chen et al., 2014] Chen, Y., Mairal, J., and Harchaoui, Z. (2014). Fast and robust archetypal analysis for representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Clark et al., 2003] Clark, R. N., Swayze, G. A., Livo, K. E., Kokaly, R. F., Sutley, S. J., Dalton, J. B., McDougal, R. R., and Gent, C. A. (2003). Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems. *Journal of Geophysical Research: Planets*, 108(E12).

[Comon and Jutten, 2010] Comon, P. and Jutten, C. (2010). *Handbook of Blind Source Separation: Independent component analysis and applications*.

[Cutler and Breiman, 1994] Cutler, A. and Breiman, L. (1994). Archetypal analysis. *Technometrics*, 36(4):338–347.

[Dabov et al., 2007] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007). Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095.

[Daubechies et al., 2004] Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457.

[De Handschutter et al., 2019] De Handschutter, P., Gillis, N., Vandaele, A., and Siebert, X. (2019). Near-convex archetypal analysis. *IEEE Signal Processing Letters*, 27:81–85.

[Dobigeon et al., 2009] Dobigeon, N., Moussaoui, S., Coulon, M., Tourneret, J., and Hero, A. O. (2009). Joint Bayesian Endmember Extraction and Linear Unmixing for Hyperspectral Imagery. *IEEE Transactions on Signal Processing*, 57(11):4355–4368.

[Du et al., 2007] Du, Q., Younan, N. H., King, R., and Shah, V. P. (2007). On the performance evaluation of pan-sharpening techniques. *IEEE Geoscience and Remote Sensing Letters*, 4(4):518–522.

[Eckstein and Bertsekas, 1992] Eckstein, J. and Bertsekas, D. P. (1992). On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318.

[Elad and Aharon, 2006] Elad, M. and Aharon, M. (2006). Image Denoising via Sparse and Redundant Representations over Learned Dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745.

[Elad and Bruckstein, 2002] Elad, M. and Bruckstein, A. M. (2002). A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567.

[Elmasry et al., 2012] Elmasry, G., Kamruzzaman, M., Sun, D.-W., and Paul Allen (2012). Principles and Applications of Hyperspectral Imaging in Quality Evaluation of Agro-Food Products: A Review. *Critical Reviews in Food Science and Nutrition*, 52(11):999–1023.

[Fan et al., 2018] Fan, H., Li, C., Guo, Y., Kuang, G., and Ma, J. (2018). Spatial–Spectral Total Variation Regularized Low-Rank Tensor Decomposition for Hyperspectral Image Denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 56(10):6196–6213.

[Fei, 2020] Fei, B. (2020). Chapter 3.6 - Hyperspectral Imaging in Medical Applications. In Amigo, J. M., editor, *Hyperspectral Imaging*, volume 32 of *Data Handling in Science and Technology*, pages 523–565. Elsevier.

[Feng and Da-Wen Sun, 2012] Feng, Y.-Z. and Da-Wen Sun (2012). Application of Hyperspectral Imaging in Food Safety Inspection and Control: A Review. *Critical Reviews in Food Science and Nutrition*, 52(11):1039–1058.

[Figueiredo and Nowak, 2003] Figueiredo, M. and Nowak, R. (2003). An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Transactions on Image Processing*, 12(8):906–916.

[Fox et al., 2017] Fox, N., Parbhakar-Fox, A., Moltzen, J., Feig, S., Goemann, K., and Huntington, J. (2017). Applications of hyperspectral mineralogy for geoenvironmental characterisation. *Minerals Engineering*, 107:63–77.

[Fu et al., 2015] Fu, Y., Lam, A., Sato, I., and Sato, Y. (2015). Adaptive Spatial-Spectral Dictionary Learning for Hyperspectral Image Denoising. In *Proc. International Conference on Computer Vision (ICCV)*.

[Ghamisi et al., 2017] Ghamisi, P., Yokoya, N., Li, J., Liao, W., Liu, S., Plaza, J., Rasti, B., and Plaza, A. (2017). Advances in Hyperspectral Image and Signal Processing: A Comprehensive Overview of the State of the Art. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):37–78.

[Goetz, 2009] Goetz, A. F. (2009). Three Decades of Hyperspectral Remote Sensing of the Earth: A Personal View. *Remote Sensing of Environment*, 113:S5–S16.

[Goetz et al., 1985] Goetz, A. F., Vane, G., Solomon, J. E., and Rock, B. N. (1985). Imaging spectrometry for earth remote sensing. *Science*, 228(4704):1147–1153.

[Gong et al., 2020] Gong, X., Chen, W., and Chen, J. (2020). A Low-Rank Tensor Dictionary Learning Method for Hyperspectral Image Denoising. *IEEE Transactions on Signal Processing*, 68:1168–1180.

[Gonzalez et al., 2018] Gonzalez, C. A. G., Absil, O., and van Droogenbroeck, M. (2018). Supervised Detection of Exoplanets in High-Contrast Imaging Sequences. *Astronomy & Astrophysics*, 613:A71.

[Gowen et al., 2015] Gowen, A. A., Feng, Y., Gaston, E., and Valdramidis, V. (2015). Recent Applications of Hyperspectral Imaging in Microbiology. *Talanta*, 137:43–54.

[Gowen et al., 2007] Gowen, A. A., O'Donnell, C. P., Cullen, P. J., Downey, G., and Frias, J. M. (2007). Hyperspectral imaging–an emerging process analytical tool for food quality and safety control. *Trends in food science & technology*, 18(12):590–598.

[Green et al., 1998] Green, R. O., Eastwood, M. L., Sarture, C. M., Chrien, T. G., Aronsson, M., Chippendale, B. J., Faust, J. A., Pavri, B. E., Chovit, C. J., Solis, M., and others (1998). Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, 65(3):227–248.

[Gregor and LeCun, 2010] Gregor, K. and LeCun, Y. (2010). Learning fast approximations of sparse coding. In *Proceedings of the 27th on international conference on machine learning (ICML)*, pages 399–406.

[Gribonval and Nielsen, 2003] Gribonval, R. and Nielsen, M. (2003). Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[He et al., 2020] He, W., Yao, Q., Li, C., Yokoya, N., Zhao, Q., Zhang, H., and Zhang, L. (2020). Non-local meets global: An iterative paradigm for hyperspectral image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2089–2107.

[Heinz and Chein-I-Chang, 2001] Heinz, D. C. and Chein-I-Chang (2001). Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(3):529–545.

[Heylen et al., 2011] Heylen, R., Burazerovic, D., and Scheunders, P. (2011). Fully Constrained Least Squares Spectral Unmixing by Simplex Projection. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4112–4122.

[Heylen et al., 2014] Heylen, R., Parente, M., and Gader, P. (2014). A review of nonlinear hyperspectral unmixing methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):1844–1868.

[Hong et al., 2021] Hong, D., He, W., Yokoya, N., Yao, J., Gao, L., Zhang, L., Chanussot, J., and Zhu, X. (2021). Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing. *IEEE Geoscience and Remote Sensing Magazine*, 9(2):52–87.

[Hong et al., 2018] Hong, D., Yokoya, N., Chanussot, J., and Zhu, X. X. (2018). An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Transactions on Image Processing*, 28(4):1923–1938.

[Huck et al., 2010] Huck, A., Guillaume, M., and Blanc-Talon, J. (2010). Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(6):2590–2602.

[Ince, 2020] Ince, T. (2020). Superpixel-Based Graph Laplacian Regularization for Sparse Hyperspectral Unmixing. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5.

[Iordache et al., 2012] Iordache, M., Bioucas-Dias, J. M., and Plaza, A. (2012). Total Variation Spatial Regularization for Sparse Hyperspectral Unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4484–4502.

[Iordache et al., 2014a] Iordache, M., Bioucas-Dias, J. M., and Plaza, A. (2014a). Collaborative Sparse Regression for Hyperspectral Unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):341–354.

[Iordache et al., 2014b] Iordache, M.-D., Bioucas-Dias, J. M., Plaza, A., and Somers, B. (2014b). MUSIC-CSR: Hyperspectral Unmixing via Multiple Signal Classification and Collaborative Sparse Regression. *IEEE Transactions on Geoscience and Remote Sensing*, 52(7):4364–4382.

[Itten et al., 2008] Itten, K. I., Dell'Endice, F., Hueni, A., Kneubühler, M., Schläpfer, D., Odermatt, D., Seidel, F., Huber, S., Schopfer, J., Kellenberger, T., and others (2008). APEX-the hyperspectral ESA airborne prism experiment. *Sensors*, 8(10):6235–6259.

[Jimenez and Plaza, 2015] Jimenez, L. I. and Plaza, A. (2015). HyperMix: An open source tool for hyperspectral imaging. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1749–1752.

[Karaca et al., 2013] Karaca, A. C., Ertürk, A., Güllü, M. K., Elmas, M., and Ertürk, S. (2013). Automatic waste sorting using shortwave infrared hyperspectral imaging system. In *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4.

[Kerekes and Baum, 2003] Kerekes, J. and Baum, J. (2003). Hyperspectral Imaging System Modeling. *Lincoln Laboratory Journal*, 14(1):117–130.

[Keshava and Mustard, 2002] Keshava, N. and Mustard, J. F. (2002). Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57.

[Kong et al., 2020] Kong, Z., Yang, X., and He, L. (2020). A comprehensive comparison of multi-dimensional image denoising methods. *arXiv preprint arXiv:2011.03462*.

[Landgrebe, 2002] Landgrebe, D. (2002). Hyperspectral image data analysis. *IEEE Signal Processing Magazine*, 19(1):17–28.

[Lecouat et al., 2020a] Lecouat, B., Ponce, J., and Mairal, J. (2020a). A flexible framework for designing trainable priors with adaptive smoothing and game encoding. *Advances in Neural Information Processing Systems*, 33:15664–15675.

[Lecouat et al., 2020b] Lecouat, B., Ponce, J., and Mairal, J. (2020b). Fully Trainable and Interpretable Non-Local Sparse Models for Image Restoration. In *Proc. European Conference on Computer Vision (ECCV)*.

[LeCun et al., 2002] LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. (2002). Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer.

[Lee and Seung, 2000] Lee, D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *Adv. in Neural Information Processing Systems (NIPS)*, 13.

[Lin and Gader, 2022] Lin, Y. and Gader, P. (2022). Addressing Spectral Variability In Hyperspectral Unmixing With Unsupervised Neural Networks. In *2022 12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5.

[Liu et al., 2017] Liu, Y., Pu, H., and Sun, D.-W. (2017). Hyperspectral Imaging Technique for Evaluating Food Quality and Safety during Various Processes: A Review of Recent Applications. *Trends in Food Science & Technology*, 69:25–35.

[Lu et al., 2020] Lu, B., Dao, P. D., Liu, J., He, Y., and Shang, J. (2020). Recent Advances of Hyperspectral Imaging Technology and Applications in Agriculture. *Remote Sensing*, 12(16).

[Lu and Fei, 2014] Lu, G. and Fei, B. (2014). Medical Hyperspectral Imaging: A Review. *Journal of Biomedical Optics*, 19(1):1–24.

[Maffei et al., 2020] Maffei, A., Haut, J. M., Paoletti, M. E., Plaza, J., Bruzzone, L., and Plaza, A. (2020). A Single Model CNN for Hyperspectral Image Denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2516–2529.

[Maggioni et al., 2013] Maggioni, M., Katkovnik, V., Egiazarian, K., and Foi, A. (2013). Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133.

[Mahesh et al., 2015] Mahesh, S., Jayas, D., Paliwal, J., and White, N. (2015). Hyperspectral Imaging to Classify and Monitor Quality of Agricultural Materials. *Journal of Stored Products Research*, 61:17–26.

[Mairal et al., 2012] Mairal, J., Bach, F., and Ponce, J. (2012). Task-Driven Dictionary Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(4):791–804.

[Mairal et al., 2014] Mairal, J., Bach, F., Ponce, J., and others (2014). Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283.

[Mairal et al., 2007] Mairal, J., Elad, M., and Sapiro, G. (2007). Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69.

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

[Miao and Qi, 2007] Miao, L. and Qi, H. (2007). Endmember Extraction From Highly Mixed Data Using Minimum Volume Constrained Nonnegative Matrix Factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777.

[Monga et al., 2021] Monga, V., Li, Y., and Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44.

[Mäkinen et al., 2019] Mäkinen, Y., Azzari, L., and Foi, A. (2019). Exact Transform-Domain Noise Variance for Collaborative Filtering of Stationary Correlated Noise. In *IEEE International Conference on Image Processing (ICIP)*.

[Mäkinen et al., 2020] Mäkinen, Y., Azzari, L., and Foi, A. (2020). Collaborative Filtering of Correlated Noise: Exact Transform-Domain Variance for Improved Shrinkage and Patch Matching. *IEEE Transactions on Image Processing*, 29:8339–8354.

[Mørup and Hansen, 2012] Mørup, M. and Hansen, L. K. (2012). Archetypal analysis for machine learning and data mining. *Neurocomputing*, 80:54–63.

[Nascimento and Bioucas-Dias, 2003] Nascimento, J. and Bioucas-Dias, J. (2003). Vertex Component Analysis: A\textasciitildeFast Algorithm to Extract Endmembers Spectra from Hyperspectral Data. In Perales, F. J., Campilho, A. J. C., de la Blanca, N. P., and Sanfeliu, A., editors, *Pattern Recognition and Image Analysis*, pages 626–635, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Nascimento and Dias, 2005] Nascimento, J. M. and Dias, J. M. (2005). Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910.

[Nesterov, 2003] Nesterov, Y. (2003). *Introductory lectures on convex optimization: A basic course*, volume 87.

[Nocedal and Wright, 1999] Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*.

[Olshausen and Field, 1996] Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

[Othman and Qian, 2006] Othman, H. and Qian, S.-E. (2006). Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage. *IEEE Transactions on Geoscience and Remote Sensing*, 44(2):397 – 408.

[Ozkan et al., 2018] Ozkan, S., Kaya, B., and Akar, G. B. (2018). Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1):482–496.

[Paatero and Tapper, 1994] Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

[Palsson et al., 2022] Palsson, B., Sveinsson, J. R., and Ulfarsson, M. O. (2022). Blind hyperspectral unmixing using autoencoders: A critical comparison. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:1340–1372.

[Palsson et al., 2020] Palsson, B., Ulfarsson, M. O., and Sveinsson, J. R. (2020). Convolutional autoencoder for spectral–spatial hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1):535–549.

[Parra et al., 1999] Parra, L., Spence, C., Sajda, P., Ziehe, A., and Müller, K.-R. (1999). Unmixing hyperspectral data. *Adv. in Neural Information Processing Systems (NIPS)*, 12.

[Peng et al., 2020] Peng, J., Sun, W., Jiang, F., Chen, H., Zhou, Y., and Du, Q. (2020). A general loss-based nonnegative matrix factorization for hyperspectral unmixing. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

[Peng et al., 2021] Peng, J., Zhou, Y., Sun, W., Du, Q., and Xia, L. (2021). Self-Paced Nonnegative Matrix Factorization for Hyperspectral Unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 59(2):1501–1515.

[Peng et al., 2014] Peng, Y., Meng, D., Xu, Z., Gao, C., Yang, Y., and Zhang, B. (2014). Decomposable Nonlocal Tensor Dictionary Learning for Multispectral Image Denoising. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Pesme et al., 2021] Pesme, S., Pillaud-Vivien, L., and Flammarion, N. (2021). Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Adv. in Neural Information Processing Systems (NeurIPS)*.

[Plaza et al., 2009] Plaza, A., Benediktsson, J. A., Boardman, J. W., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., and others (2009). Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, 113:110–122.

[Plaza et al., 2011] Plaza, A., Martín, G., Plaza, J., Zortea, M., and Sánchez, S. (2011). Recent developments in endmember extraction and spectral unmixing. *Optical Remote Sensing: Advances in Signal Processing and Exploitation Techniques*, pages 235–267.

[Qi et al., 2016] Qi, N., Shi, Y., Sun, X., and Yin, B. (2016). TenSR: Multi-Dimensional Tensor Sparse Representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Rasti and Koirala, 2022] Rasti, B. and Koirala, B. (2022). SUnCNN: Sparse Unmixing Using Unsupervised Convolutional Neural Network. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

[Rasti et al., 2022] Rasti, B., Koirala, B., Scheunders, P., and Chanussot, J. (2022). Misicnet: Minimum simplex convolutional network for deep hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15.

[Rasti et al., 2021] Rasti, B., Koirala, B., Scheunders, P., and Ghamisi, P. (2021). UnDIP: Hyperspectral unmixing using deep image prior. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15.

[Rasti et al., 2018] Rasti, B., Scheunders, P., Ghamisi, P., Licciardi, G., and Chanussot, J. (2018). Noise Reduction in Hyperspectral Imagery: Overview and Application. *Remote Sensing*, 10(3):482.

[Rasti et al., 2014] Rasti, B., Sveinsson, J. R., Ulfarsson, M. O., and Benediktsson, J. A. (2014). Hyperspectral Image Denoising Using First Order Spectral Roughness Penalty in Wavelet Domain. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2458–2467.

[Rasti et al., 2017] Rasti, B., Ulfarsson, M. O., and Ghamisi, P. (2017). Automatic Hyperspectral Image Restoration Using Sparse and Low-Rank Modeling. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2335–2339.

[Rasti et al., 2023a] Rasti, B., Zouaoui, A., Mairal, J., and Chanussot, J. (2023a). Image Processing and Machine Learning for Hyperspectral Unmixing: An Overview and the HySUPP Python Package. *arXiv preprint arXiv:2308.09375*.

[Rasti et al., 2023b] Rasti, B., Zouaoui, A., Mairal, J., and Chanussot, J. (2023b). SUnAA: Sparse Unmixing using Archetypal Analysis. *IEEE Geoscience and Remote Sensing Letters*.

[Ren et al., 2023] Ren, L., Hong, D., Gao, L., Sun, X., Huang, M., and Chanussot, J. (2023). Orthogonal Subspace Unmixing to Address Spectral Variability for Hyperspectral Image. *IEEE Transactions on Geoscience and Remote Sensing*.

[Rickard et al., 1993] Rickard, L. J., Basedow, R. W., Zalewski, E. F., Silverglate, P. R., and Landers, M. (1993). HYDICE: An airborne system for hyperspectral imaging. In *Imaging Spectrometry of the Terrestrial Environment*, volume 1937, pages 173–179.

[Roberts et al., 1998] Roberts, D. A., Gardner, M., Church, R., Ustin, S., Scheer, G., and Green, R. (1998). Mapping chaparral in the Santa Monica Mountains using multiple endmember spectral mixture models. *Remote sensing of environment*, 65(3):267–279.

[Schaepman et al., 2015] Schaepman, M. E., Jehle, M., Hueni, A., D'Odorico, P., Damm, A., Weyermann, J., Schneider, F. D., Laurent, V., Popp, C., Seidel, F. C., and others (2015). Advanced radiometry measurements and Earth science applications with the Airborne Prism Experiment (APEX). *Remote Sensing of Environment*, 158:207–219.

[Schaepman et al., 2009] Schaepman, M. E., Ustin, S. L., Plaza, A. J., Painter, T. H., Verrelst, J., and Liang, S. (2009). Earth system science related imaging spectroscopy—An assessment. *Remote Sensing of Environment*, 113:123–137.

[Shi et al., 2021a] Shi, Q., Tang, X., Yang, T., Liu, R., and Zhang, L. (2021a). Hyperspectral Image Denoising Using a 3-D Attention Denoising Network. *IEEE Transactions on Geoscience and Remote Sensing*.

[Shi et al., 2021b] Shi, S., Zhao, M., Zhang, L., Altmann, Y., and Chen, J. (2021b). Probabilistic generative model for hyperspectral unmixing accounting for endmember variability. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15.

[Shlezinger et al., 2023] Shlezinger, N., Whang, J., Eldar, Y. C., and Dimakis, A. G. (2023). Model-based deep learning. *Proceedings of the IEEE*.

[Simon and Elad, 2019] Simon, D. and Elad, M. (2019). Rethinking the CSC Model for Natural Images. In *Adv. Neural Information Processing Systems (NeurIPS)*.

[Somers et al., 2011] Somers, B., Asner, G. P., Tits, L., and Coppin, P. (2011). Endmember variability in spectral mixture analysis: A review. *Remote Sensing of Environment*, 115(7):1603–1616.

[Studer et al., 2012] Studer, V., Bobin, J., Chahid, M., Mousavi, H. S., Candes, E., and Dahan, M. (2012). Compressive Fluorescence Microscopy for Biological and Hyperspectral Imaging. *Proceedings of the National Academy of Sciences*, 109(26):E1679–E1687.

[Sun et al., 2017] Sun, W., Yang, G., Wu, K., Li, W., and Zhang, D. (2017). Pure endmember extraction using robust kernel archetypoid analysis for hyperspectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 131:147–159.

[Teboulle, 1992] Teboulle, M. (1992). Entropic proximal mappings with applications to nonlinear programming. *Mathematics of Operations Research*, 17(3):670–690.

[Tucker, 1966] Tucker, L. (1966). Some Mathematical Notes on Three-Mode Factor Analysis. *Psychometrika*, 31(3):279–311.

[Wang et al., 2021] Wang, M., Wang, Q., Chanussot, J., and Li, D. (2021). Hyperspectral Image Mixed Noise Removal Based on Multidirectional Low-Rank Modeling and Spatial-Spectral Total Variation. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1):488–507.

[Wang et al., 2017] Wang, R., Li, H.-C., Pizurica, A., Li, J., Plaza, A., and Emery, W. J. (2017). Hyperspectral Unmixing Using Double Reweighted Sparse Regression and Total Variation. *IEEE Geoscience and Remote Sensing Letters*, 14(7):1146–1150.

[Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 –612.

[Wei et al., 2020] Wei, K., Fu, Y., and Huang, H. (2020). 3-D Quasi-Recurrent Neural Network for Hyperspectral Image Denoising. *IEEE Transactions on Neural Networks and Learning Systems*.

[Winter, 1999] Winter, M. E. (1999). N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In Descour, M. R. and Shen, S. S., editors, *Imaging Spectrometry V*, volume 3753, pages 266 – 275. SPIE. Backup Publisher: International Society for Optics and Photonics.

[Xiong et al., 2020] Xiong, F., Zhou, J., Lu, J., and Qian, Y. (2020). SMDS-Net: Model Guided Spectral-Spatial Network for Hyperspectral Image Denoising. *arXiv:2012.01829*.

[Xu et al., 2022] Xu, M., Yang, Z., Ren, G., Sheng, H., Liu, S., Liu, W., and Ye, C. (2022). L1 Sparsity-Constrained Archetypal Analysis Algorithm for Hyperspectral Unmixing. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

[Yang et al., 2010] Yang, Z., Zhou, G., Xie, S., Ding, S., Yang, J.-M., and Zhang, J. (2010). Blind spectral unmixing based on sparse nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 20(4):1112–1125.

[Yao et al., 2021] Yao, J., Hong, D., Xu, L., Meng, D., Chanussot, J., and Xu, Z. (2021). Sparsity-enhanced convolutional decomposition: A novel tensor-based paradigm for blind hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14.

[Yao et al., 2019] Yao, J., Meng, D., Zhao, Q., Cao, W., and Xu, Z. (2019). Nonconvex-Sparsity and Nonlocal-Smoothness-Based Blind Hyperspectral Unmixing. *IEEE Transactions on Image Processing*, 28(6):2991–3006.

[Yasuma et al., 2010] Yasuma, F., Mitsunaga, T., Iso, D., and Nayar, S. K. (2010). Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253.

[Yu et al., 2022] Yu, Y., Ma, Y., Mei, X., Fan, F., Huang, J., and Li, H. (2022). Multi-stage convolutional autoencoder network for hyperspectral unmixing. *International Journal of Applied Earth Observation and Geoinformation*, 113:102981.

[Yuan et al., 2014] Yuan, Q., Zhang, L., and Shen, H. (2014). Hyperspectral Image Denoising with a Spatial–Spectral View Fusion Strategy. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2314–2325.

[Yuan et al., 2019] Yuan, Q., Zhang, Q., Li, J., Shen, H., and Zhang, L. (2019). Hyperspectral Image Denoising Employing a Spatial–Spectral Deep Residual Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1205–1218.

[Zare and Ho, 2013] Zare, A. and Ho, K. (2013). Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing. *IEEE Signal Processing Magazine*, 31(1):95–104.

[Zhang et al., 2014] Zhang, H., He, W., Zhang, L., Shen, H., and Yuan, Q. (2014). Hyperspectral Image Restoration Using Low-Rank Matrix Recovery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4729–4743.

[Zhang et al., 2017] Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

[Zhang et al., 2011] Zhang, L., Zhang, L., Mou, X., and Zhang, D. (2011). FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386.

[Zhang et al., 2018a] Zhang, S., Li, J., Li, H., Deng, C., and Plaza, A. (2018a). Spectral–Spatial Weighted Sparse Regression for Hyperspectral Image Unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3265–3276.

[Zhang et al., 2018b] Zhang, X., Sun, Y., Zhang, J., Wu, P., and Jiao, L. (2018b). Hyperspectral Unmixing via Deep Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 15(11):1755–1759.

[Zhao and Yang, 2015] Zhao, Y.-Q. and Yang, J. (2015). Hyperspectral Image Denoising via Sparse Representation and Low-Rank Constraint. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):296–308.

[Zhou and Rodrigues, 2021] Zhou, C. and Rodrigues, M. R. (2021). ADMM-based hyperspectral unmixing networks for abundance and endmember estimation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18.

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

[Zhuang and Bioucas-Dias, 2017] Zhuang, L. and Bioucas-Dias, J. M. (2017). Hyperspectral Image Denoising Based on Global and Non-Local Low-Rank Factorizations. In *IEEE International Conference on Image Processing (ICIP)*.

[Zhuang et al., 2019] Zhuang, L., Lin, C.-H., Figueiredo, M. A., and Bioucas-Dias, J. M. (2019). Regularization parameter selection in minimum volume hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9858–9877.

[Zouaoui et al., 2023] Zouaoui, A., Muhawenayo, G., Rasti, B., Chanussot, J., and Mairal, J. (2023). Entropic descent archetypal analysis for blind hyperspectral unmixing. *IEEE Transactions on Image Processing*.

[Zymnis et al., 2007] Zymnis, A., Kim, S.-J., Skaf, J., Parente, M., and Boyd, S. (2007). Hyperspectral image unmixing via alternating projected subgradients. In *Asilomar Conference on Signals, Systems and Computers*, pages 1164–1168.