POLITECNICO DI BARI

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE (D.M.270/04)

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA:
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE (LM 17-24)

TECHNICAL REPORT ON COMPUTER VISION:
IMAGE PROCESSING

SPECTRAL UNMIXING

Professor:
- GUERRIERO Andrea

Students:
- PANSINI Germano (589281)
- LONGO Pierluigi (590926)

A.A. 2022 / 2023

TABLE OF CONTENTS:

1 Abstract

This technical report presents a study and a MATLAB code project development based on a pivotal technique by the name of Spectral Unmixing that is used in remote sensing aiming to decipher constituent materials from mixed spectral signatures obtained from hyperspectral imagery. Our implementation is tested on real case scenario exploiting a publicly benchmark dataset "DLR HySU", available at this link https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-12760/22294_read-73262/, that consists of airborne imagery acquired by a HySpex spectrometer, a 3K RGB camera and is also complemented with in-situ ground-truth data acquired with a SVC field spectrometer. This dataset is particularly designed to test the main steps of Spectral Unmixing and provide all the spectral libraries to validate the obtained results. Our MATLAB-based implementation integrates advanced algorithm based on a combination of geometric and statistical methods, offering a robust and efficient approach to tackle challenges concerning this topic.

We initially introduce the state-of-the-art algorithms and optimization techniques and then, in the 7th chapter "**Project Overview – Guide To the Code**" is presented a detailed breakdown of our MATLAB-based custom implementation of Spectral Unmixing. Results are presented in both quantitative metrics and visual representations. The modular design of our code ensures adaptability and scalability, making it valuable end reproducible with others dataset in the domain of remote sensing and spectral analysis. Furthermore, conclusions, future challenges and possible applications and developments are mentioned in the last chapter.

2 SPECTRAL UNMIXING

Spectral unmixing is a technique used in remote sensing, particularly in the field of hyperspectral imagery analysis, to extract information about the composition of a scene from its spectral measurements, decomposing a pixel's spectral signature into its constituent pure spectral signatures. In real-world scenes, the measured spectra in hyperspectral data often result from a mixture of multiple materials or substances, also due to the low spatial resolution of cameras acquiring aerials imagery. The process of spectral unmixing aims to separate these mixed spectra into their constituent pure spectral signatures, also known as endmembers, and then determine the proportion of each endmember present in the mixed pixel. This process is analogous to decomposing a colour into its constituent primary colors in the RGB model. There are a few common methods used for spectral unmixing but we're going to focus our attention on LSU, which we then used in our MATLAB-based implementation, and NSU.

1. **Linear Spectral Unmixing (LSU)**
   This method assumes that the mixed spectrum in a pixel is a linear combination of the endmember spectra of the individual materials within it, where the proportions (abundances) of each endmember determine the observed spectrum. The task is to find the endmember spectra and their abundances that best approximate the observed spectrum.

2. **Non-linear Spectral Unmixing (NSU)**
   In some cases, the relationship between the observed spectrum and endmember

spectra may not be strictly linear. Non-linear spectral unmixing techniques consider more complex relationships and models between endmembers and observed spectra. So, the spectral signature of a mixed pixel may not be a simple linear combination of endmembers spectra.

3. **Sparse Unmixing**
   This method assumes that only a few endmembers contribute significantly to a mixed pixel's spectrum. It aims to find a sparse set of endmembers and their corresponding abundances to explain the observed spectrum. It is particularly useful when working with highly mixed pixels where LSU may not perform well due to the presence of unknown materials and noise.

4. **Joint Spatial-Spectral Unmixing (JSSU)**
   This approach considers both the spectral and spatial information, improving results. It considers the fact that neighbouring pixels are likely to have similar endmember compositions, spectral characteristics, and spatial properties.

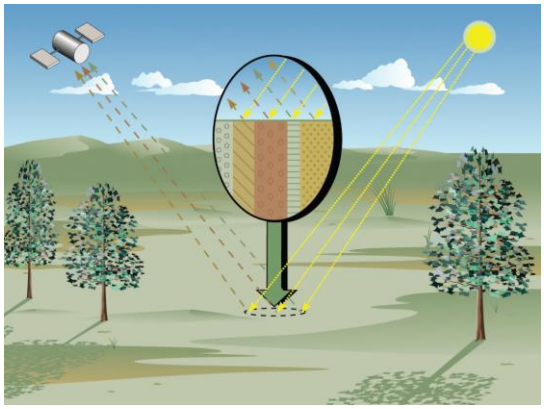
Spectral unmixing has remote sensing applications that can be applied in various fields like environmental monitoring, geology, agriculture, mineral exploration, and more. It allows researchers to identify and quantify the presence of different materials or substances in a scene, providing valuable information for a range of applications such as land cover classification, pollution assessment, and resource management.

2.1 LINEAR MIXING

The dynamics behind the mixing of two or more substances depends largely upon the kind of mixture within a pixel that scatters the incident solar radiation. The image illustrates the reflecting surface as a checkerboard mixture of endmembers, and the incident radiation bounces only once upon its surface. By this model, if the total surface area is divided proportionally according to the fractional abundances of the constituent substances, then the reflected radiation conveys with the same proportions the characteristics of the associated materials. In this sense, a linear relationship exists between the fractional abundance of the substances comprising the area being imaged and the spectrum of the reflected radiation. If we have K spectral bands, and we denote the ith endmember spectrum as s_i and the abundance of the ith endmember as a_i , the observed spectrum x for any pixel in the scene can be expressed as:

$$\boldsymbol{x = a_1 s_1 + a_2 s_2 + \cdots + a_M s_M + w = \sum_{i=1}^{M} a_i s_i + w = S_a + w}\ ,$$

where $\boldsymbol{M}$ is the number of endmembers, $\boldsymbol{S}$ is the matrix of endmembers, and $\boldsymbol{w}$ is an error term accounting for additive noise. (Keshava N. , 2003)



Linear Spectral Unmixing is based on a set of four key assumptions:

1) **Absence of Multiple Scattering**: no significant occurrence of multiple scattering between different surface components so the interaction between different materials in a pixel is primarily linear.

2) **Spectral Contrast**: Each surface component within the image should exhibit sufficient spectral contrast to enable their reliable separation implying that the spectral signatures should be distinguishable from one another.

3) **Unity Total Land Cover**: the total land cover in each pixel is equal to unity meaning that the combined contributions of all surface components should sum up to one, reflecting a complete coverage of the pixel.

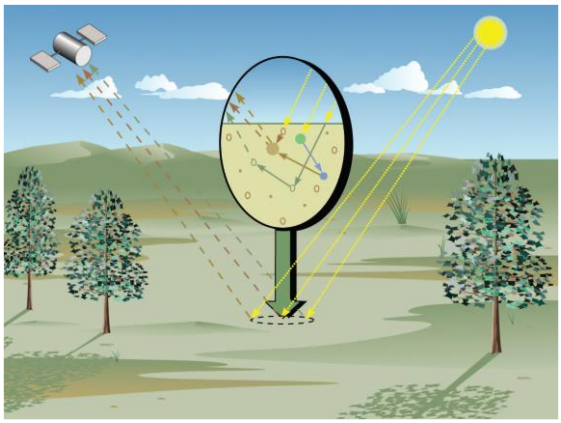
4) **Knowledge of Endmembers**: the spectral signatures of each surface component are known or can be accurately estimated. This is crucial for accurately unmixing the pixel spectra and determining the abundance fractions of each endmember.

These assumptions provide the foundation for the application of Linear Spectral Unmixing in remote sensing and hyperspectral image analysis. They guide the process of decomposing mixed pixel spectra into their constituent endmembers and estimating their abundances, allowing for valuable insights into the composition of the observed scene.

2.2 NON-LINEAR MIXING

Nonlinear Spectral Unmixing (NSU) extends traditional Linear Spectral Unmixing (LSU) by relaxing some of the key assumptions made before. While NSU offers greater flexibility and the ability to model complex spectral mixing phenomena, it introduces different assumptions and considerations.

The image depicts a more complicated scenario. The arrangement of the constituent material substances is not as orderly as in the figure because the substances comprising the medium are not organized proportionally on the surface. This intimate mixture of materials results when each component is randomly distributed in a homogeneous way. As a result, the incident radiation can experience reflections with multiple substances, and the aggregate spectrum of reflected radiation may no longer uphold the linear proportions (either in mass fraction or in volume) of the constituent substance spectrum. (Keshava N. , 2003)



Here are some of the primary assumptions and characteristics of Nonlinear Spectral Unmixing:

1) **Nonlinearity**: the relationship between observed mixed pixel spectra and the underlying endmember spectra is nonlinear.

2) **Endmember Variability**: due to factors like illumination conditions, sensor artifacts, and atmospheric effects there's a possible variation in endmembers' spectra. This method aims modelling spectral diversity more effectively.

3) **Endmember Uncertainty**: considers uncertainty in endmember spectra. In practical scenarios, endmember spectra may not be precisely known but may exhibit some degree of uncertainty or variability.

4) **Model Flexibility**: provides a more flexible modelling framework that incorporate various nonlinear models to represent the relationship between observed spectra and endmember fractions. Common models include polynomial models, neural networks, or other nonlinear functions.

5) **Complexity Handling**: handles complex spectral mixing scenarios, where phenomena such as spectral saturation, shadowing, and multiple interactions between materials occur.

6) **Noisy Data Handling**: can better handle noisy data incorporating techniques that reduce the impact of noise on the unmixing results and is robust in the presence of data imperfections.

7) **Data-Driven Approach**: NSU may not require explicit knowledge of endmember spectra because can learn the endmembers directly from the data, making it suitable for situations where endmembers are unknown or difficult to determine.

8) **Computational Complexity**: involves more complex optimization and modelling, which can result in increased computational demands compared to LSU.

It's important to note that NSU is a more advanced and computationally intensive technique compared to LSU but the choice between the two depends on the specific characteristics of the data and the complexity of the spectral mixing in the scene. NSU is particularly valuable when dealing with challenging scenarios, such as urban environments, vegetation canopies, and areas with significant nonlinear effects.

In our MATLAB-based implementation we test LSU because of the nature of the benchmark dataset of DLR, where the number such as the type of endmembers with their specific spectral libraries is given a priori.

2.3 Supervised Spectral Unmixing

The user provides information about the endmembers that are expected to be present in the scene. This information could be derived from ground-truth data, expert knowledge, or prior observations.
Steps involved:

- ***Endmember Selection****:* The user identifies and selects the endmember spectra that correspond to the known materials in the scene.
- ***Abundance Estimation***: Using the selected endmember spectra, the algorithm estimates the abundances of these endmembers in each pixel's spectrum.

Advantages

- Utilizes prior knowledge, leading to more accurate results.
- Can be tailored to specific applications.

Limitations

- Requires accurate knowledge of endmembers.
- May not perform well if endmembers are not accurately represented or if new materials are present.

2.4 Unsupervised Spectral Unmixing

In unsupervised spectral unmixing, the algorithm identifies endmembers and their abundances without relying on prior information from the user.
Steps involved:

- ***Endmember Extraction***: The algorithm automatically identifies potential endmembers within the data.
- ***Abundance Estimation***: The algorithm estimates the abundances of these automatically identified endmembers in each pixel's spectrum.

Advantages

- More automated, suitable when endmembers are unknown or numerous.
- Can discover new materials or endmembers not anticipated by the user.

Limitations

- May lead to less accurate results due to the absence of user guidance.
- Can be sensitive to noise and outliers in the data.

Each approach has its own strengths and weaknesses, and the choice between supervised and unsupervised spectral unmixing depends on factors such as the availability of prior information, the accuracy of endmember selection, the complexity of the scene, and the goals of the analysis. Some applications might benefit from a combination of both approaches, utilizing user-provided information where available and leveraging unsupervised techniques to explore unexpected materials or features in the data. (Cerra, et al., 2021)

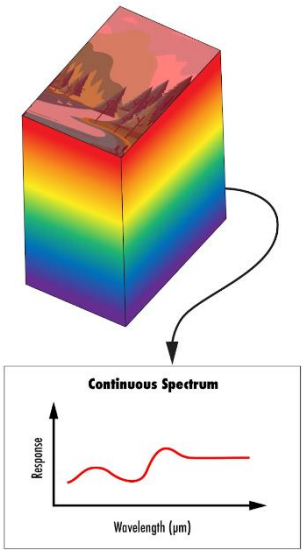
The process of spectral unmixing aims at providing accurate information at sub-pixel level on the scene, by decomposing the spectral signature associated with an image element in signals typically belonging to macroscopically pure materials, or endmembers. The contribution of a given material to the spectrum of an image element is a fractional quantity, usually named abundance.
The steps to follow for the realization of the program are:

1. **Estimation** of the number of materials present in the scene.
2. **Dimensionality reduction**, as an optional step carried out by removing non-relevant spectral ranges or projecting the data onto a new parameter space, which can be defined also based on results from the previous step.
3. **Endmember extraction**, in which the spectra related to materials present in the scene, often referred to as endmembers, are estimated.
4. **Abundance estimation**, in which the fractional coverage of each pixel is estimated in terms of the pure materials present on ground.

3 Imagery

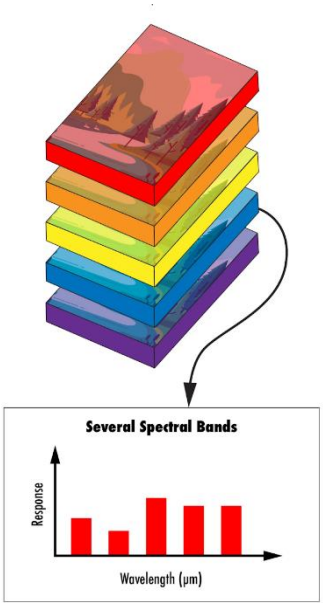
Application and capabilities strictly depend on what type of dataset we are working with, when talking about remote sensing field. In fact, an important assumption is the difference between **Hyperspectral** and **Multispectral** imaging, both used to capture information about the surface and its features but differ primarily in terms of the number of spectral bands they capture and the level of spectral detail they provide.

3.1 Hyperspectral Imaging:



- Hyperspectral imaging involves capturing data in many narrow and contiguous spectral bands across a wide range of wavelengths.
- The number of spectral bands in hyperspectral imagery can range from tens to hundreds or even more.
- This high spectral resolution enables hyperspectral imagery to provide very detailed information about the spectral characteristics of different materials and substances.
- Hyperspectral data is well-suited for applications that require precise identification and analysis of materials, such as mineral exploration, agriculture, environmental monitoring, and urban planning.

3.2 Multispectral Imaging:



- Multispectral imaging involves capturing data in a limited number of broader spectral bands, usually covering key regions of interest in the electromagnetic spectrum.
- The number of spectral bands in multispectral imagery is relatively small, typically ranging from a few to several tens.
- While multispectral imagery provides less detailed spectral information compared to hyperspectral imagery, it still offers valuable insights into the composition and characteristics of different materials.
- Multispectral data is commonly used in applications such as land cover classification, vegetation analysis, and urban planning.

Therefore, the main difference between these two approaches lies in the level of spectral detail and the number of spectral bands captured. Hyperspectral imagery provides a higher level of spectral detail with numerous narrow bands, while multispectral imagery captures a smaller number of broader bands. The choice between these two imaging techniques depends

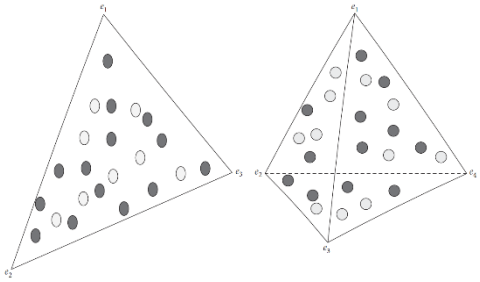
on the specific application and the trade-off between spectral resolution, spatial resolution, and data processing complexity.
After having learned these general theoretical notions necessary to understand what the project is based on and how to develop it, we started making an application plan for the analytics program to achieve our goal.
First, we chose the approach of the algorithm because there are two different strategies that lies on two different results for the project. These two strategies are called Supervised and Unsupervised Spectral Unmixing.
They are used to perform the task of decomposing mixed pixel spectra into their constituent endmember spectra and their corresponding abundances. These approaches differ in how they utilize prior information and user involvement during the unmixing process.

4 Linear Mixing Model

Linear mixing assumes that photons arriving at the sensor interact with only one substance. However, when the size of the mixed element is small and the incident photon has multiple reflections and refraction, it will interact with a variety of substances, leading to nonlinear mixing.
As we said before, for our purpose was took in consideration only the linear spectral unmixing, therefore let's look of its methods. Linear spectral unmixing methods are divided into four categories:

- **Geometric method**: it treats the vertices of a single body as endmembers. Therefore, they are mainly used for endmembers extraction.
- **Nonnegative matrix factorization (NMF)**: it has obvious nonconvexity and is easy to achieve local minimization. To solve this problem, the endmembers constraint and abundance constraint are briefly summarized.
- **Bayesian method**: the goal of this method is to achieve spectral unmixing by constructing endmembers and the maximum posterior probability of abundance. The likelihood function and prior information are sorted out, respectively. This method can realize both extraction and abundance estimation of endmembers and also can only realize abundance estimation when the endmembers are known.
- **Sparse regression**: its goal is to estimate the abundance with the regression technique when the endmembers spectrum signatures are known. This part will be sorted out from two aspects of fitting error and sparse regression algorithm. (Jiaojiao Wei, 2020)

4.1 Geometric Method



Ideally, the spatial distribution of all pixels in the hyperspectral dataset is considered being in a convex simplex. A convex simplex contains all data points. In the convex simplex space formed by data points, the vertex of the convex simplex is the endmembers. Therefore, the endmembers extraction is to obtain the endmembers by finding the vertex of the corresponding convex simplex. In the two-dimensional spatial data, the convex simplex is regarded as a triangle, while, in the three-dimensional spatial data, the convex simplex is a pyramid, as shown in Figure, respectively. A convex simplex is a polyhedron in multidimensional space. Endmembers extraction is one of the important steps of

hyperspectral linear unmixing. Endmembers extraction methods are based on geometry, and the conventional methods include for example pixel purity index (PPI), N-FINDR.

4.2 Spectral Unmixing Based on Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) is a data analysis technique used to decompose a matrix into two or more smaller matrices, ensuring that all the involved matrices have only non-negative values. This method is used in data processing to extract hidden patterns or components from raw data, making it easier to interpret. The study of hyperspectral unmixing based on NMF is also based on blind source separation theory (BSS), which is a signal processing technique used to separate a set of mixed signals into their original source signals without prior knowledge of the source signals or the mixing process.
Here's how NMF works:

1. **Input Matrix**: we start with an input data matrix, often referred to as matrix $\boldsymbol{V}$, which contains the data we want to analyse. This data can represent anything, such as text documents, images, or numerical values.
2. **Factorization**: The goal of NMF is to factorize this input matrix V into two smaller matrices, **W** and $\boldsymbol{\theta}$, so that V can be approximated by their multiplication. This is formalized as: $\boldsymbol{V = \theta W}$.
    - **Matrix W**: it represents the features or hidden patterns in the data. Each column of W is called a "component" and represents a set of relevant features.
    - **Matrix $\boldsymbol{\theta}$**: it contains the weights of how the components in $\boldsymbol{W}$ contribute to reconstruct each data sample or data point in $\boldsymbol{V}$.
3. **Non-Negativity Constraint**: a key aspect of NMF is that all entries (values) in $\boldsymbol{W}$ and must $\boldsymbol{\theta}$ be non-negative. This means that negative values are not allowed in the components or weights.
4. **Optimization**: The goal is to find matrices W and H that minimize the error between the original matrix $\boldsymbol{V}$ and its approximation $\boldsymbol{\theta * W}$. This can be achieved through optimization methods that adjust matrices $\boldsymbol{W}$ and $\boldsymbol{\theta}$ to reduce the reconstruction error.

4.3 Spectral Unmixing Based on the Bayesian Method

The unmixing of mixed pixels is an ill-conditioned inverse process, so the result is not unique. An ill-conditioned inverse process refers to a situation where you are attempting to calculate the inverse of a matrix, but the matrix is so "ill-conditioned" that the calculation becomes challenging or unstable. In other words, the matrix is very close to being singular (non-invertible, an invertible matrix that, when multiplied for the original matrix, yield to the identity matrix, which has ones on its main diagonal), and even small errors in calculations can lead to imprecise or even unstable results. Bayesian method can incorporate meaningful prior information into the modelling process through good statistical means to model the variability and uncertainty existing in spectral data, abundance, and endmembers. The main idea of Bayesian method is to deduce the posterior probability density through the prior distribution and likelihood probability:

$$p(\theta, W|Y, \varphi) = \frac{p(Y|\theta,w,\sigma^2)p(\theta|\varphi)p(W|\varphi)p(\sigma^2|\varphi)}{p(Y)},$$

where , σ^2, θ, and W obey some probability distribution, and the parameters of these distributions functions make up the super-parameter φ. $p(Y|\theta, w, \sigma^2)$ is the **likelihood function**, $p(\theta|\varphi)$ is the **prior distribution** of endmembers, $p(W|\varphi)$ is the prior distribution of abundance, and $p(\sigma^2|\varphi)$ is the prior distribution of noise.

1. **Likelihood Distribution**: let the mixed model of the observed spectrum be written as a vector: $Y = \theta w + e$, where e is noise. At present, the expression forms of likelihood function are mainly divided into two types. We can express the first kind of likelihood function model as a multivariate Gaussian model]. Its expression is

$$p(y|w, \beta) = N(y|w\theta, \beta^{-1}I) = (2\pi)^{-N/2}\exp\left[\frac{-\beta}{2}\|y - \theta w\|_2^2\right].$$

2. **Prior Distribution**: it summarizes the prior distribution of endmembers from four aspects. We can regard uniform distribution terminal element as the nonnegative real number information distribution, the uniform distribution between zero and infinity. the information of uniform distribution is insufficient. Gaussian distribution is modelled with Gaussian distribution as the endmembers. On the premise that the endmembers obey the Gaussian distribution, maximum expectation algorithm is used to solve the mean value and variance, but they cannot measure the variability of endmembers spectrum. The beta distribution of endmembers can express the skewness of distribution and spectral variability. The distribution treats each terminal element of each band as an inferred beta distribution. Inferred means that the beta distribution is estimated from observed data, instead a beta distribution is a continuous probability distribution used to model the variability that assumes values between 0 and 1. It's characterized by two main parameters, typically denoted as α (alpha) and β (beta), which determine its shape. The probability density function (PDF) of the beta distribution is defined as:

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} * x^{\alpha-1} * (1 - x)^{\beta-1}.$$

Where:

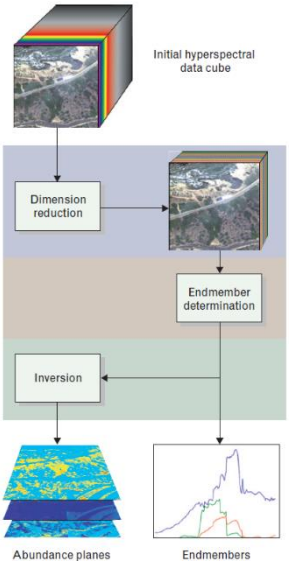
- x is a variable ranging between 0 and 1.
- α and β are the distribution parameters.
- $B(\alpha, \beta)$ is the beta function, which ensures that the area under the distribution curve sums to 1.

4.4 Spectral Unmixing Based on the Sparse Method

The aim is to estimate the abundance of endmembers in spectral images based on the known endmembers' spectral library. It assumes that each observed feature linearly combines only a few spectra from a known spectral library. The known endmembers spectrum library contains a large amount of spectral information of pure endmembers, from which several endmembers' spectra are selected to approximate the spectra of mixed pixels, which will definitely lead to sparsity of abundance.

5 Stage of Unmixing

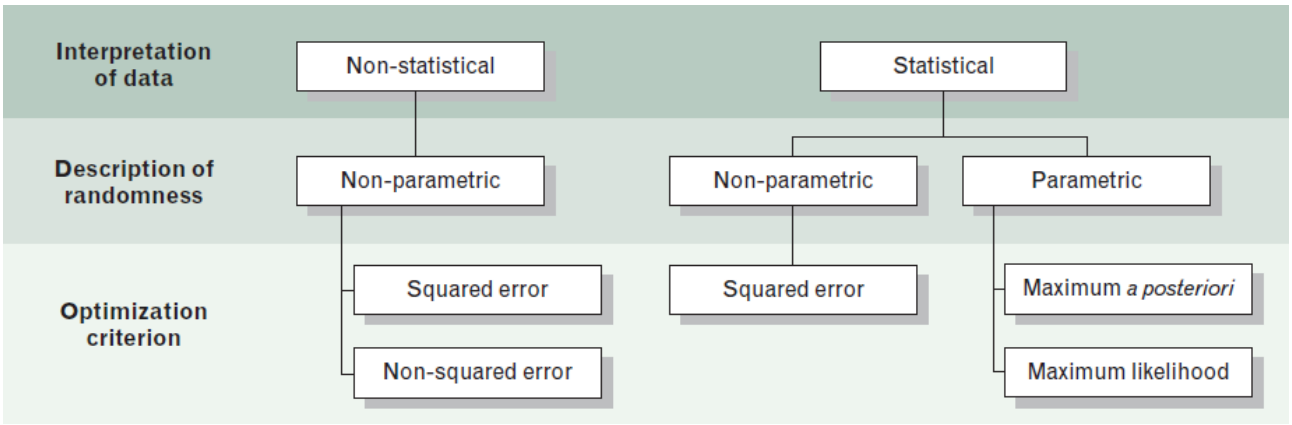
We can decompose the complete end-to-end unmixing problem as a sequence of three consecutive stages: dimension reduction, endmember determination, and inversion.
In the dimension-reduction stage we reduce the dimension of the data in the scene. This step is optional and is invoked only by some algorithms to reduce the computational load of subsequent processing. In the endmember-determination stage we estimate the set of distinct spectra (endmembers) that constitute the mixed pixels in the scene. Finally, in the inversion stage we generate abundance planes that allow us to estimate the fractional abundances for each mixed pixel from its spectrum and the endmember spectra.



The top-down examination of the unmixing problem yields a set of three criteria that categorize unmixing algorithms. In order, these criteria are:

- **Interpretation of data**, which indicates how an algorithm interprets mixed-pixel spectra;
- **Description of randomness**, which indicates how an algorithm incorporates the randomness of the data;
- **Optimization criterion**, which indicates what objective function is being optimized by the algorithm.

An algorithm can interpret the data it processes in one of two ways. If an algorithm processes a mixed pixel by using statistical measures (e.g., means or covariances), then the algorithm is statistical. Algorithms that perform their task without statistical measures are non-statistical. This distinction becomes especially important in target detection, among other places, where statistical characterizations of nontarget behaviour (background or clutter) can complicate the detection of low-probability targets. (Keshava N. , 2003)



To represent the aggregate behaviour of data, algorithms can use statistical measures, which may also be parameters in analytical expressions that represent all or part of a probability density function.

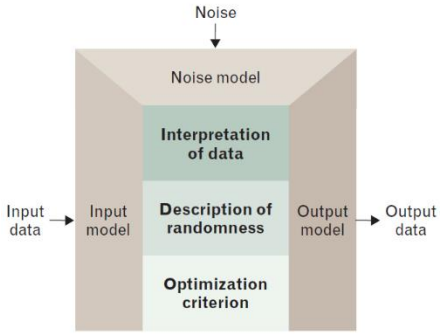
- Techniques vested with the assumption that the received data originate from a parameterized probability density function are considered **parametric**.
- Algorithms that do not impose the structure of a particular density function are **nonparametric**.

These algorithms are deemed optimal if they optimize an objective function.

- **Parametric algorithms** optimize some combination of densities and fall in the category of either maximum likelihood or maximum *a posteriori* solution.
- **Non-parametric algorithms** can utilize one of a multitude of cost functions, but clearly the most prevalent optimization criterion is minimization of squared error.

Each of these criteria captures a fundamental aspect of how the algorithm relates to the data it processes, and because the criteria are not specific to any algorithm, the hierarchical partitioning they induce map from one type of algorithm to another. The spectral unmixing process consists of the concatenation of three consecutive and disparate operations, and to consolidate their assessment it has been applied the same criteria to each taxonomy to consistently organize the algorithms at every stage.

In this bottom-up approach we can identify boundaries for delineating algorithms based on the operating characteristics that are unique to each stage. The features that distinguish algorithms are organized into three categories: output characteristics, input characteristics, and noise modelling.



5.1 Dimensionality Reduction (optional)

Dimension-reduction algorithms do not reduce the dimension of data with the goal of reconstructing an approximation to the original signal. Instead, the goal of dimension

Interpretation of data		Non-statistical	Statistical	
Description of randomness		Non-parametric		
Optimization criterion		Squared error		Non–squared error
				Signal-to-noise-ratio
Specifics	**Outputs**	Orthogonal axes		Non-orthogonal axes
	Inputs	Utilizes reduced data set	Utilizes all data in scene	
	Noise	No noise in signal model		Additive observation noise
Algorithm		Optical real-time adaptive spectral identification system (ORASIS)	Principal-component analysis	Maximum noise fraction Noise-adjusted principal components

reduction is to arrive at a minimal representation of the signal in a lower-dimensional space that sufficiently retains the requisite information for successful unmixing in the lower dimension.

Dimensionality Reduction Algorithms

The category of statistical algorithms derives their transformations from statistical information about the data and can be further differentiated by the optimization criterion that is utilized.

5.1.1 Principal- component analysis (PCA)

It's technique based on squared error, identifies orthogonal axes for dimension reduction by performing an eigen decomposition of a covariance estimate of the data,

$$\boldsymbol{\Gamma_x} = \boldsymbol{E}[(\boldsymbol{x}-\boldsymbol{\mu_x})(\boldsymbol{x}-\boldsymbol{\mu_x})^T] = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}}[\boldsymbol{x(n)}-\boldsymbol{\mu_x}][\boldsymbol{x(n)}-\boldsymbol{\mu_x}]^T$$

where μ_x is the mean vector of the pixel set. The resulting eigen decomposition can be expressed as $\boldsymbol{\Gamma_x} = \boldsymbol{U\Sigma U^T}$, where $\boldsymbol{U}$ is a unitary matrix of eigenvectors, and $\boldsymbol{S}$ is a diagonal matrix of eigenvalues. The magnitude of an eigenvalue indicates the energy residing in the data along the component of the data parallel to the associated eigenvector. The larger eigenvalues identify basis components whose average contribution to $(\boldsymbol{x}-\boldsymbol{\mu_x})$ in the squared-error sense is greater than those with smaller eigenvalues. Hence the effective dimensionality of the data can be estimated by counting the number of significantly non-zero eigenvalues.

5.1.2 Maximum noise fraction (MNF)

This technique optimizes the signal-to-noise ratio (SNR). This approach, based on the LMM (Linear Mixing Model), requires estimating the covariance $\hat{\boldsymbol{\Gamma}}_{\boldsymbol{w}}$ for the additive noise in addition to the covariance $\hat{\boldsymbol{\Gamma}}_{\mathbf{x}}$ of the data. Given a vector v, the ratio of noise energy in v to the received signal energy is given by

$$\frac{\boldsymbol{v^T\hat{\Gamma}_w v}}{\boldsymbol{v^T\hat{\Gamma}_x v}}$$

The axes that optimize this ratio are the left-hand eigenvectors of $\hat{\boldsymbol{\Gamma}}_{\mathbf{w}}\hat{\boldsymbol{\Gamma}}_{\mathbf{x}}^{-\mathbf{1}}$ and, unlike the axes for PCA, are not necessarily orthogonal. They do, however, identify and order the components of the received signal possessing the maximum SNR.

5.2 Endmember Determination

The objective of endmember-determination algorithms is to estimate the constituent spectra that occupy the columns of $\boldsymbol{S}$ in the LMM. Result of endmember extraction is a set of endmember spectra that collectively explain the data variance and can be used for unmixing purposes.
From a strictly mathematical viewpoint, the determination of $\boldsymbol{S}$ is comparable to the estimation of a nonorthogonal subset of basis vectors. More physical interpretations, however, begin with the stipulation that the vectors must have non-negative entries to be physically realizable. Furthermore, endmembers should retain physical characteristics of the constituent substance such as absorption bands and spectral intervals of high and low reflectance. Consequently, identifying endmembers that satisfy both physical and

<table>
<tr><td colspan="2">Interpretation of data</td><td colspan="3">Non-statistical</td><td colspan="5">Statistical</td></tr>
<tr><td colspan="2">Description of randomness</td><td colspan="4">Non-parametric</td><td colspan="4">Parametric</td></tr>
<tr><td colspan="2" rowspan="2">Optimization criterion</td><td colspan="3">Non-squared error</td><td colspan="2" rowspan="2">Squared error</td><td colspan="3">Maximum likelihood</td></tr>
<tr><td colspan="3">Geometric</td><td colspan="3">Gaussian</td><td>Non-Gaussian</td></tr>
<tr><td rowspan="4">Specifics</td><td>Outputs</td><td colspan="6">Deterministic endmembers</td><td>Stochastic (Gaussian) endmembers</td><td>Deterministic endmembers</td></tr>
<tr><td>Inputs</td><td colspan="3">Convex hull of pixel spectra</td><td colspan="2">All pixel spectra and abundances</td><td colspan="3">All pixel spectra</td></tr>
<tr><td>Noise</td><td colspan="3">No noise in signal model</td><td colspan="4">Additive observation noise</td><td colspan="2">No noise in signal model</td></tr>
<tr><td></td><td colspan="3"></td><td>Additive meas. noise</td><td></td><td>Gaussian</td><td></td><td></td></tr>
<tr><td>Algorithm</td><td colspan="2">Minimum-volume transform</td><td>Penalty-based shrink-wrapping</td><td>Block least squares</td><td>Total block least squares</td><td>Fuzzy K-means</td><td>Non-linear least squares</td><td>Gaussian class estimation</td><td>Independent component analysis</td></tr>
<tr><td></td><td>DPFT</td><td>FPFT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

mathematical imperatives is a considerable challenge, making autonomous endmember determination the hardest part of the unmixing problem.

Endmember Extraction Algorithms

Statistical methods of endmember determination identify endmembers by optimizing objective functions derived from the statistics of the data. Furthermore, the non-parametric algorithms attempt this without optimizing a parametric model, but instead by minimizing an objective function using the statistics derived from the data. A common objective function for this goal is squared error.

5.2.1 Fuzzy *K*-means

Partitions algorithm, which is a variation on the well known *K*-means iterative clustering algorithm that progressively minimizes an objective function $\boldsymbol{J_q(A,S)}$ to simultaneously arrive at optimal estimates of deterministic endmembers and abundances for $\boldsymbol{N}$ pixels. This function is expressed as

$$J_q(A,S) = \sum_{i=1}^{M}\sum_{n=1}^{N}(A_{in})^q(d_{in})^2$$

A*in* is an estimate of the $\boldsymbol{ith}$ abundance for the $\boldsymbol{nth}$ pixel. The factor $\boldsymbol{d_{in}}$ is the squared error between the $\boldsymbol{nth}$ pixel and the $\boldsymbol{ith}$ centroid, or estimated endmember $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ and is given by

$$(\boldsymbol{d_{in}})^{\boldsymbol{2}} = [\boldsymbol{x}(\boldsymbol{n}) - \hat{\boldsymbol{s}}_{\boldsymbol{i}}]^{\boldsymbol{T}}\boldsymbol{W}[\boldsymbol{x}(\boldsymbol{n}) - \hat{\boldsymbol{s}}_{\boldsymbol{i}}]$$

The weighting matrix $\boldsymbol{W}$ applies different weights to each abundance, and the inverse class covariance may be inserted here. The minimization of $\boldsymbol{J_q(A,S)}$ is constrained by physical restrictions on the values of abundances, and these requirements are inserted into the estimator for $\boldsymbol{A}$. Moreover, the shapes of the regions associated with each class vary with $\boldsymbol{q}$. Suggested values fall in the range $\boldsymbol{1.5 < q < 3.0}$. The optimization of $\boldsymbol{J_q(A,S)}$ is accomplished iteratively until a minimum value is achieved by using final estimates of $\widehat{\boldsymbol{S}}$ for the endmembers and $\widehat{\boldsymbol{A}}$ for the abundances.

5.2.2 Non-Linear List Squares (NNLS)

This method models the additive noise **w** in the LMM as Gaussian, and assumes both **S** and **a** are deterministic unknowns for a single mixed pixel **x**. This technique integrates *a priori* knowledge into least-squares formulations for unmixing by initializing an iterative estimator for **S** and **a** with guesses for what endmembers and abundance values should be. Associated covariances are set to values that quantify the confidence of the initial guesses. As a consequence of estimating **S** and **a** concurrently, the forward density can be alternately expressed as $\boldsymbol{p_{x|Sa}(x|Sa)}$. By virtue of Gaussianity, the likelihood function is quadratic in form, but the matrix product **Sa** recasts the simpler traditional maximum-likelihood optimization into a nonlinear least-squares problem.
This is a technique that attempt to fuse the geometric interpretation of linear mixing with Gaussian mixture modelling and maximum-likelihood estimation techniques.

5.2.3 Stochastic mixing model (SMM)

This approach introduces the concept of hard endmember classes as the stochastic extension of deterministic endmembers, and it assumes that all data in a scene are Gaussian and arise from a linear combination of at least one hard class. Geometrically, each stochastic endmember describes one cluster of pixels on the perimeter of the data cloud that is parameterized by a mean, covariance, and prior probability $\boldsymbol{N(\mu, \Gamma, \pi)}$ so that every mixed pixel belongs to a combination of hard classes. If each combination of hard classes is a mixed class with its own Gaussian statistics (i.e., mean, covariance, and prior probability), then optimal unmixing determines the combination of classes having the greatest likelihood of realizing that pixel. After an initial partition of the data, the reclassified pixels are used to generate new class statistics, and the procedure repeats until the SMM class parameters cease to change.

5.2.4 Geometric Endmember Determination

Geometric approaches to endmember determination are non-parametric and non-statistical, and instead exploit the strong parallelism between the LMM and the theory of convex. These approaches rely on the assumption that pixel spectra from a scene reside within a

high-dimensional volume, and by virtue of the LMM, endmembers must reside at the extremities of this volume for mixed pixels to arise.
The determination process consists of two steps.

1. **Data reduction**, which like dimension reduction essentially minimizes computation by discarding information, in the specific the process discards pixels, not bands. Only the perimeter of the volume occupied by the scene data is necessary to establish the location of the endmembers, and consequently the pixels residing within the convex hull of the data are discarded.
2. **Shrink-Wrapping**, which employs a class of procedures known as minimum-volume transforms (MVT) to iteratively fit a multifaceted simplex of minimum volume around the convex hull, where a simplex is a geometric surface in an *n*-dimensional hyperspace defined by exactly ***n* + 1** vertices.

The PPI (Pixel Purity Index) and NFINDR (N-FINDR) are two main algorithms used for endmember extraction in hyperspectral image analysis.

PPI

- PPI is a **geometric** method for endmember extraction used to find the most spectrally pure pixels in a hyperspectral image.
- The algorithm works by searching for pixels that have endmembers lying on the extreme convex hull vertices of the data distribution.
- It assumes that pure pixels are located on the boundary of the data distribution, making them suitable endmembers.
- The result is a set of pure pixel spectra that represent the endmembers.

NFINDR

- It is a **statistical** method for endmember extraction.
- NFINDR aims to find endmembers that can best describe the spectral variability in the hyperspectral data while adhering to the non-negativity constraint, which is often valid for physical materials.
- The algorithm iteratively selects endmembers by fitting them to the data's residuals (the difference between the observed spectrum and the linear combination of endmembers).
- The process continues until a pre-defined number of endmembers is obtained.
- The
- NFINDR is useful for extracting endmembers from hyperspectral data with a linear mixing model assumption.

Both PPI and NFINDR are useful in hyperspectral unmixing, where the goal is to estimate the fractions (abundances) of endmembers within each pixel. The endmembers extracted by these algorithms serve as the basis for spectral unmixing models. The choice between these methods may depend on the nature of the data, the specific application, and user preferences, as they have different underlying principles for endmember selection.

5.2.4.1 NFINDR

Non-Negative Matrix Factorization Based on Determinant Ratio is an algorithm used for endmember extraction in hyperspectral data analysis that aims to identify pure spectral signatures within a hyperspectral data cube by iteratively searching for endmembers that maximize a determinant ratio criterion.
Here's an overview of how NFINDR works:

Let $\boldsymbol{E}$ be the augmented matrix of endmembers:

$$\boldsymbol{E} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \dots & \mathbf{1} \\ \boldsymbol{e_1} & \boldsymbol{e_2} & \dots & \boldsymbol{e_l} \end{bmatrix},$$

where $\boldsymbol{e_i}$ is a column vector containing the bands of endmember i. The volume (V) of the simplex formed by using the endmember estimates is proportional to the determinant of $\boldsymbol{E}$:

$$\boldsymbol{V(E)} = \frac{\mathbf{1}}{(\boldsymbol{l}-\mathbf{1})!}\boldsymbol{abs}(|\boldsymbol{E}|)\ ,$$

where $(\boldsymbol{l}-\mathbf{1})$ is the number of dimensions occupied by the data. This algorithm finds the set of pixels with the largest possible volume $(\boldsymbol{V})$, by "inflating" a simplex inside the data. The procedure begins with a random set of vectors.
To refine the estimate of the endmembers, every pixel in the image must be evaluated as to its likelihood of being a pure or nearly pure pixel.

To do this, the volume must be calculated with each pixel in place of each endmember.
A trial volume is calculated for every pixel in each endmember position by replacing that endmember and finding the volume. If the replacement results in an increase in volume, the pixel replaces the endmember. This procedure is repeated until there are no more replacements of endmembers.
Once the pure pixels are found, their spectrum can be used to make the original image unmixed by using either linear inversion or non-negatively constrained least squares. This produces a set of images, each of which shows the abundance of an endmember. The algorithm has been successfully used to reproduce published endmember spectrum and component images. (X. Zhang, 2009)

5.2.4.2 PPI

The PPI calculation selects a random vector through the n-dimensional data cloud (each random vector is constrained to pass through the center - the mean value - of the data cloud) and then projects each pixel in the image onto the random vector. A histogram is calculated which shows how the pixels are projected onto the random vector. Pixels are considered pure if they fall into the tails of the histogram distribution. The histogram tails are defined by the PPI threshold value entered by the user. Throughout the PPI process, ENVI keeps track of which pixels are identified as pure for each random vector. Each time the same pixel is identified as pure using a new random vector, its value in the output image is incremented by one. Then a new random vector is chosen, and the process is repeated. The result of the PPI routine is an image where the value of each pixel corresponds to the number of times it was identified as a pure pixel during all the PPI iterations. (NV5GeospatialSoftware, s.d.)

5.3 Inversion Taxonomy

The objective of these algorithms is to determine the fractional presence of each endmember in the received pixel spectrum or, in terms of the LMM, find the vector **a** whose entries weight the columns of **S** to yield **x**.
Any estimate of **a** should obey the following constraints for:

1. **Non-Negativity**: the abundances should be non- negative to be meaningful in a physical sense ($\boldsymbol{a_i} \geq \mathbf{0}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{M}$).
2. **Purity**: a fractional abundance coefficient cannot exceed 1($\boldsymbol{a_i} \leq \mathbf{0}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{M}$).
3. **Additivity**: full additivity requires the abundances for a mixed pixel to sum to one, with the implicit assumption that all the endmembers comprising the pixel spectrum in **x** are present in the columns of **S**, $\sum_{i=1}^{M} \boldsymbol{a_i} = \mathbf{1}$.
   Partial additivity is a generalization that requires only the sum of abundances to be less than or equal to one, and it applies when the set of endmembers in the scene might be incomplete, which occurs when $\sum_{i=1}^{M} \boldsymbol{a_i} < \mathbf{1}$.

Inversion Algorithms

Inversion algorithms are dominated by approaches that invoke some aspect of the method of

Interpretation of data		Non-statistical							Statistical		
Description of randomness		Non-parametric									
Optimization criterion		Non-squared error			Squared error						
Specifics	**Outputs**	Continuous-valued abundances									
	Full additivity	No	Yes		No	Yes	No				Yes
	Non-negativity	No	Yes		No		Yes		No		Yes
	Purity	No	Yes		No						Yes
	Inputs	Deterministic endmembers								Endmembers not required	
	Noise	Additive obs. noise	No noise in signal model						Additive observation noise		
Algorithm		Filter vector algorithm	I-diverg-ence	Confidence estimates	ULS	ULS with full additivity	NNLS	Project-ion on convex sets	MVUE	Ground-truth-based estimators	Fuzzy *K*-means
										Classical / Inverse	

least squares. Many statistical and parametric algorithms indirectly fall under the umbrella of least-squares analysis either by the squared-error-minimizing properties of the singular-value decomposition and eigen decomposition, or the Euclidean geometry of Gaussian analysis.

5.3.1 Unconstrained Least Squares (ULS)

Solution for **a**: $\widehat{\boldsymbol{a}}^{\boldsymbol{U}} = (\boldsymbol{S^T S})^{-1} \boldsymbol{S^T x}$. Under the assumption of the LMM and no additive noise, this unconstrained estimate for **a** minimizes $|\boldsymbol{x} - \boldsymbol{S}\widehat{\boldsymbol{a}}^{\boldsymbol{U}}|_{\mathbf{2}}$. This estimate exists when there are more bands than columns (a reasonable assumption for hyperspectral sensing), and when **S** has full column rank (i.e., endmembers are linearly independent).

The simplest variations on $\hat{\boldsymbol{a}}^{\boldsymbol{U}}$ incorporate the key physical constraints on **a** by constraining the set of allowable solutions for $\hat{\boldsymbol{a}}$. Full additivity requires the abundances in **a** to sum to one, and this requirement restricts the solution to lie on the hyperplane given by $\sum_{i=1}^{M} \boldsymbol{a_i} = \mathbf{1}$. The general solution for a least-squares estimates having linear constraints is given by:

$$\hat{\boldsymbol{a}}^{\boldsymbol{F}} = \hat{\boldsymbol{a}}^{\boldsymbol{U}} - (\boldsymbol{S^T S})^{-1}\boldsymbol{Z^T}[\boldsymbol{Z}(\boldsymbol{S^T S})^{-1}\boldsymbol{Z^T}]^{-1}(\boldsymbol{Z}\hat{\boldsymbol{a}}^{\boldsymbol{U}} - \boldsymbol{b})$$

where **Z** is a 1 ¥ *M* row vector having all ones and **b** = 1. Closer examination of ˆ**a** *F* reveals that this solution consists of the unconstrained least-squares solution ˆ**a***U* with an additive correction term that depends on the matrix of endmembers **S** and the error incurred by ˆ**a***U* in satisfying the full additivity constraint. The second constraint, non-negativity, is not as easily addressed in closed form as full additivity. Minimizing $|\boldsymbol{x} - \boldsymbol{Sa}|^{\mathbf{2}}$ while maintaining $\boldsymbol{a_i} \geq \mathbf{0}$ for $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{M}$, falls in the domain of quadratic programming with linear inequalities as constraints.

5.3.2 Non-Negative Least-Squares Algorithm (NNLS)

The approach here is to iteratively estimate **a** and, at every iteration, find a least-squares solution for just those coefficients of **a** which are negative by using only the associated columns of **a**. By selectively refining and moving those entries of **a** which still violate the non-negativity condition, and integrating the newfound coefficients with the existing non-negative values, the algorithm acquires a new estimate of **a**. The procedure is repeated until the algorithm converges to a final non-negative estimate $\hat{\boldsymbol{a}}^{\boldsymbol{NN}}$, which has a subset of entries that are exactly zero.

6 DLR HySU - A Benchmark Dataset for Spectral Unmixing

As a new approach of the subject the first assumption was to find a good base where to start to understand the problem of unmixing. The DLR paper was a good choice for starting to build our knowledge about the problem. The DLR project aims at providing information about the differences between different approaches and different methods useful for the spectral unmixing. The DLR team provide by himself to create a new dataset that allow them to study from scratch instead of starting from uncertain results.

6.1 Targets

The team deployed five different materials: bitumen, red painted metal sheets, blue fabric, red fabric, and green fabric. The materials have been chosen to cover as many different ranges of hyperspectral signs. The organization of the targets consists of five different materials organized in five different dimensions targets to test how precise the unmixing algorithm can be. In addition to the targets were placed three small sub-pixel targets hidden in the surrounding area to test target detection algorithms.

6.2 Dataset acquisition

The targets acquisition consists of two different approaches:

- Airborne measurement: acquired from an airplane using an HySpex camera and an RGB camera.
- Field measurements: acquired from a ground-based reflectance measurement using an SVC HR camera.

6.2.1 HySpex Camera

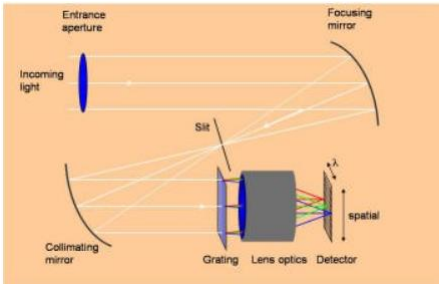
The imaging spectrometer consists of two HySpex pushbroom cameras:

- HySpex VNIR-1600 (Visible Near Infrared): its features 1600 geometric pixels and 128 spectral channels covering the wavelength range 416–992 nm.
- HySpex SWIR-320m-e (Shortwave Infrared): it measures the upwelling radiation in the wavelength range 968–2498 nm at 256 spectral channels for 320 geometric pixels.
- 3K system: it consists of three 35 mm Canon EOS cameras equipped with Zeiss 50 mm lenses used to acquire RGB images.

6.2.2 HySpex

The camera images the scene line by line using the so-called "pushbroom" scanning mode. One narrow spatial line in the scene is imaged at a time, and this line is split into its spectral components before reaching the sensor array. On the 2d sensor array, one dimension is used for spectral separation and the second dimension is used for imaging in one spatial direction. The second spatial dimension in the scene arises from scanning the camera over the scene (aircraft movement). The result can be seen as one 2d image for each spectral channel, or alternatively every pixel in the image contains one full spectrum.

6.2.3 Operating principle



The camera fore optic images the scene onto a slit which only passes light from a narrow line in the scene. After collimation, a dispersive element (in our case a transmission grating) separates the different wavelengths, and the light is then focused onto a detector array. The net effect of the optics is that for each pixel interval along the line defined by the slit, a corresponding spectrum is projected on a column of detectors on the array. The data readout from the array thus contains a slice of a hyperspectral image, with spectral information in one direction and spatial (image) information in the other. By scanning over the scene, the HySpex camera collects slices from adjacent lines, forming a hyperspectral image or "cube", with two spatial dimensions and one spectral dimension. Note that the scanning is often intrinsic to the application: In remote sensing, scanning is provided by aircraft or satellite movement. Also, in many industrial quality control applications, products conveniently pass the sensor on their conveyor belt.

6.2.4 HySpex Subsets

The HySpex data were postprocessed to build a consistent benchmark dataset, indeed also the VNIR data were kept from the acquisition and the spectral bands above the 900 nm have been discarded. The resulting dataset consists of 135 spectral bands in the range of 417-903 nm. Each subset corresponds to a given number of materials K present in the scene. The team defined five different HySpex spatial subsets for the DLR HySU benchmark, organized as follows:

- **Full** (86 x 123) contains the whole area of interest and its surroundings, with multiple materials present.
- **All Targets** (42 x 24) includes all targets of all sizes, within a non-homogeneous background containing grass, a reference white panel and some image elements close to the road.
- **All Targets Masked** (42 x 24, with a total of 884 valid pixels) is the same as above, but with non-homogeneous areas in the background masked out.
- **Small Targets** (12 x 12) contains only the 0.25 and 0.5m targets. As the HySpex data are resampled to a 0.7m grid, in this subset all pixels are mixed with the exception of the surrounding grass.
- **Large Targets** (13 x 16) represents the subset containing only the 3m targets (five in total) and the surrounding homogeneous grass. This subset is aimed at providing the easiest setting for dimensionality estimation, endmember extraction with pure pixel assumption, and abundance estimation.

6.2.5 SVC (Spectra Vista Corporation) HR 1024i Camera

For the ground truth dataset, they made reflectance measurements of the reference targets with an SVC HR 1024i field spectrometer equipped with a 4° lens optic. This camera is used to compare the HySpex acquisition results with the in-situ values of the reflectance, and if the

HySpex measurement lie with the SVC acquisition then the dataset is suitable for the unmixing process.

6.3 Dimensionality Estimation

This step was carried out before identifying the materials within a scene using endmember extraction algorithms, whenever their number is not known a priori. The output of dimensionality estimation is an integer representing the estimated number of dimensions, which is usually considered equal to the number of different materials within the scene. The team applied two different algorithms to identify the real informational content of a scene using eigenvalues analysis after projecting the data onto a suitable space.
The two methods are: the Hyperspectral Signal Identification by Minimum Error (HySime) and the Harsanyi–Farrand–Chang (HFC). HySime is based on the list square error minimization, instead HFC aims at separating noise and signal eigenvalues, being formulated as a detection problem. Another important thing to notice is that both the algorithms were tested on different datasets to test the case where other targets were added also the methods had to recognize them, therefore the number of materials extracted from the scene had to be increased.

6.4 Endmember Extraction

This process is useful to identify spectra related to materials which are homogeneous at a relevant scale in a hyperspectral scene. All the algorithms used by DLR team works under the pure pixel assumption, which means that every pixel of the image took in consideration is composed only by one material and not by a composition of them. The algorithms took in consideration are:

- **Vertex Component Analysis (VCA)**: it iteratively determines endmembers as extreme pixels on the convex hull and performs orthogonal subspace projection (OSP) with respect to the determined endmembers, taking into account noise influences in the process;
- **N-FINDR**: it initializes the endmembers as random pixels in the variable space and iteratively substitutes them with their spectral neighbours, keeping the final set spanning the maximum volume;.
- **Automatic Target Generation Process (ATGP)**: it's also based on OSP, but its initialization parts lying to the VCA method.
- **Pixel Purity Index (PPI)**: it selects extreme pixels in the data cloud projected in the variable space by drawing a set of random lines and choosing pixels marked more often extremes.

The algorithms performances were evaluated by associating to each material in HySpex spectral library the extracted endmember yielding the minimum spectral angle. An important thing to take on mind is that a single extracted endmember could be the nearest neighbours to two different materials in the library, but this doesn't bias the results of the analysis, as both spectral angles cannot be simultaneously small. All algorithms with pure pixel assumptions

were tested on the *Large Targets* dataset and their performance is reported in Figure. The two algorithms that obtained the best results are VCA and N-FINDR.
It's important to note that ATGP is missing the green fabric endmember and the largest distortion for bitumen is because the endmember chosen is a mixed pixel with the adjacent red metal sheets.
As the team expected the results obtained from algorithms with no pure pixel assumption, working on *Small Target* yield the best performance.
However, all algorithms had problems with the bitumen identification, because it's difficult to retrieve in mixed setting due to the absence of clear absorption features in the analysed spectral range. Furthermore, also green fabric is a difficult material to find, because some spectral features in common with the surrounding grass spectra.

6.5 Abundance Estimation

This is the last step of DLR project, in which the materials abundances have been estimated. Four algorithms were evaluated: unconstrained least squares (UCLS), non-negative least squares (NNLS), fully constrained least squares (FCLS) and least squares with least absolute shrinkage and selection operator (LASSO). All these algorithms are based on least squares minimization, but the constraints applied on the abundances are distinct: UCLS finds the plain least squares solution without any constraints; NNLS and FCLS both require abundances to be non-negative (the so-called non-negativity constraint), with FCLS requiring in addition that the abundances sum to one (the so-called sum-to-one constraint). Given that all the tested algorithms were tested to perform for each pixel independently, the results had to be obtained only once for the full subset. In conclusion the team found that the best results of estimation across different target materials and size were given by the NNLS and LASSO algorithms. Till that moment the DLR team only used HySpex image as input for the abundance estimation process. After that they test the robustness of the obtained results with the spectral library containing the SVC acquisitions on the ground (the so called Ground Truth).

The comparison of both approaches leads to interesting conclusions dependent on target material and size. For all target sizes, it is apparent that the results are fairly robust for red metal, red fabric and green fabric, but bitumen and blue fabric display some degree of sensitivity to the spectral library used.
There is also a clear dependence on target size.
The target area error for the larger targets (3 and 2 m) for all materials degrades from <10% when using the HySpex spectral library to around 10% when using the SVC library.
The degradation is much more severe for the smallest 0.25m targets, while the results are comparable for the mid-size targets (0.5 and 1 m). Overall, we conclude that the abundance estimation results for resolved targets are robust, which is consistent with the qualitative similarity of the two spectral libraries.
Abundance estimation for sub-pixel targets is sensitive to the spectral library used and better results are obtained when the library is selected directly from the image.

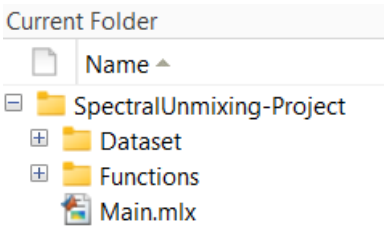
6.6 Future Development

There are other problems that can be investigated in detail with the DLR HySU benchmark dataset, such as:

- **Joint Endmember Extraction and Abundance Estimation**: the team tested the endmember extraction and abundance estimation steps separately, instead a more difficult problem could be to joint this to steps in one single process.
- **Hidden Target Detection**: as we said in the project introduction chapter, in the dataset were included the main target for the spectral unmixing, but also three small hidden targets, namely a 0.25m square red metal target, a 0.5m square blue fabric target and a 0.5m square green fabric target. These three targets were placed across the test field over different backgrounds to test anomaly detection. So, the spectral unmixing can be applied to estimate abundances and therefore pinpoint the desired material, by having a different approach depending on whether the number, spectra and/or size of targets is given as input.

7 Project Overview – Guide to the code

Our MATLAB-based implementation of Spectral Unmixing follows a modular architecture: there's a principle *main* script and separately we provide a *Dataset* folder and a *Functions* folder that groups all the individual live script functions run by the main. This leads decomposing in sorted blocks the structure of the algorithm, aiming a better readability and leading a more efficient development of the code itself, also reducing computational cost and increasing speed because it saves in memory only variables that are strictly essential for the method and not all of them used in functions. Initially we are going to discuss the benchmark dataset provided from DLR, then focus on each individual live scripts explaining their contribute to the whole process and finally analyze the obtained results.



7.1 Prerequisites: MATLAB Add-ons

To run our code is mandatory installing additional packages and toolboxes from the MATLAB Add-on Manager:

- **"Image Processing Toolbox Hyperspectral Imaging Library"**
  Provide functions and tools for hyperspectral image processing and visualization: read, write, and process hyperspectral data captured by using the hyperspectral imaging sensors in a variety of file formats like national imagery transmission format (**NITF**), environment for visualizing images (**ENVI**), tagged image file format (**TIFF**), and metadata text extension (**MTL**) file formats.
  This library presents a set of algorithms for **endmember extraction**, **abundance map estimation, dimensionality reduction**, **band selection**, **spectral matching**, and **anomaly detection**. (Hyperspectral Image Processing, s.d.)

- **"Mapping Toolbox"**
  Provides algorithms and functions for transforming geographic data and creating map displays such as trimming, interpolation, resampling, coordinate transformations, and other techniques.

- **"read_envihdr"**
  Lead to reads the ASCII ENVI-generated image header file and returns all the information in a structure of parameters:
  - samples - number of samples in the image (columns).
  - lines - number of lines in the image (rows).
  - bands - number of bands in the image […] (Tuszynski, 2023)

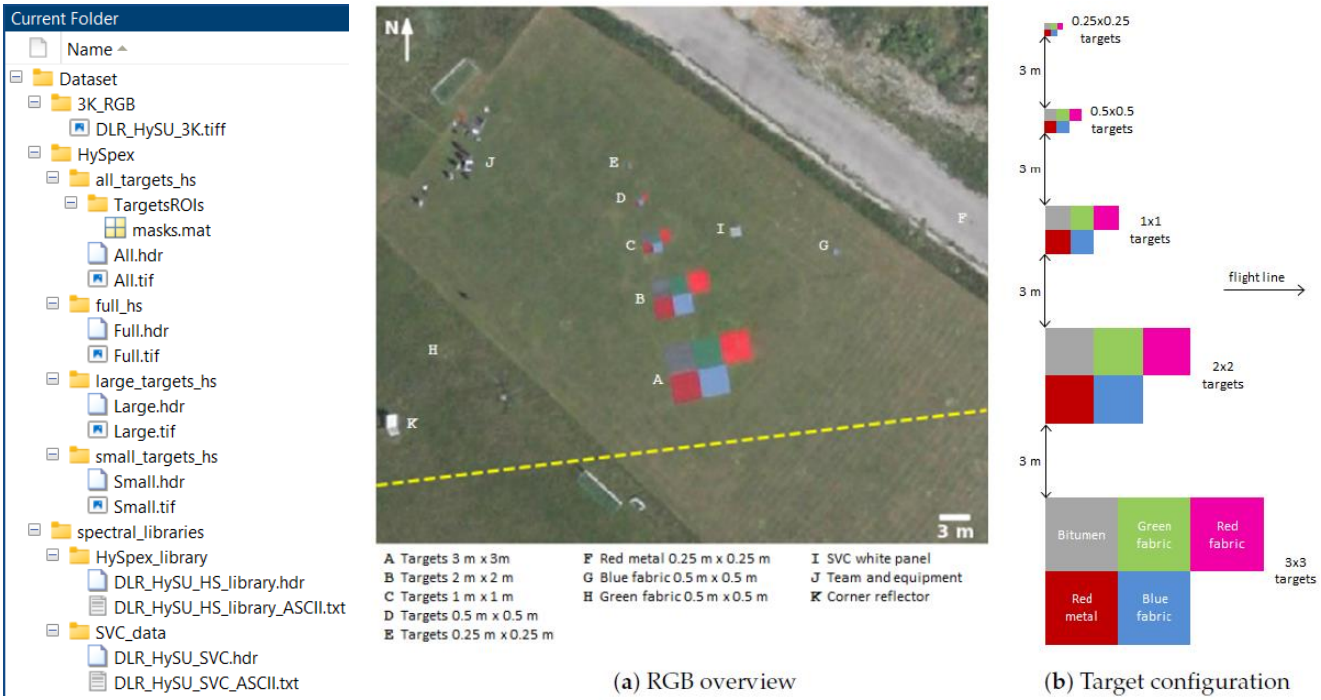
7.2 Dataset

The directory of the provided *Dataset* folder is organized in three subfolders as follows:

1) **3K_RGB**
   This folder contains the high resolution geo-referenced RGB image in the format .tiff.

2) **HySpex**
   Contains the hyperspectral dataset split in four subfolders representing different versions of the same acquisition in .tiff format with the associated header file .hdr containing wavelengths and other metadata:
   a. **all_targets** – provides the subset with all targets including ROIs with locations of the target grouped by size and ROIs grouped by material type.
   b. **full** – provides the full subset containing targets of interest.
   c. **large_targets** – only includes targets of dimension [3m x 3m].
   d. **small_target**s – only includes targets of dimensions [0.5 m x 0.5 m] and [0.25 m x 0.25 m].

3) **spectral_library**
   Contains the spectra collected from the image acquired by HySpex and from in-situ measurements by the SVC in file format .txt and the associated .hdr.

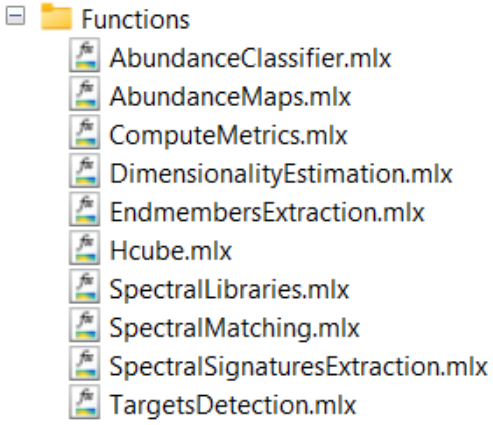


(a) RGB overview (b) Target configuration

7.3 Main

The sorted live script functions in the format "*.mlx"* followed in the *main* process are now listed and explained. **Functions** folder contains:

1) **TargetsDetection**
2) **SpectralLibraries**
3) **SpectralSignaturesExtracti on**
4) **Hcube**
5) **DimensionalityEstimation**
6) **EndmembersExtraction**
7) **ComputeMetrics**
8) **SpectralMatching**
9) **AbundanceMaps**
10) **AbundanceClassifier**



In the *main* script we simply call them, when necessary, with the respective attributes given as input of the functions. Many of them are independent but some there are few functions that depend on the outputs provided by others, so we suggest using the given order of this list.

7.4 Functions

7.4.1 TargetsDetection

```
TargetsDetection()
```

Function Purpose
The script analyses hyperspectral data for targets detection and segmentation and visualizes endmembers grouped by size and grouped by typology exploiting two types of masks contained in *masks.mat* provided from DLR: *materials* and *targetsROI,* both with same spatial dimensions as *Full* subset: 86x123 pixels.
For each endmembers outputs the original RGB image, the segmented one and the overlayed highlighted region of interest.
"masks.mat" contains:

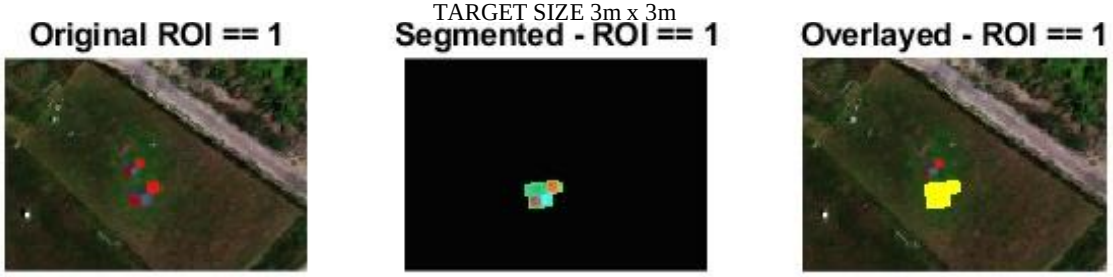
- ***materials*** (86x123x5) – for each of the five endmembers (Bitumen, Red Metal Sheets, Blue Fabric, Red Fabric and Green Fabric) this binary mask provides the location of the individual material.
  - *1*: pixel contains the element
  - *0*: pixel doesn't contain endmember.

- ***targetsROI*** (86x123) – this mask provides the location of the endmembers grouped by size:
  - *0*: no endmember is detected;
  - *1*: represent 3m x 3m endmembers;
  - *2*: represents 2m x 2m endmembers;
  - *3*: represents 1m x 1m endmembers;
  - *4*: represents 0.5m x 0.5m endmembers;
  - *5*: represents 0.25m x 0.25m endmembers.

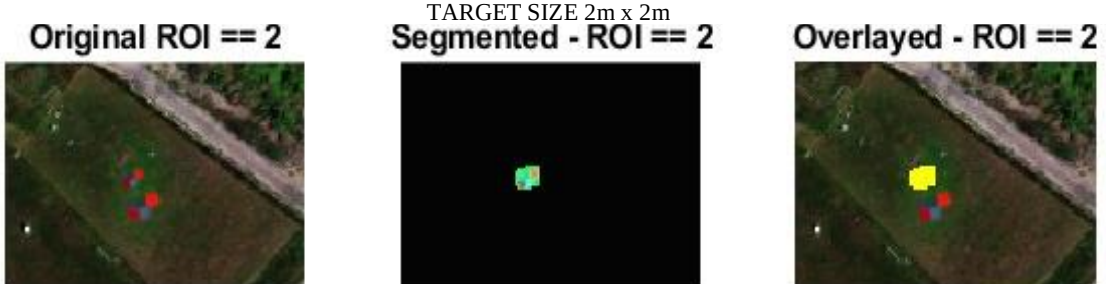
We used both mask in this way: create the datacube of the original image to plot its RGB version, create an empty matrix with the same dimension of the datacube that represents the hyperspectral *Full* image and iterate over the 3rd dimension in the case of materials and over the numbers of sizes in case of targetsROI filling with 1 in the same pixel position if the masks contain an integer (1 in the first case, 1 to 5 in the last one).
For the segmented ROI we isolate values contained in the original datacube that corresponds to positions detected with integer values in the masks creating a new datacube and visualizing it, for the overlayed ROI we overlay the original image with these obtained positions. Obtained figures are shown below.

7.4.1.1 Results using *taregetsROI* mask (86x123) on *Full.tif* subset

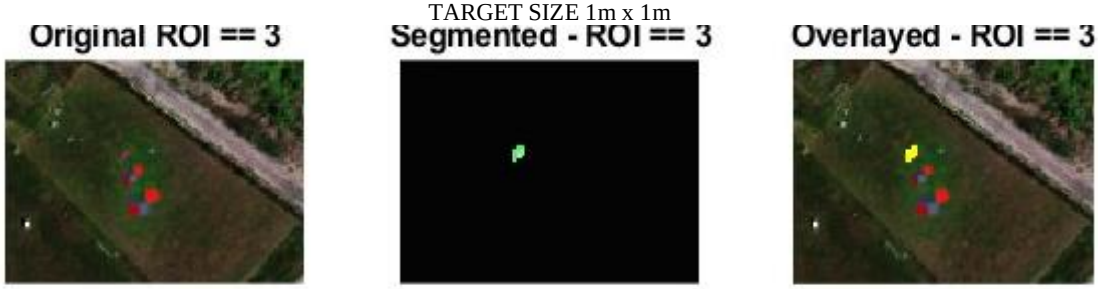
TARGET SIZE 3m x 3m



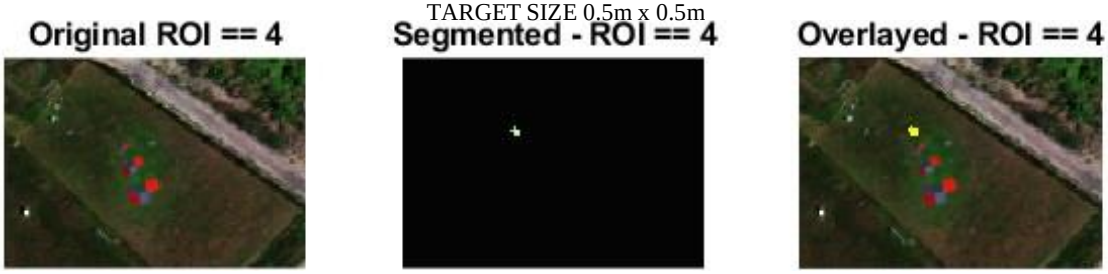
TARGET SIZE 2m x 2m



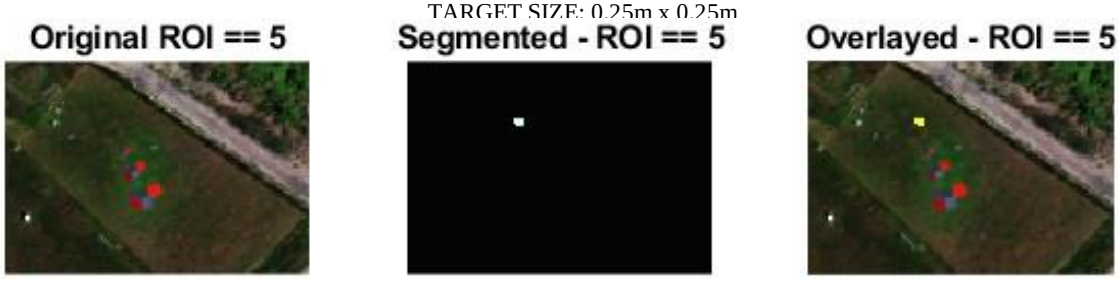
TARGET SIZE 1m x 1m



TARGET SIZE 0.5m x 0.5m

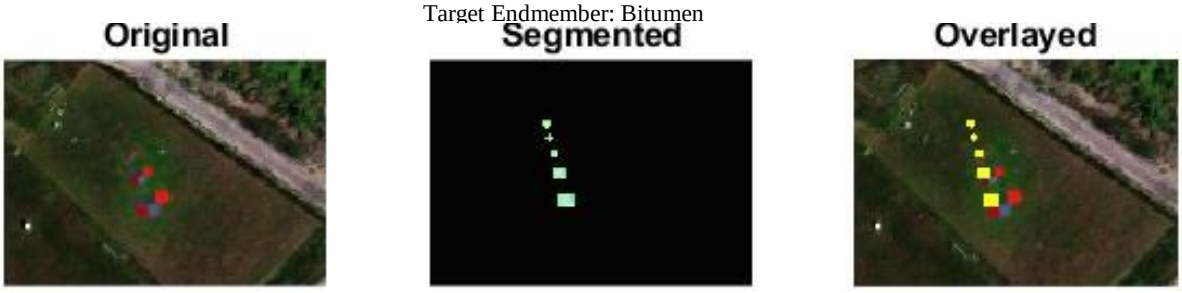


TARGET SIZE: 0.25m x 0.25m

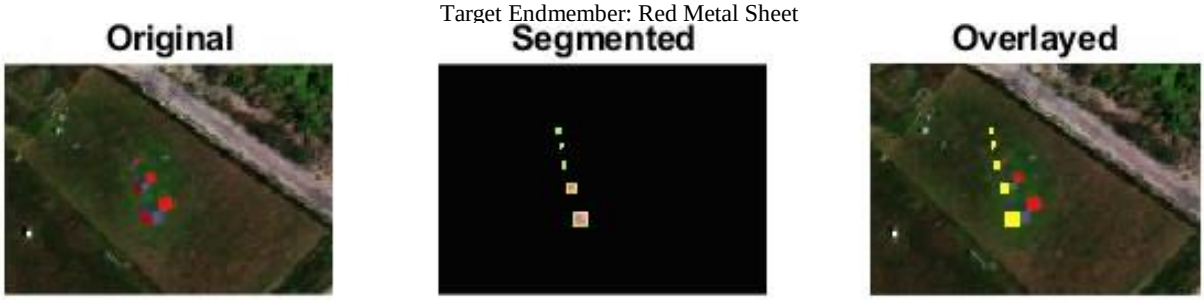


7.4.1.2 Results using *materials* mask (86x123x5) on *Full.tif* subset

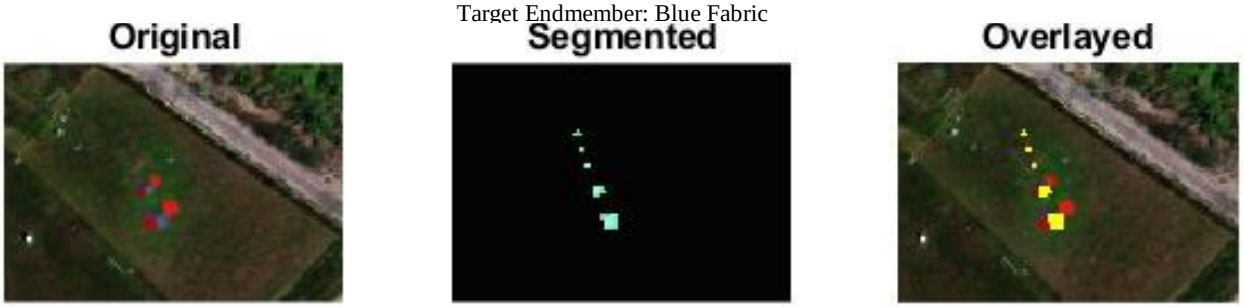
Target Endmember: Bitumen



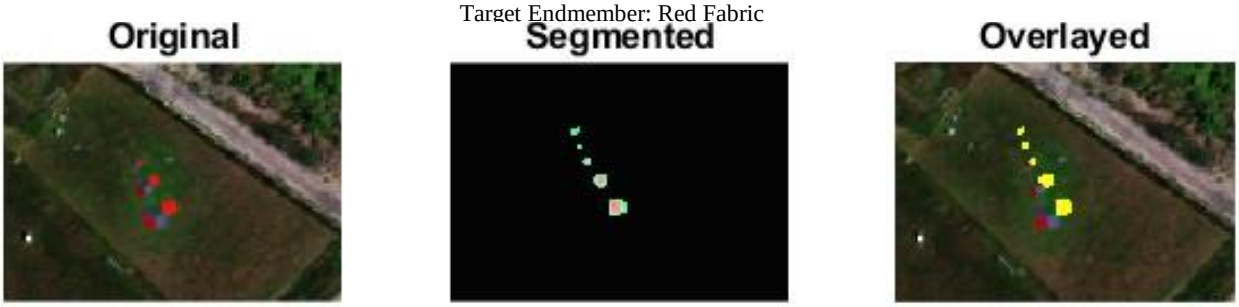
Target Endmember: Red Metal Sheet



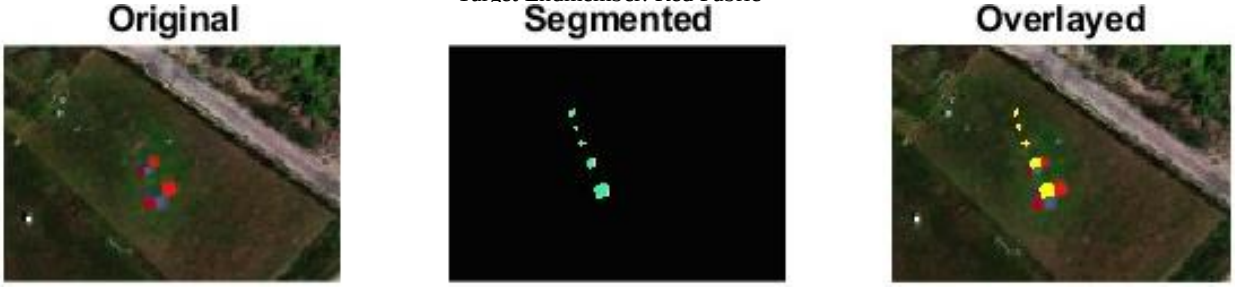
Target Endmember: Blue Fabric



Target Endmember: Red Fabric



Target Endmember: Red Fabric



7.4.2 SpectralLibraries

```
[y_hs, y_svc, colors, spectraNames] = SpectralLibraries(spectral_hdr, ascii_hs, ascii_svc)
```

Function Purpose
This function aims extracting crucial information contained in the two proposed spectral libraries of DLR relative to the two different acquisitions of the different sensors: HySpex and SVC. The information is given in *DLR_HySU_HS_library_ASCII.txt* and *DLR_HySU_SVC_ASCII.txt*.

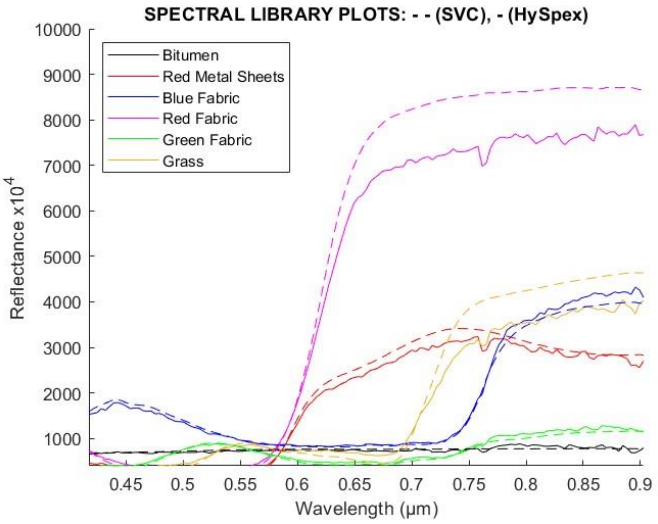
Input

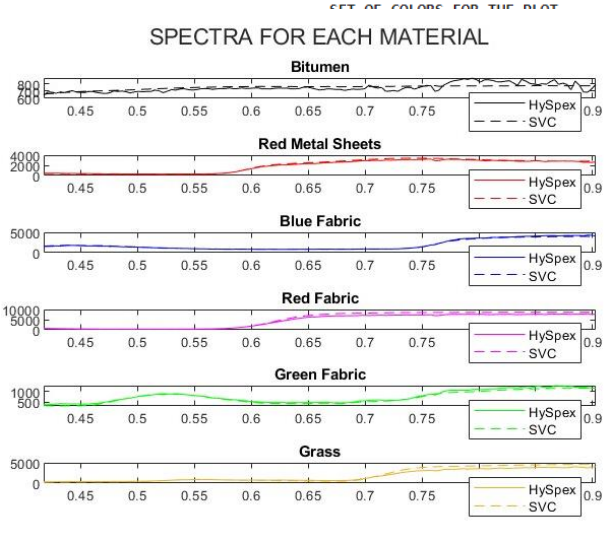
- **spectral_hdr** – name or path of an ENVI header file .hdr containing metadata.
- **ascii_hs** – name or path of ASCII file for HySpex data .txt containing spectra.
- **ascii_svc** – name or path of an ASCII file for SVC data .txt containing spectra.

Output

- **y_hs (135x6) –** reflectance values collected from HySpex sensor in 135 bands for each endmember plus the Grass.
- **y_svc (135x6)** – reflectance values collected from SVC spectrometer in 135 bands for each endmember plus the Grass.

Both spectra are plotted together on the same horizontal axis representing the number of wavelengths used in the reflectance acquisition of the endmembers: 135 bands (from 417.4nm to 902.79nm). This function also output each individual spectrum separate.





- **colors (6x3)** – our manually defined set of colours associated to each endmember spectra that follows the set of colors shown in DLR research paper. This helps comparing our results with the theoretical ones.

- **spectraNames (6x1)** – cell with extracted names of the endmembers' spectra acquired from DLR libraries (in order: Bitumen, Red Metal Sheet, Blue Fabric, Red

Column of sig	Endmember Class Name
1	'Bitumen'
2	'Red Metal Sheets'
3	'Blue Fabric'
4	'Red Fabric'
5	'Green Fabric'
6	'Grass'

7.4.3 SpectralSignaturesExtraction

```
spectral_library = SpectralSignaturesExtraction(file_tif, file_hdr, spectraNames, colors)
```

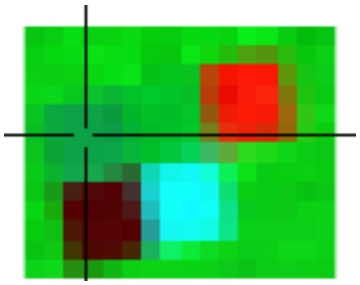
Function Purpose
According to theoretical reasons, to extract spectra correctly we must acquire them in a pure pixel condition, so when we may retain material spectra is not interfered with another pixel's signature. It is mandatory to use image with high spatial resolution or compute them at the best location where you can consider that true: at the centre of the material for each of them. This function leaves the user free to point it with cursor on the RGB image, for each of the wanted endmembers, and then extract the value of reflectance for each of the 135 bands. We collect then all in a single matrix with at the first column specifying wavelength expressed in μm, same matrix as the equipped one by DLR. Finally, the manually extracted signatures are plotted in one graph and overlayed with the DLR ones to test if our application of pure pixel extraction is truthful. To be able to match for each endmember both spectra (from DLR and ours) is mandatory to set same individual colors we used before.

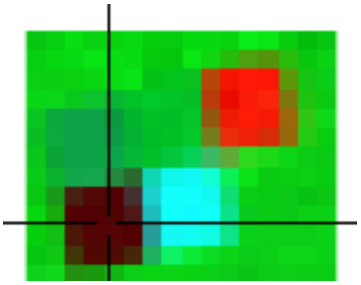
Input

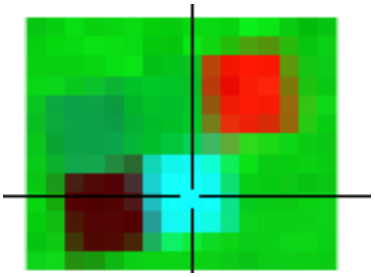
- **file_tif** – name or path of a .tiff file containing the acquired image you want to extract spectra from.
- **file_hdr** – name or path of the associated header file.
- **spectraNames (6x1)** – spectra name of each material output from **SpectraLibraries**.
- **colors (6x3)** – DLR set of colours for endmember spectra extracted from **SpectraLibraries**.

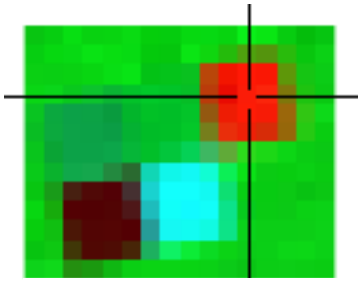
Output

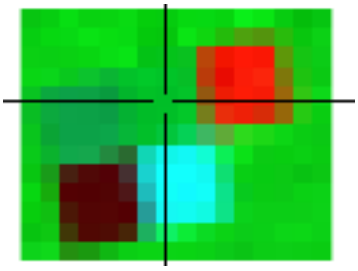
- **spectral_library** (**135x7**) – matrix of reflectance for each endmember + Grass computed for 135 bands. First column represents the wavelengths.

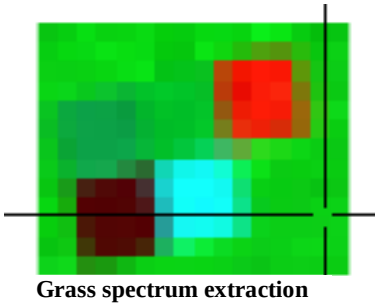

Bitumen spectrum extraction


Red Metal Sheet spectrum extraction

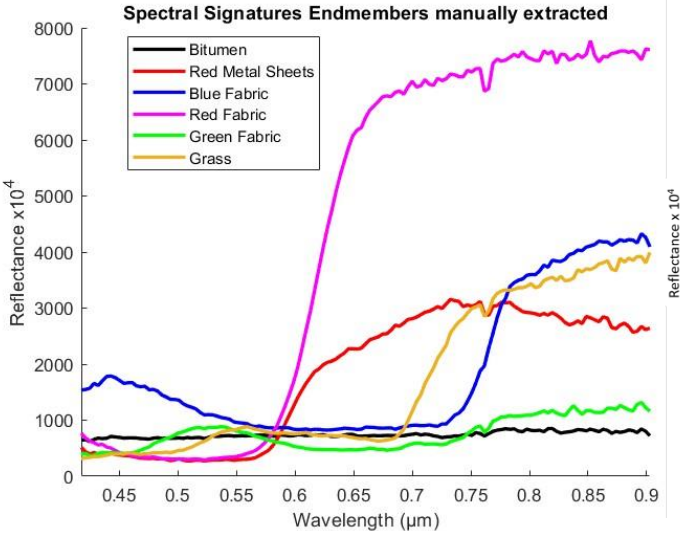

Blue Fabric spectrum extraction


Red Fabric spectrum extraction

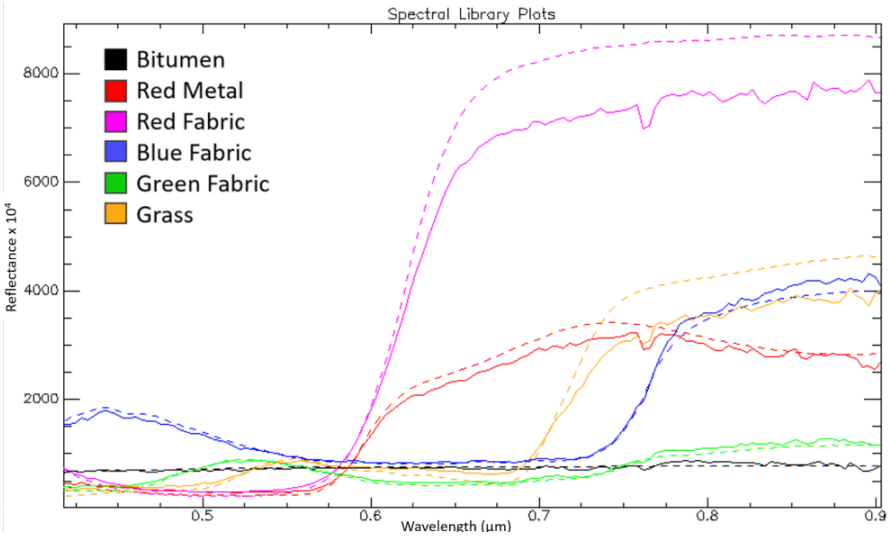

Green Fabric spectrum extraction


Grass spectrum extraction

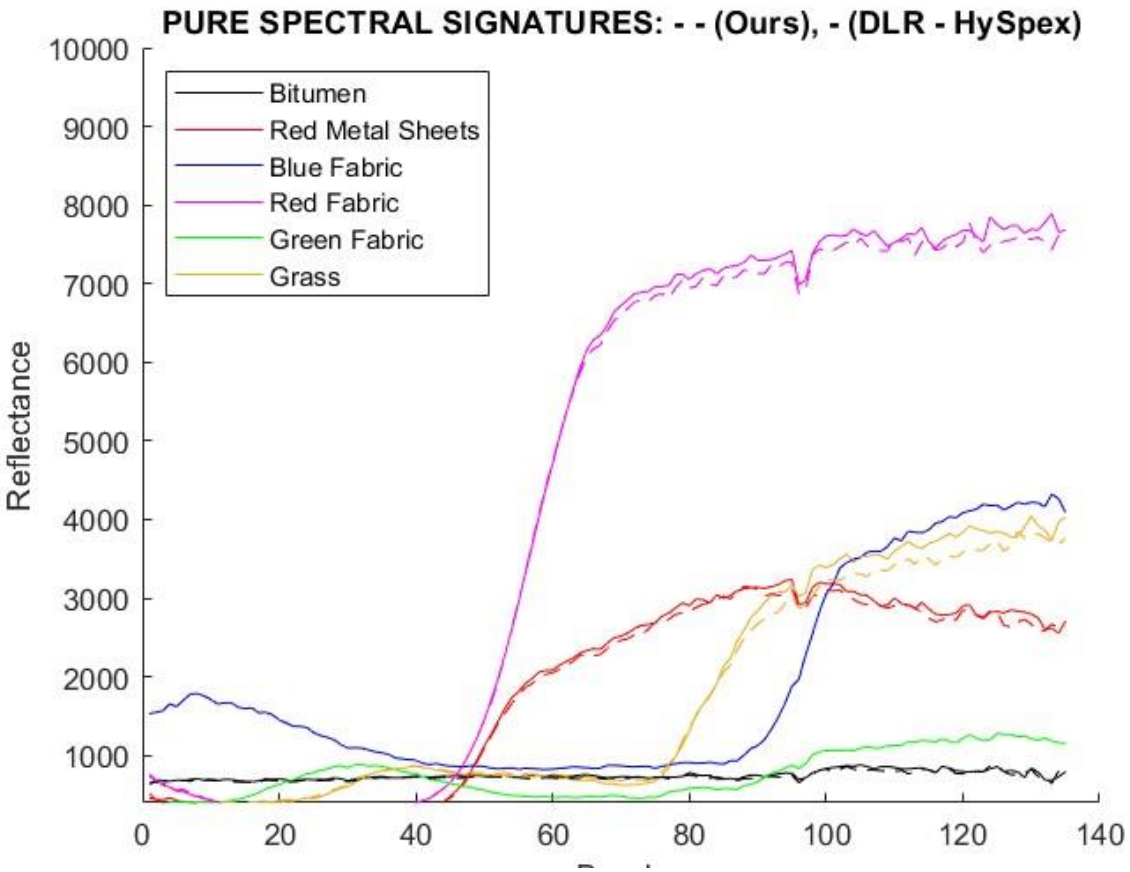
As we can see from our result, the manually extracted signatures plot is strongly comparable to the DLR one. Pure pixel condition is respected.



Spectra Signatures manually extracted from SpectralSignaturesExtraction function.

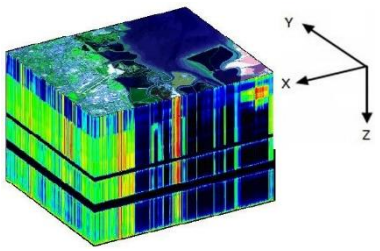


Spectra Signatures image taken from DLR paper (SVC -- & HySpex -



Pure pixel spectra Signatures comparison between our extracted spectra and the DLR one.

7.4.4 Hcube




```
[hcube_cell, hcube_full, hcube_all, hcube_large, hcube_small] = Hcube(file_names)
```

Function Purpose
This function allows creating for each of the imagery cases scenarios the hypercube needed for further processing and analysis. The hypercube allows us to store and organize a vast amount of spectral and spatial information and here's a breakdown of the components of a hyperspectral hypercube:

- **Spatial Dimensions (m x n)** – two spatial dimensions. These dimensions represent the X and Y coordinates in the image or scene being captured and indicate where in the scene each spectral measurement was taken.

- **Spectral Dimension p** – represents the different wavelengths or spectral bands at which data was collected. Each spectral band contains information about how the scene appears at a specific wavelength. The spectral dimension is often visualized as the "third" dimension in the hypercube.

- **Radiometric Values** – : At each spatial and spectral coordinate in the hypercube, a radiometric value is recorded. This value represents the intensity or brightness of the reflected or emitted light at that specific wavelength and location.

Input

- **file_names** – a cell containing names of the .tiff and associated .hdr (same name was given) such as `file_names = {'Full', 'All', 'Large', 'Small'};`

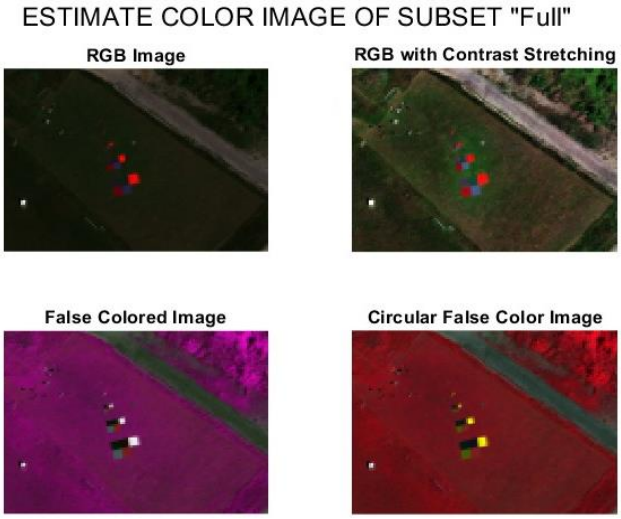
Output

- **hcube_cell (1x4)** – cell containing the hypercube computed for each imagery
- **hcube_full (1x1) –** hypercube for the subset *Full*
- **hcube_all (1x1)** – hypercube for the subset *All*
- **hcube_large (1x1)** – hypercube for the subset *Large*
- **hcube_small (1x1)** – hypercube for the subset *Small*

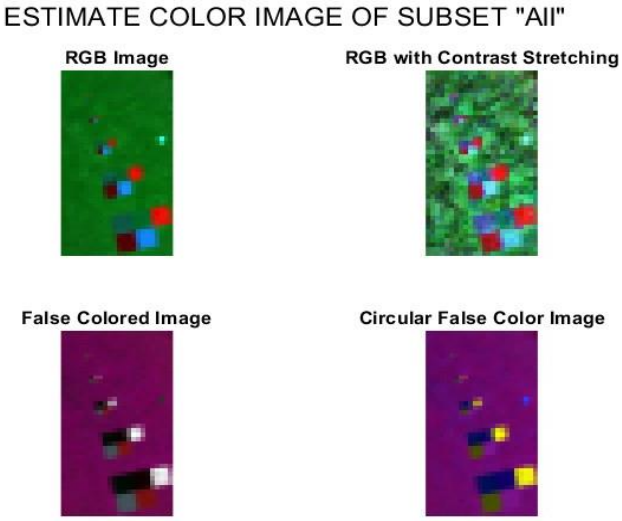
  For each of the created datacube are plotted 4 images using the function `colorize(hcube_cell{i}, 'Method', 'indicatethemethodtouse');`

  - `'indicatethemethodtouse'`= **RGB Image**
  - `'indicatethemethodtouse'`= **RGB Contrast Stretching Image**
  - `'indicatethemethodtouse'`= **False Colored Image**
  - `'indicatethemethodtouse'`= **Color Infrared Image (CIR)**

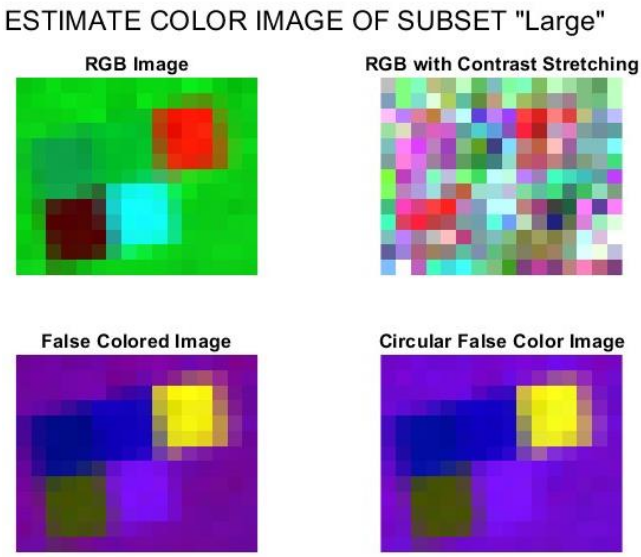
(MathWorks, s.d.)



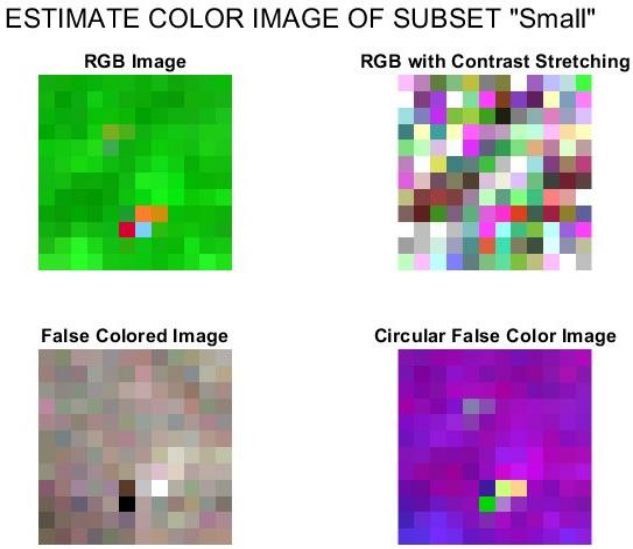
Imagery for the subset *Full*



Imagery for the subset *All*



Imagery for the subset *Large*



Imagery for the subset *Small*

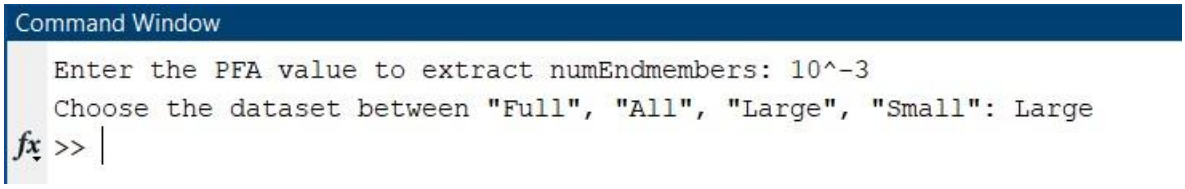
7.4.5 DimensionalityEstimation

```
resultsTable = DimensionalityEstimation(files_tif, files_hdr)
```

Function Purpose
This function is designed for dimensionality estimation from hyperspectral data using various Probability of False Alarm (**PFA**) values and multiple input datasets. It iterates through different PFA values: `PFA_values = [10^0, 10^-1, 10^-2, 10^-3, 10^-4, 10^-5, 10^-6];  % PFA values to iterate over` and different datasets (referred to as 'Full,' 'All,' 'Large,' and 'Small') to estimate the number of endmembers in each dataset under different noise conditions using the countEndmembersHFC function.
Then, it stores the results in the numEndmembers array and results are displayed in the form of a table, showing the number of endmembers for each dataset and PFA value. Finally, the user is prompted to select a specific PFA value and dataset referring to the previous table.

Input

- **files_tif** = – name or path of a .tiff, or multiple files, containing the acquired image you want to extract spectra from. (`files_tif = {'Full.tif', 'All.tif', 'Large.tif', 'Small.tif'};`)
- **files_hdr** = name or path of a .hdr, or multiple files, containing the associated metadata.(`files_hdr = {'Full.hdr', 'All.hdr', 'Large.hdr', 'Small.hdr'};`)
- Users must manually select which PFA and which dataset the function has to extract from the number of endmembers:



Output

- **resultsTable** – number of endmembers computed for each of the dataset with different PFA values.

	PFA	Full	All	Large	Small
1	1	135	135	135	135
2	0.1000	8	7	7	5
3	0.0100	6	6	6	2
4	0.0010	5	5	5	2
5	0.0001	4	5	4	2
6	0	4	5	4	2
7	0	4	4	4	2

We finally save in the workspace `numEndmembers for PFA=0.00100 and Large:5`, since in the following function we need this variable to extract the spectral signatures.
N.B. PFA is a statistical metric used as a threshold in signal processing and detection theory that represents the probability that a detector or algorithm will produce a false positive or false alarm when detecting a particular signal or target. The algorithm sets a threshold based on a given PFA level, and any statistic exceeding this threshold is considered as an endmember. By adjusting the PFA threshold, you can control the trade-off between correctly identifying endmembers (true positives) and producing false alarms. Standard value is 10^{-3}. Without pure pixel condition smaller targets correspond to highly mixed pixels as expected. (Cerra, et al., 2021)

7.4.6 EndmembersExtraction

```
endmembers_cell = EndmembersExtraction(hcube_cell, numEndmembers_extracted, file_names, spectraNames,colors)
```

Function Purpose

Function performing endmembers extraction from the previously computed hyperspectral data cubes using two different algorithms:

- **NFINDR** (Non-Negative Matrix Factorization Based on Determinant Ratio)
- **PPI** (Pixel Purity Index).

The code also provides a visualization of the endmembers and a comparison of the extraction algorithms. The purpose is to allow users to compare the results of two endmember extraction algorithms and visualize the endmember spectra in relation with the manually extracted one by `SpectralSignaturesExtraction`. The visualization provides insight into the quality and number of endmembers extracted, helping users choose the most suitable algorithm for their hyperspectral data analysis. (MathWorks, s.d.) Here's an explanation of the code:

Input

- **hcube_cell** – cell that contains the hcube for each dataset output of the `Hcube` function.
- **numEndmembers_extracted** – number of endmembers extracted by the `DimensionalityEstimation` function.
- **file_names** – data previously set for `Hcube` function.
- **spectraNames** – names of endmembers extracted in `SpectralLibraries` function.
- **colors** – set of colors used from DLR.

- Users must manually select which data cube the function has to extract from the NFINDR – PPI signatures, watching the displayed information:

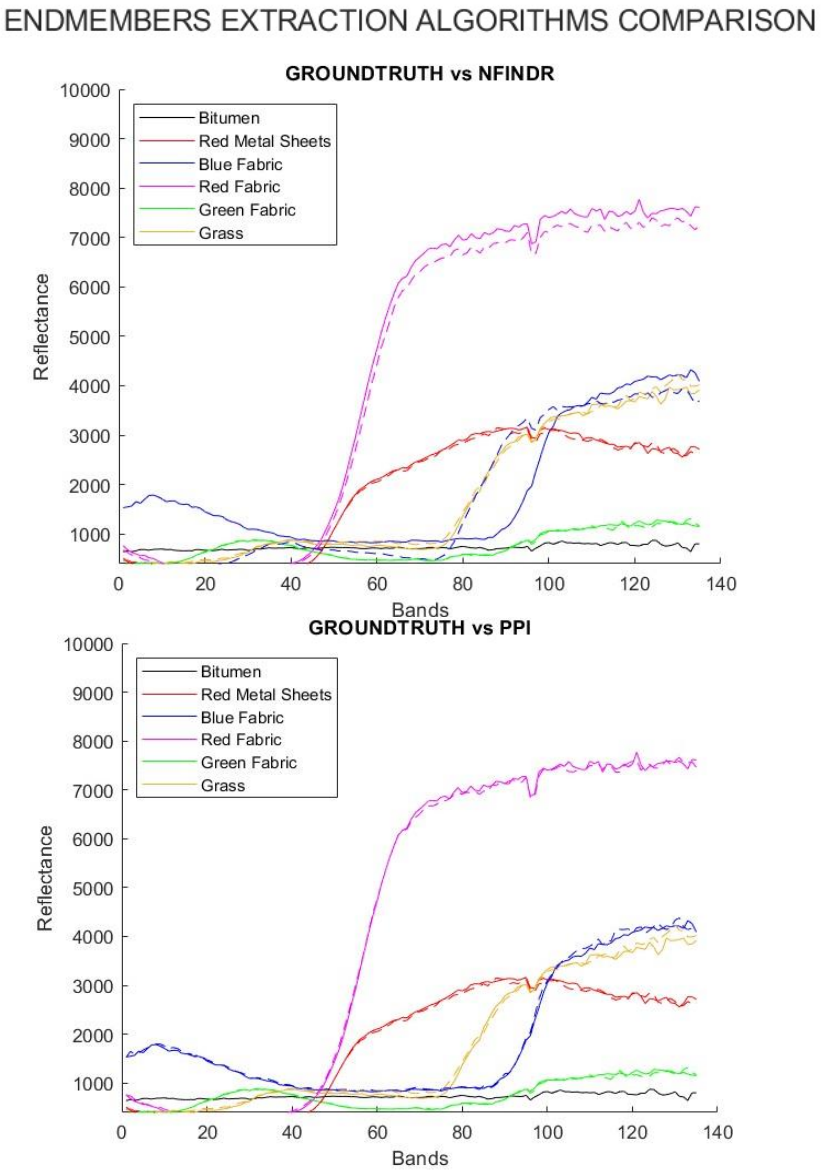
```
Choose the hypercube to use:
1. hcube_cell{1} of Subfolder Full
2. hcube_cell{2} of Subfolder All
3. hcube_cell{3} of Subfolder Large
4. hcube_cell{4} of Subfolder Small

Enter the number of the hypercube to use: 2
>>
```

Ouput

- **endmembers_cell** – cell that stores the extracted endmembers in the endmembers_cell cell array with variable names.

N.B. Since the two different algorithms manage the datacube in different ways, the endmembers signatures computed have different columns' position in the final matrix and, considering that each colour is assigned in a sequential order, we had to relocate each of them to be able to compare them in the final plots. (MathWorks, s.d.)



All algorithms struggle identifying the bitumen material, which is difficult to retrieve in a mixed setting due to the absence of clear absorption features in the analyzed spectral range. As we can see from our outputs, and as we saw from the DLR application tests, not all the targets have robust results. Indeed, they depends on materials and size. For all target sizes bitumen and blue fabric display some degree of sensitivity to the spectral library used.

7.4.7 ComputeMetrics

```
resultTable = ComputeMetrics(signatures, endmembers, spectraNames)
```

Function Purpose

RMSE (Root Mean Square Error) and **RMSPE** (Root Mean Square Percentage Error) are both metrics used to evaluate the accuracy of a model and in this case assess how well the endmembers extraction algorithms results (NFINDR and PPI) align with the actual values presented in the spectral library. The code normalizes both `signatures` and `endmembers` with **Min-Max Normalization**. This step scales the data between 0 and 1.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values
$y_1, y_2, \ldots, y_n$ are observed values
n is the number of observations

Input

- **signatures** – manually extracted endmembers' spectra with pure pixel condition by `SpectralSignaturesExtraction` function. (OSBERVED VALUE y_i)
- **endmembers** – automatic extracted endmembers' spectra by NFINDR or PPI. (PREDICTED VALUES $\hat{y}_i$)
- **spectraNames** - names of endmembers extracted in `SpectralLibraries` function.

Output

- **resultTable** – RMSE and RMPSE computed for each endmember column between the manually extracted spectra and the NFINDR or PPI computed one.

We wanted to see which performance was better: NFINDR or PPI. So, we evaluated this function for both of them:

```
% METRICS ON EXTRACTED SIGNATURES (RMSE - RMSPE)
Metrics_nfindr = ComputeMetrics(signatures, endmembersCell{1}, spectraNames);
Metrics_ppi = ComputeMetrics(signatures, endmembersCell{2}, spectraNames);
```

Metrics_NFINDR (RMSE - RMPSE)

SpectraName	RMSE	RMSPE
{'Bitumen' }	NaN	NaN
{'Red Metal Sheets'}	0.02	3.21
{'Blue Fabric' }	0.22	70.58
{'Red Fabric' }	0.01	2.08
{'Green Fabric' }	0.04	9.65
{'Grass' }	0.04	9.19

Metrics_PPI (RMSE - RMPSE)

SpectraName	RMSE	RMSPE
{'Bitumen' }	NaN	NaN
{'Red Metal Sheets'}	0.02	3.21
{'Blue Fabric' }	0.01	4.67
{'Red Fabric' }	0.01	1.7
{'Green Fabric' }	0.04	9.65
{'Grass' }	0.04	9.19

	NFINDR	**PPI**
PROS	• Low values for Red Metal Sheets, Red Fabric, Green Fabric and Grass	• Lowest error for the other endmembers.
CONS	• Couldn't extract Bitumen signature. • Don't perform well with Blue Fabric either.	• Couldn't extract Bitumen signature.

7.4.8 SpectralMatching

```
SpectralMatching(hcube, spectral_library, spectraNames)
```

Function Purpose
This function performs spectral matching for each endmember in the manually extracted library and displays the results using two similarity measures: Spectral Angle Mapper (**SAM**) and Spectral Information Divergence (**SID**).
The spectral matching method compares the spectral signature of each pixel in the hyperspectral data cube with a reference spectral signature from the custom `SpectralSignaturesExtraction` spectrum output. (MathWorks, s.d.)
Spectral matching method, specified as the comma-separated pair consisting of 'Method' and one of these values: ‘sam’ and ‘sid’.
To compute the distance scores of the spectrum of the hyperspectral data pixels with respect to the reference spectrum we used the external function `spectralMatch`:

```
score = spectralMatch(libData, hcube, "Method", method);
```

- method = **'sam'** – spectral angle mapper (SAM) method, which measures the similarity between two spectra by computing the angular distance between them.

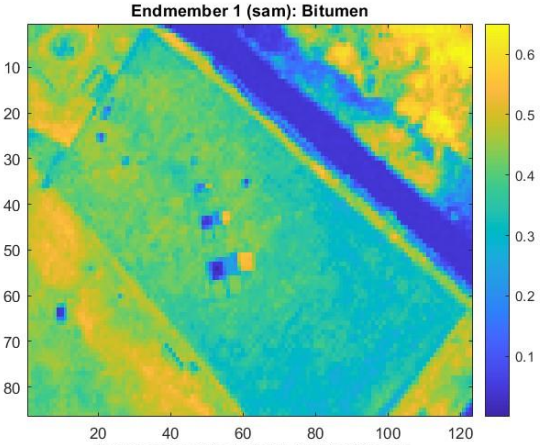
- method = **'sid'** – spectral information divergence (SID) method, which measures the similarity between two spectra by computing the difference between their probability distribution values.

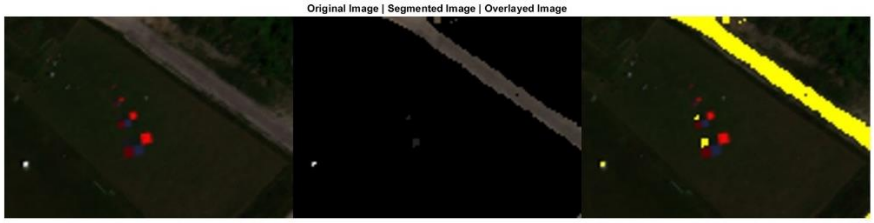
Input
- **hcube** – hcube output of the `Hcube` function you want to use to compare spectra.
- **spectral_library (135x7)** – matrix of reflectance for each endmember + Grass computed for 135 bands output of `SpectralSignaturesExtraction` function.
- **spectraNames** – names of endmembers extracted in `SpectralLibraries` function.

Output
We then display for each method and each endmember:

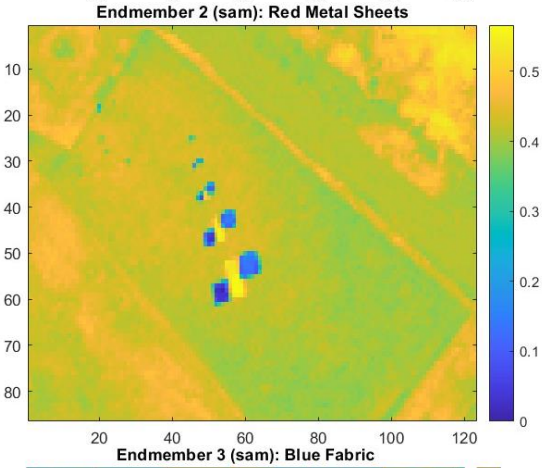
- **Distance scores** – pixels with low distance scores are stronger matches to the reference spectrum and are more likely to belong to the endmember region.
- **RGB**
- **Segmented Image** – contains only the regions that are segmented from the original data cube.
- **Overlayed Image**

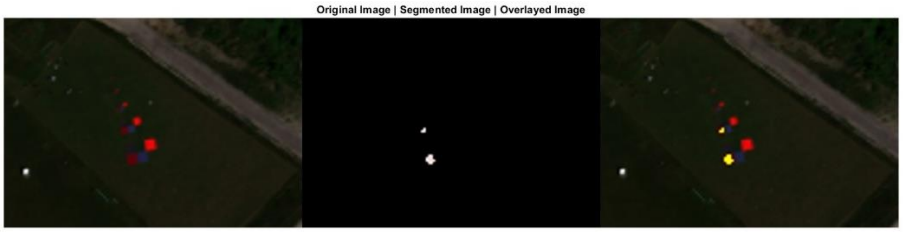
7.4.8.1 SAM (‘*Full.tif*’ Subset) experimental results analysis



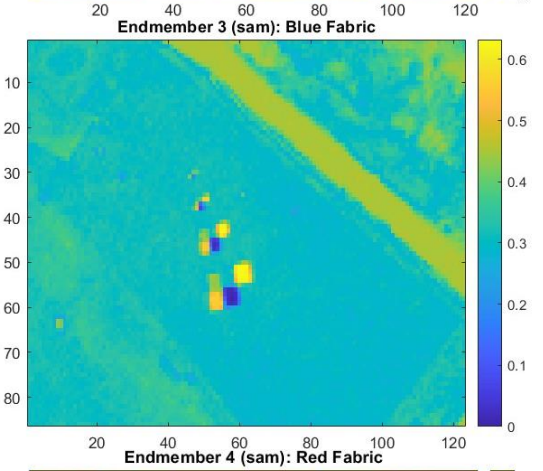


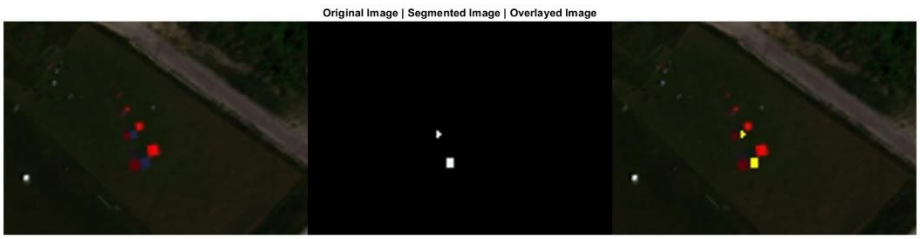
Threshold = 0.05



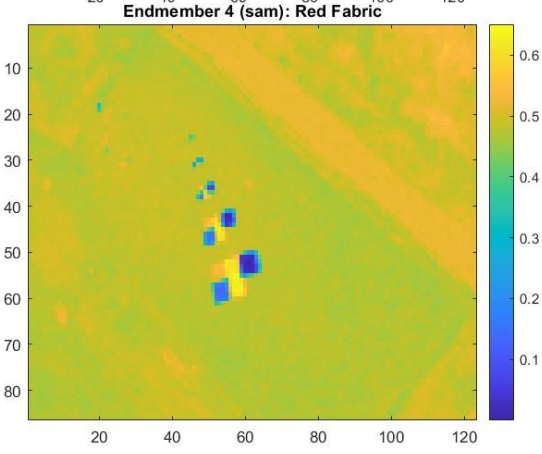


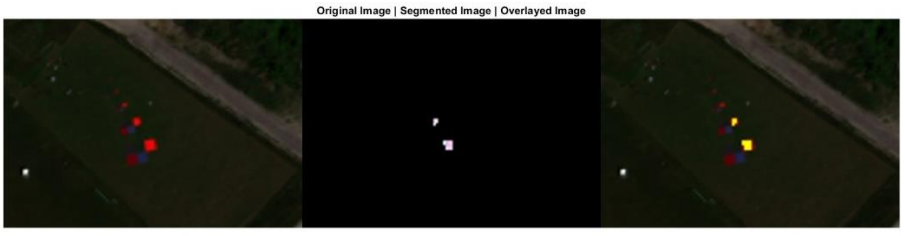
Threshold = 0.05



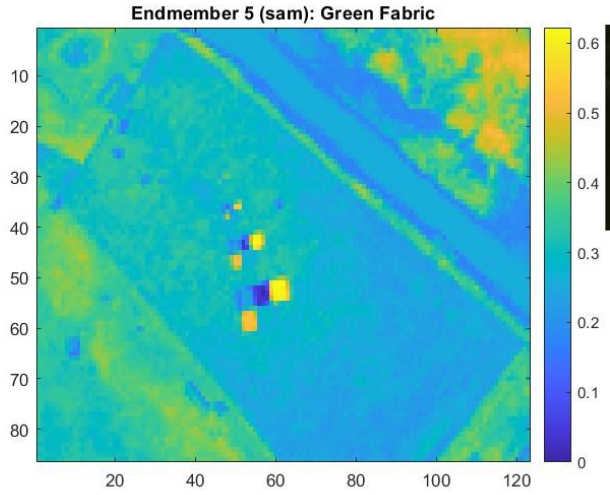


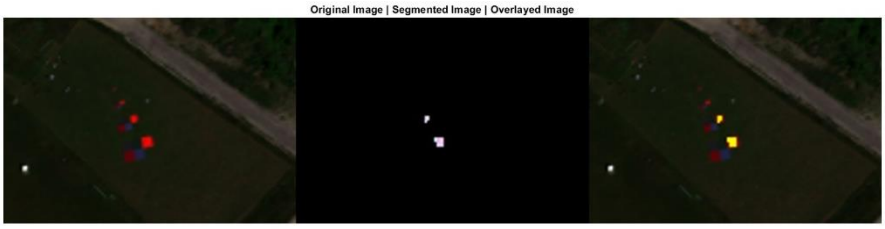
Threshold = 0.05



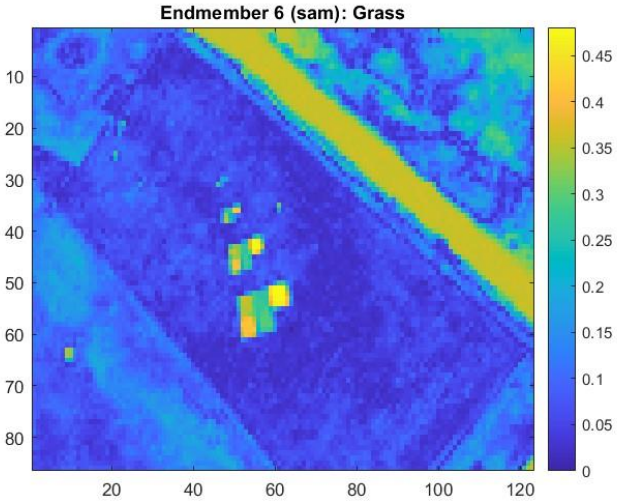


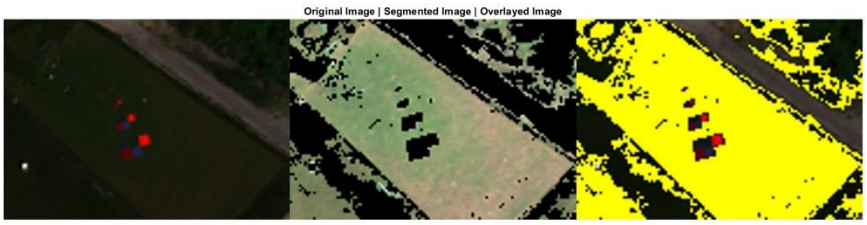
Threhsold = 0.05





Threshold = 0.05



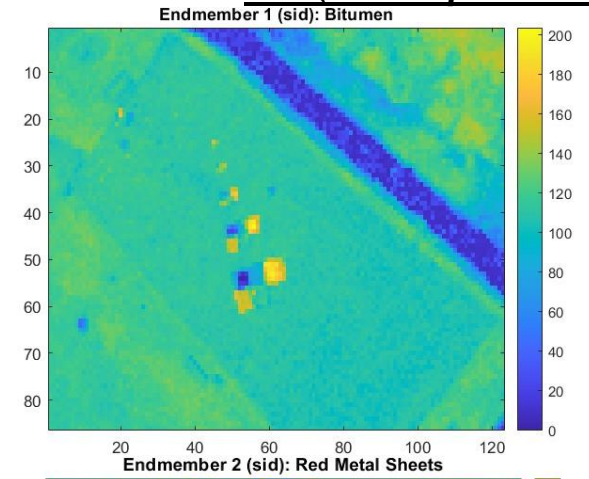


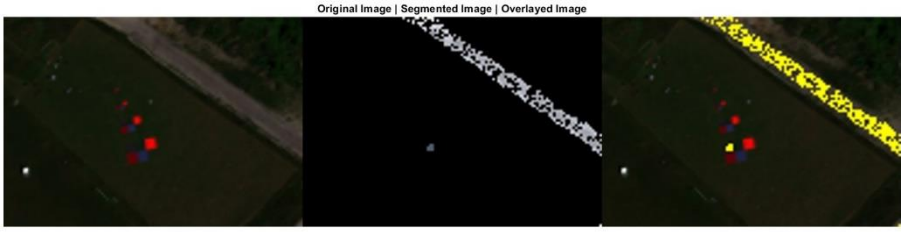
Threshold = 0.1

SAM on *Full.tif*

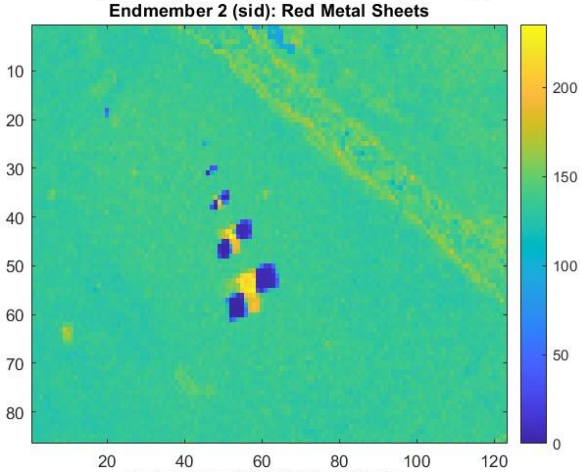
- Whit these two threshold values, 0.05 for the endmembers and 0.1 for Grass, SAM correctly match the 3mx3m and 2mx2m objects but cannot perform well with smaller ones. If we try with higher values, it can also match them but due to the low spatial resolution of this subset image, that causes highly mixed pixels, SAM will consider pixels that don't belong to the specific endmember as part of it. We've preferred to have all the matches to be correct instead of having all the endmembers matched but with a lot of noise and wrong more matches.
- From the images it appears clear that some materials have a signature that has an angular distance with the surrounding ones more marked. This is especially true for Red Metal Sheets and Red Fabric.

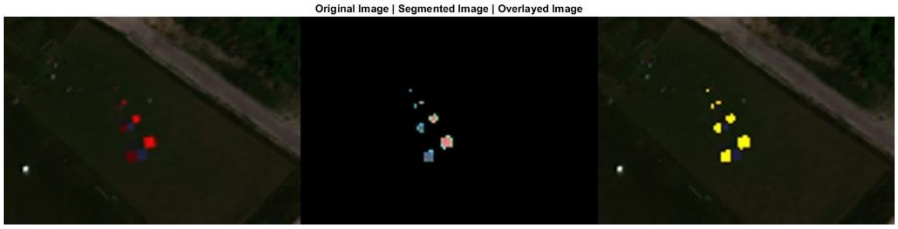
7.4.8.2 SID (‘*Full .tif*’ Subset) experimental results analysis



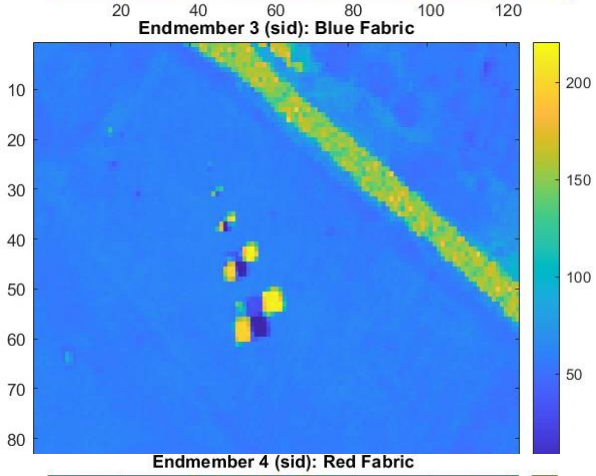


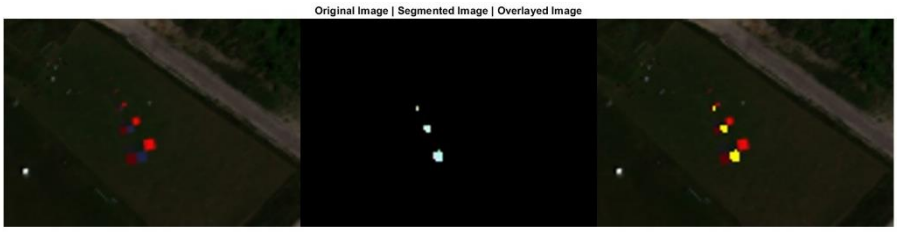
Threshold = 20



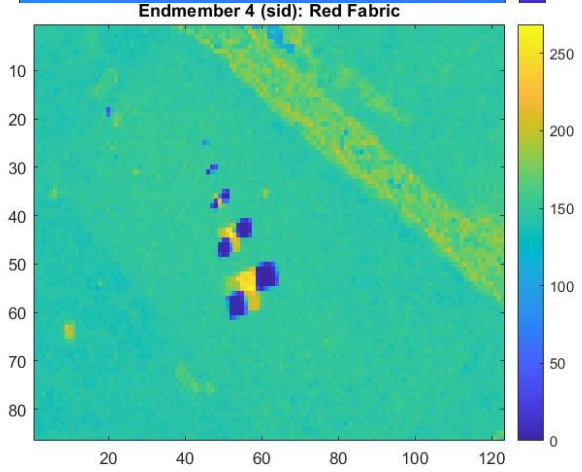


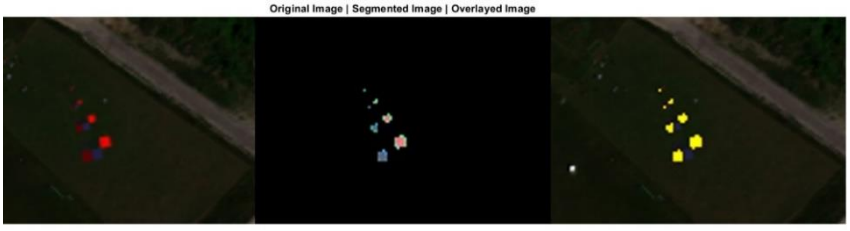
Threshold = 20



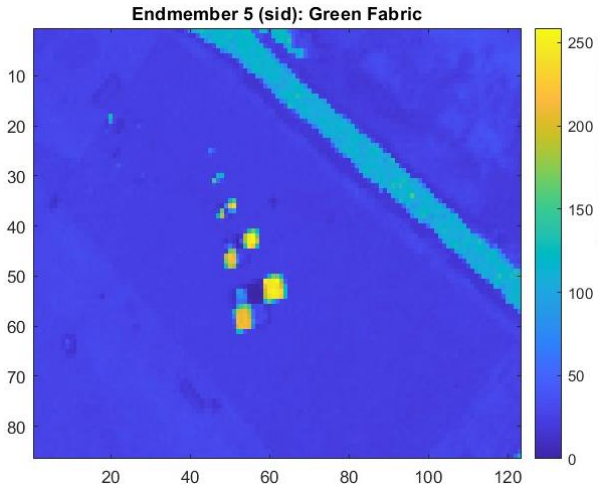


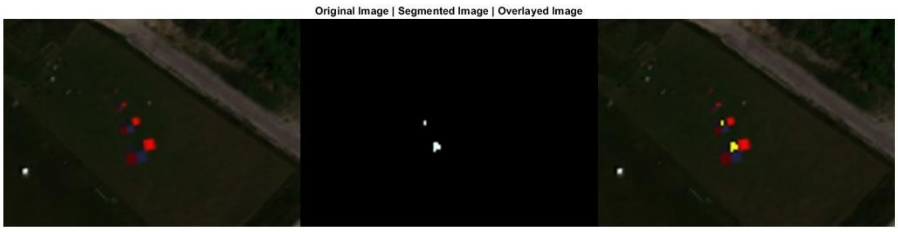
Threshold = 20



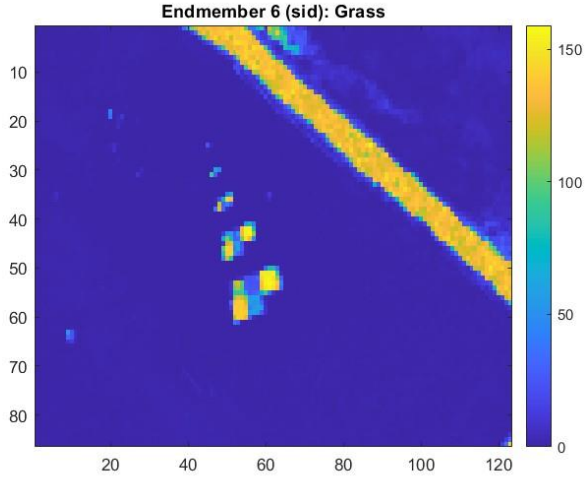


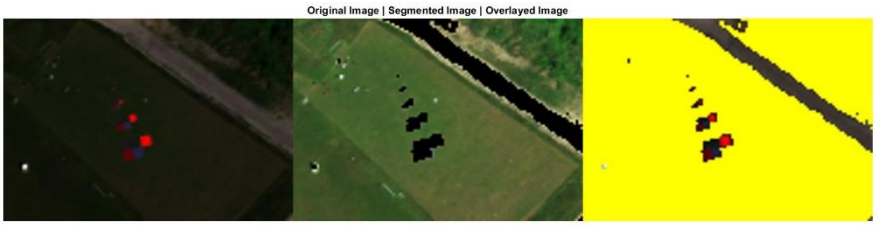
Threshold = 20





Threshold = 8

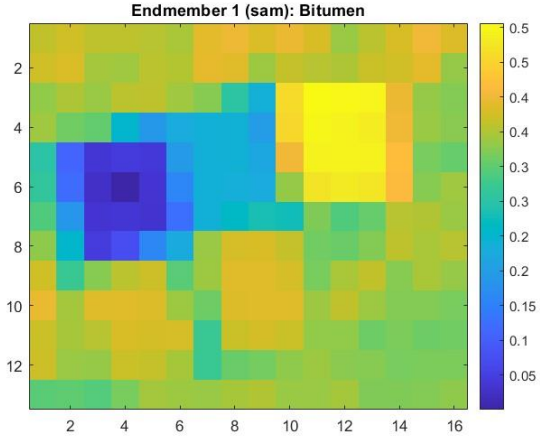


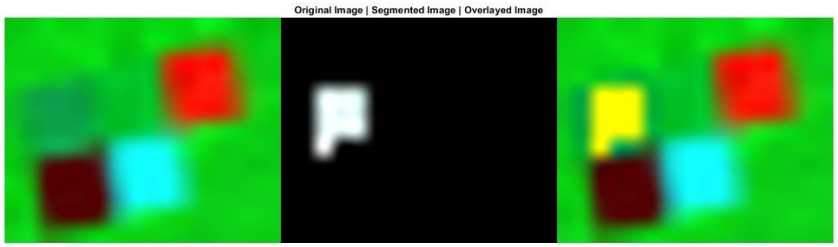


Threshold = 20

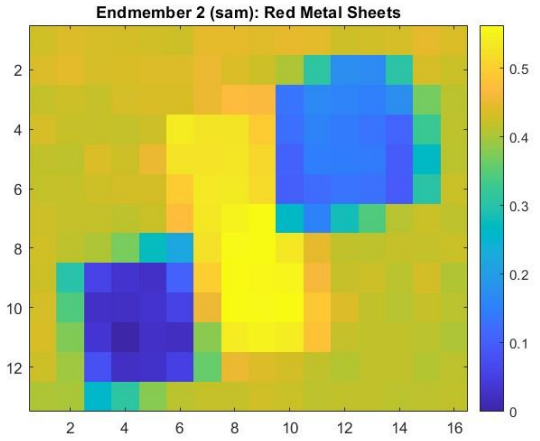
SID on *Full.tif*
• Red Fabric and Red Metal Sheets are the worst to be matched because they may share a similar probability distribution of the spectra so SID, by computing the difference between them, return low values and could not recognize them as two different objects. • Same with Green Fabric and its surrounding, in fact we had to lower the threshold from 20 to 8.

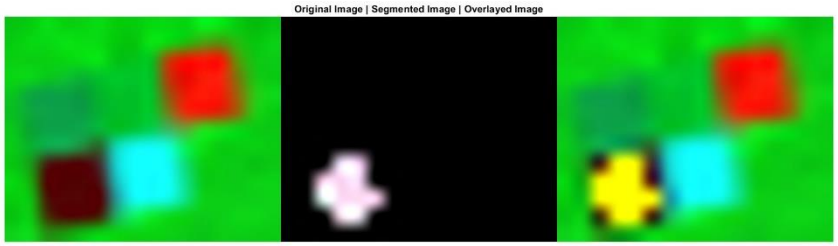
7.4.8.3 SAM (‘*Large.tif*’ Subset) experimental results analysis



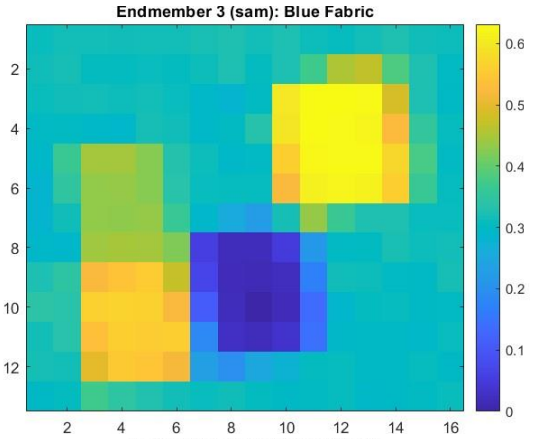


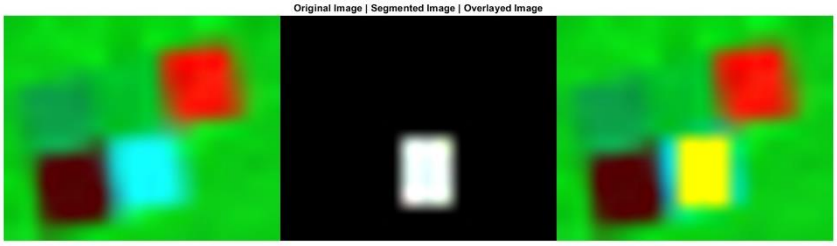
Threshold = 0.05



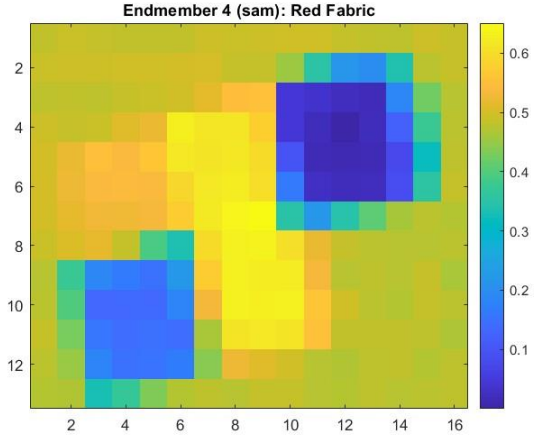


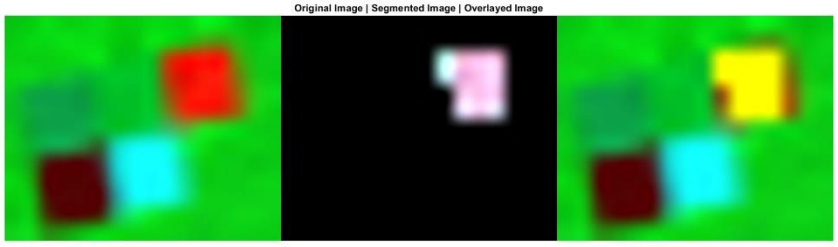
Threshold = 0.05



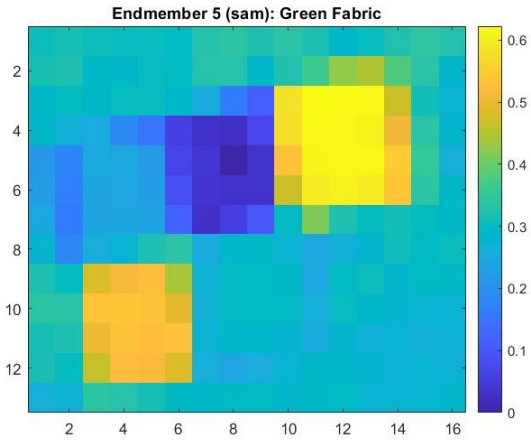


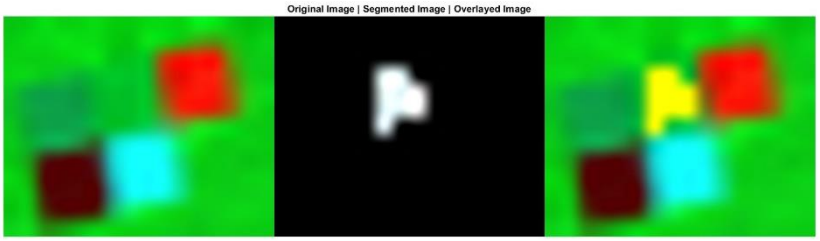
Threshold = 0.05



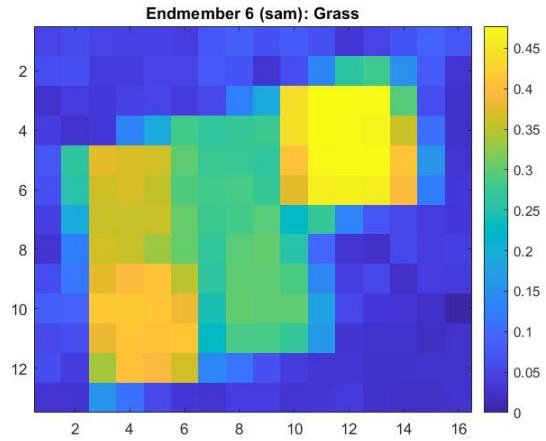


Threshold = 0.05





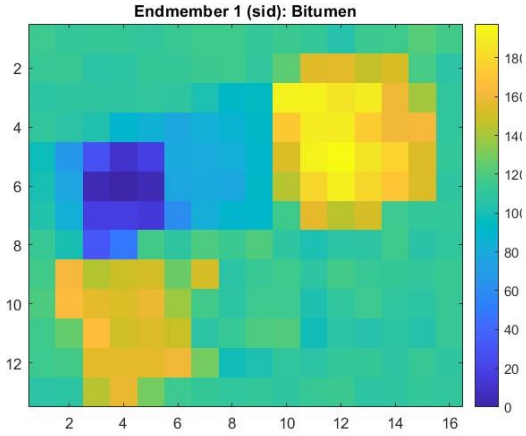
Threshold = 0.1

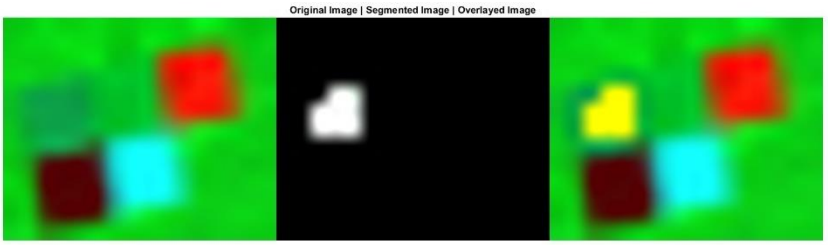


Threshold = 0.1

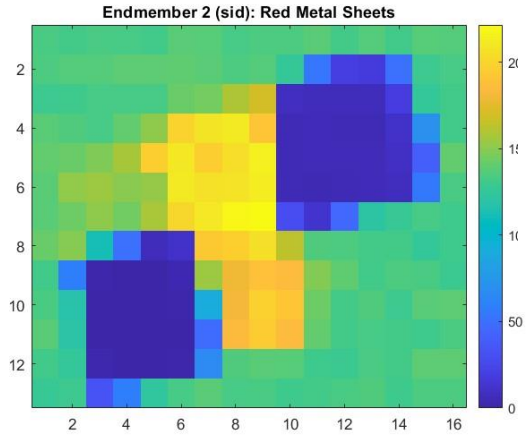
SAM on '*Large.tif*
• With larger object it's easier matching the spectra for each endmember, because we have less mixed pixels. • We can have the confirm that the colour red shared from two different objects (Red Fabric and Red Metal Sheets) can cause problems in the spectral match process but, with a proper threshold SAM, still can work well. This is less problematic in this case because of the more quantity of pixels we have per object considering they are the largest ones in the whole scene.

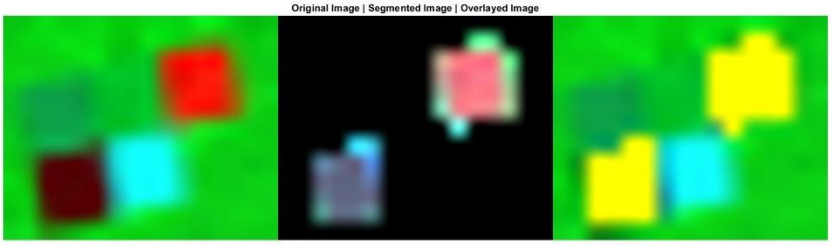
7.4.8.4 SID (‘*Larg.tif*’ Subset) experimental results analysis



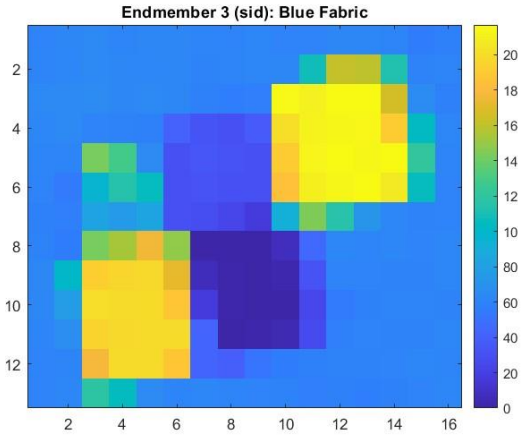


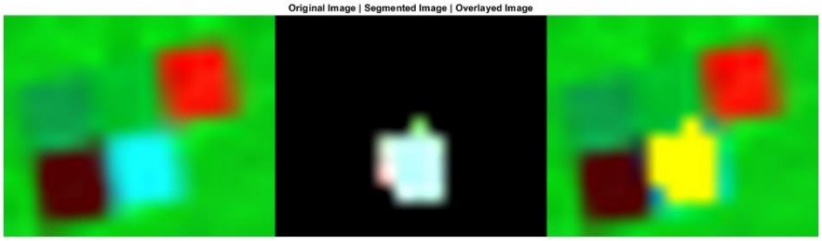
Threshold = 20



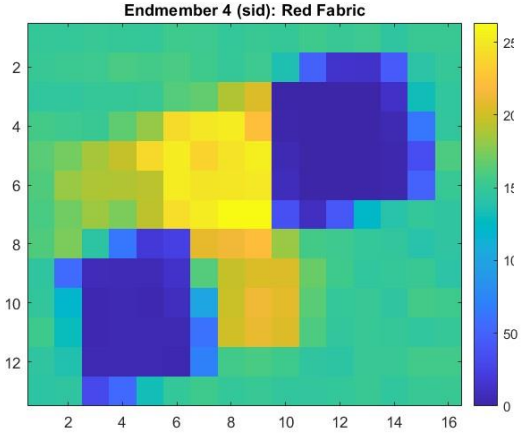


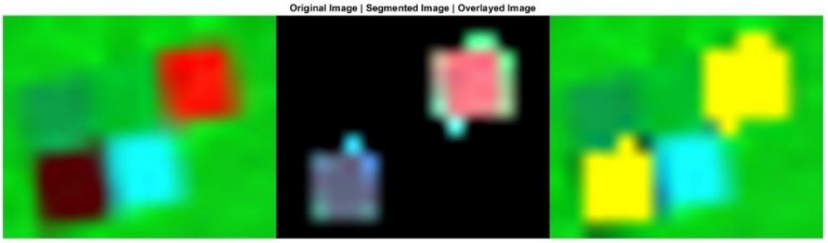
Threshold = 20



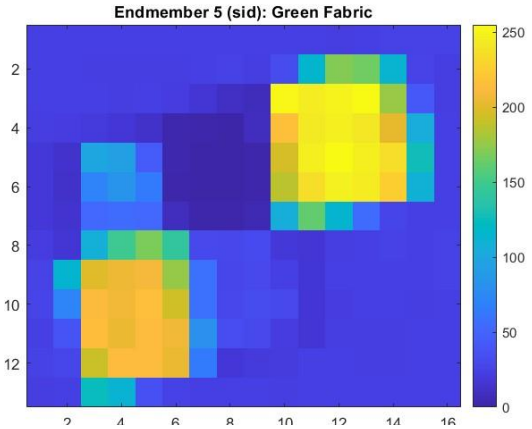


Threshold = 20

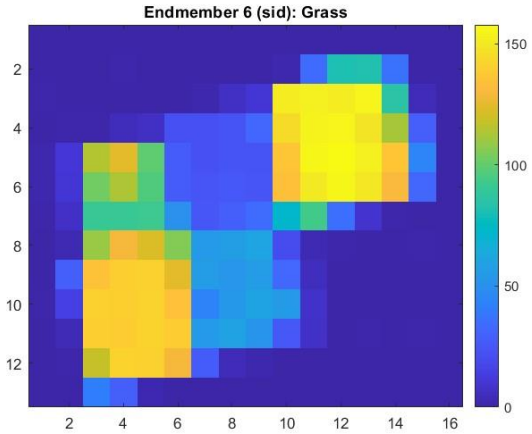


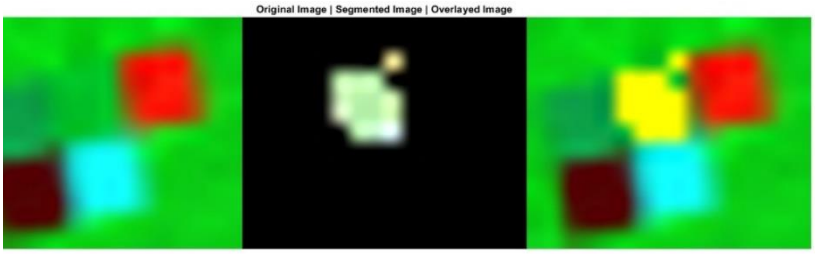


Threshold = 20



Threshold = 8





Threshold = 20

SID on '*Large.tif*
• Considering that SID didn't work well with Red objects that have similar in the whole scene, when we had both large and small objects, we tried to apply it with '*Large.tif*' to have another test and since it doesn't work well also in this case we can say that SID isn't recommended when you have multiple objects with the same colours. With less marked results, we can see this problem happening also between Green Fabric and Grass.

7.4.9 AbundanceMaps

```
abundanceMaps = AbundanceMaps(hcube_cell, lib_hs, spectraNames)
```

Function Purpose

The abundance map identifies the proportion of each endmember present in the spectra of each pixel. Each pixel is categorized as either pure or mixed, and the set of abundance values quantifies the proportion of each endmember within the pixel. For a hyperspectral data cube of spatial dimensions M-by-N containing P endmembers, there exist P abundance maps, each of size M-by-N. (Keshava N. J., 2002).

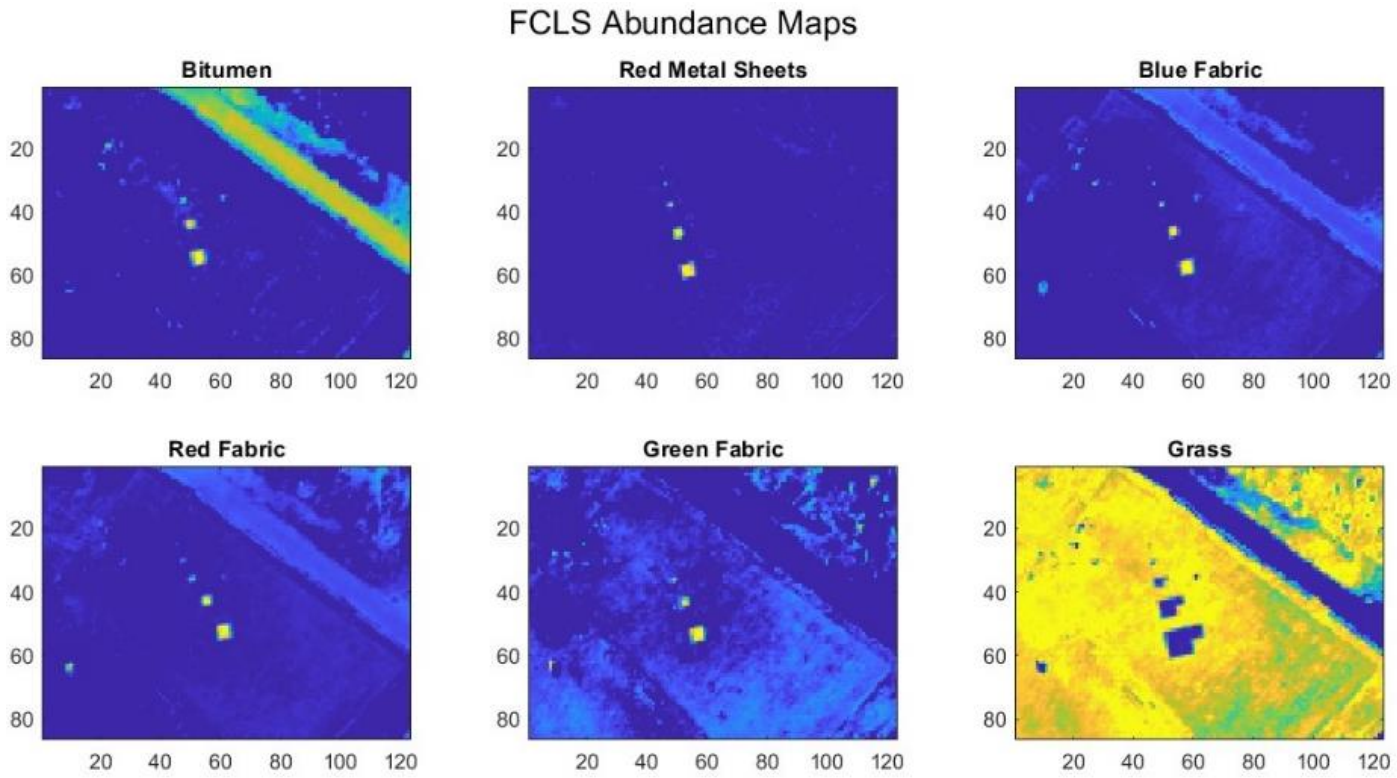
Input

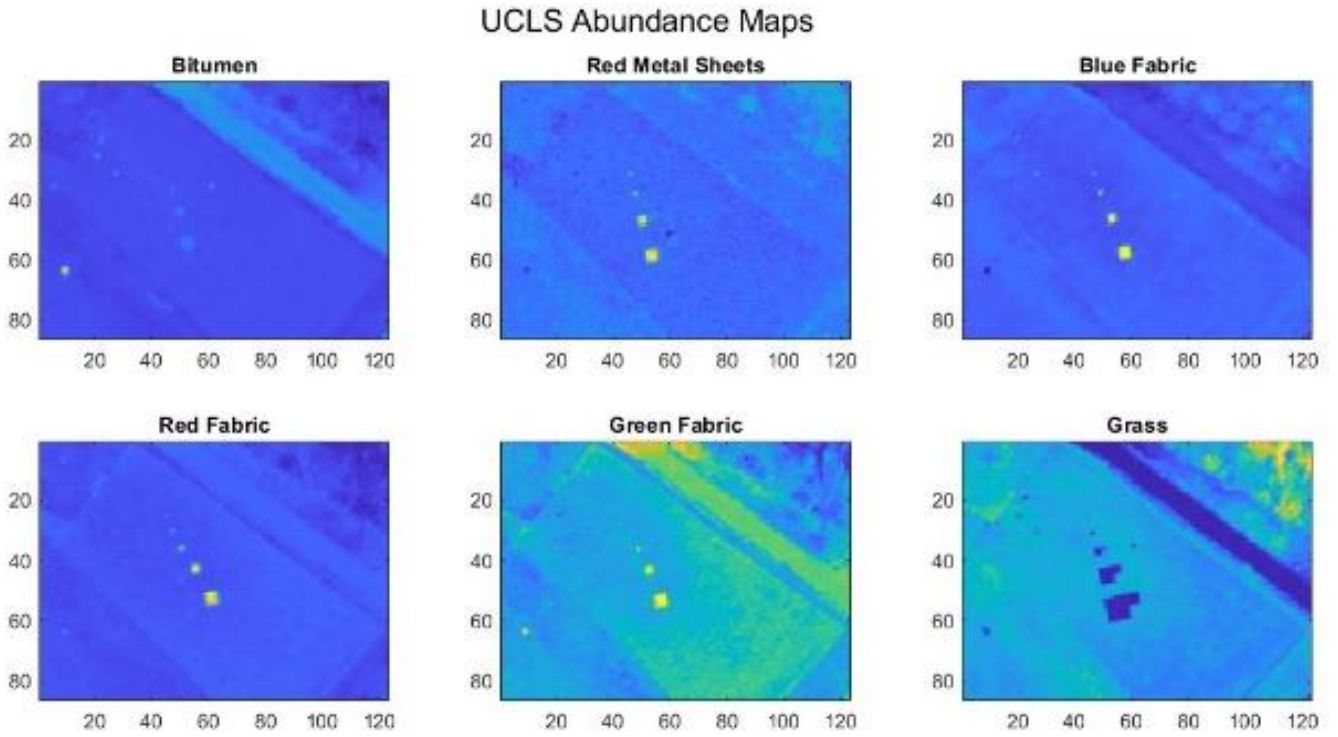
- **hcube_cell –** cell that contains the hcube for each dataset output of the `Hcube` function.
- **lib_hs –** endmembers spectral signature.
- **spectraNames** – names of endmembers extracted in `SpectralLibraries` function.

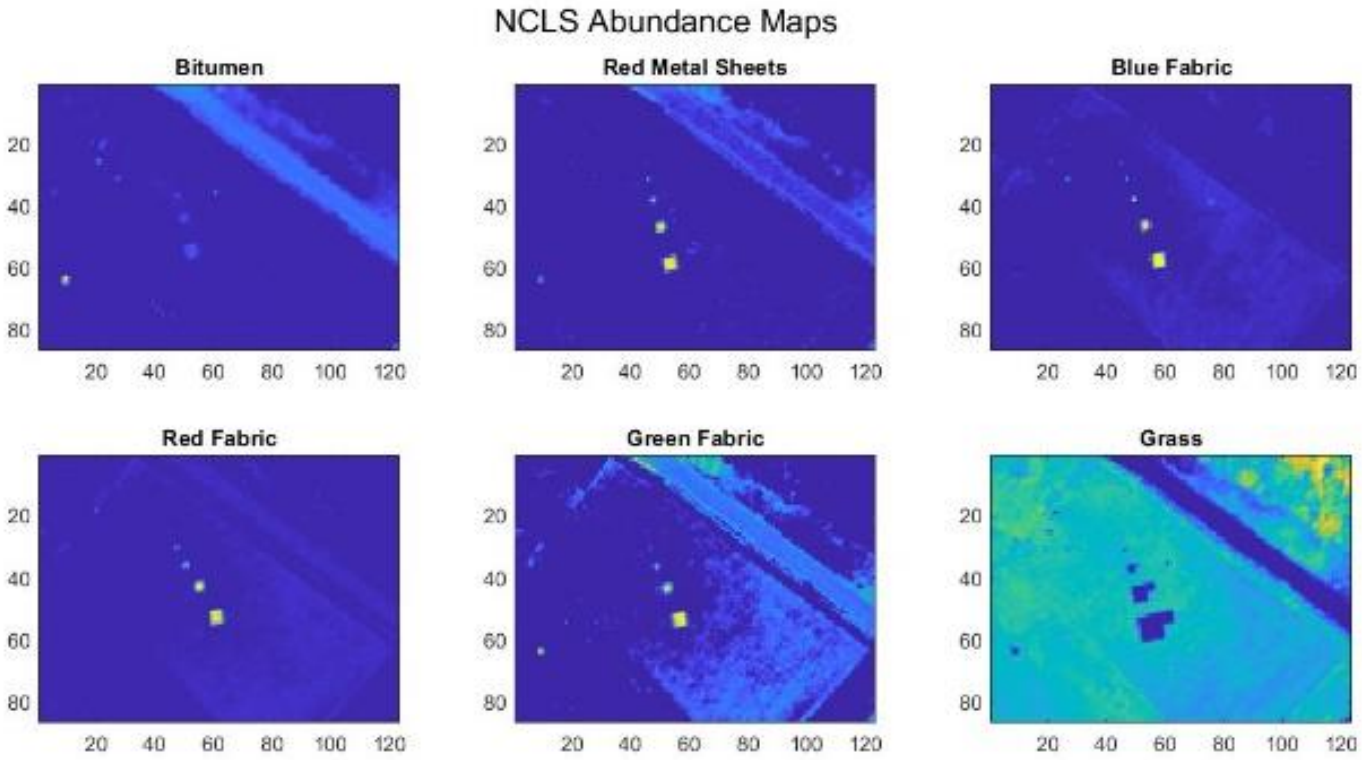
Output

- This function generates abundance maps of materials within a hyperspectral data cube using the following spectral unmixing methods: `methods = {'fcls', 'ucls', 'ncls'};` and visualizing the results. For each method, it computes the abundance map using the `estimateAbundanceLS` function, which estimates the abundance maps of the endmembers in a hyperspectral data cube by using the least-squares method. abundanceMap = estimateAbundanceLS(hcube_cell{choice}.DataCube, lib_hs, 'Method', method);
  (MathWorks, s.d.)

7.4.9.1 Experimental Results analysis on "*Full.tif*"




UCLS Abundance Maps
Bitumen
Red Metal Sheets
Blue Fabric
Red Fabric
Green Fabric
Grass
20
40
60
80
100
120


NCLS Abundance Maps
Bitumen
Red Metal Sheets
Blue Fabric
Red Fabric
Green Fabric
Grass
20
40
60
80
100
120

	FCLS	UCLS	NCLS
PRO	Best mean accuracy for the 3mx3m, 2mx2m, 1mx1m targets for each endmember. The only one that unmix Bitumen.	Works well with: Red Metal, Blue Fabric, Red Fabric and Green Fabric.	Works well with: Red Metal, Blue Fabric, Red Fabric and Green Fabric.
CONS	FCLS requires abundances to be non-negative (the so-called non-negativity constraint) and that they sum to one (the so-called sum-to-one constraint). This appears disadvantageous for unmixing small sub-pixels targets.	The lack of any abundance constraint seems to make UCLS particularly prone to noise, therefore delivering less meaningful abundances. Don't work well with Bitumen and Grass.	Don't work well with Bitumen and Grass.

7.4.10 AbundanceClassifier

```
AbundanceClassifier(abundanceMap, colormapColors, tickLabels, max_endmember)
```

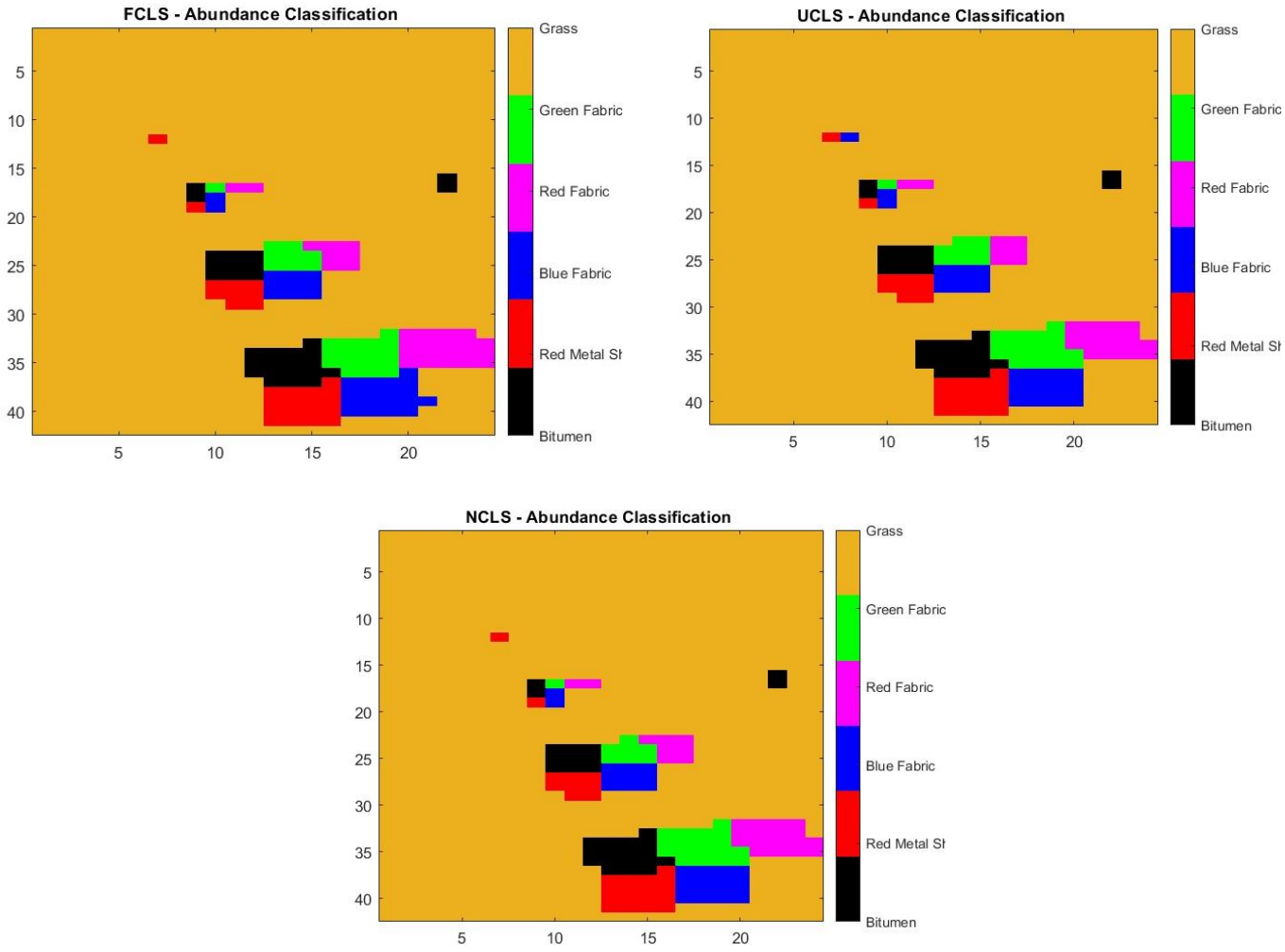
Function Purpose
This function shows the application of Maximum Abundance Classification (MAC) to identify distinct regions within a hyperspectral image. Each pixel is classified by selecting the channel number of the highest abundance value for each of them and associating it with the corresponding endmember class. The channel number returned for each pixel corresponds to the column in spectral_library that contains the endmember signature associated with the maximum abundance value of that pixel. (Mathworks, s.d.)

Input

- **abundanceMap** - abundance maps of materials computed in `AbundanceMaps` function, we consider plotting three different images, one for each method we used before: `methods = {'fcls', 'ucls', 'ncls'};`
- **colormapColors = colors –** our set of colors identifying specific endmember.
- **max_endmember –** max number of endmembers.

Output

- We then display a color-coded image of the pixels classified by maximum abundance value.



7.5 Conclusions

Spectral Unmixing is a powerful and crucial tool in hyperspectral remote sensing and geospatial analysis. The work developed was aimed at obtaining a performance evaluation of some techniques used to unmix hyperspectral imagery and over the development of this project, we've explored the state-of-art methodologies and applications of SU aiming to exploit its potential through a real-world scenario exploiting a benchmark-dataset proposed by DLR. In this concluding chapter, we sum up the key findings and insights gained through our investigation, underline the implications of SU and supply recommendations for possible future research and applications.

7.5.1 Key Findings and Insights

1) **Endmember Extraction**
   This is a pivotal step of Spectral Unmixing to get spectral signatures ground truth references to validate next results. Endmember libraries can be constructed using a variety of automatic techniques a posteriori (N-FINDR, PPI, FIPPI, etc..) or a priori including laboratory measurements and field surveys. This step can greatly influence the accuracy of the whole unmixing process.

2) **Abundance Estimation**
   The estimation of abundance maps within hyperspectral images is central to the spectral unmixing process. We've explored techniques such as linear unmixing, nonlinear unmixing, and sparse unmixing to get the fractional contributions of endmembers within each pixel. These abundance maps supply crucial information about land cover, material distribution, and changes over time.

3) **Algorithm Selection**
   We discussed different spectral unmixing algorithms, each with its own strengths and limitations. The choice of an appropriate algorithm depends on the characteristics of the data and the objectives of the analysis. While linear unmixing is computationally efficient and widely used, nonlinear and sparse unmixing methods offer enhanced capabilities for complex scenes.

4) **Applications**
   Spectral unmixing application are disparate: land cover classification, mineral exploration, environmental monitoring, and urban planning just to cite a few of them. These applications demonstrate the versatility and utility of spectral unmixing in addressing real-world challenges.

5) **Challenges**
   The research has not only enlightened the potential of Spectral Unmixing but also acknowledged its associated challenges. These include endmember variability, spectral mixing, and the curse of dimensionality. Addressing these challenges is pivotal for advancing the state of the art.

7.5.2 Implications

This methodology has the potential to revolutionize the way we observe and understand our planet. Its utility in the remote sensing domain has already led to improved resource management, early detection of environmental changes, and enhanced geological exploration. Furthermore, its integration with emerging technologies, such as machine learning and deep learning, promises even more accurate and efficient analyses.

7.5.3 Recommendations for Future Research

As we conclude this study, it is important to consider the following recommendations for future research in the field of spectral unmixing:

1) **Advanced Algorithms**
   After the research done during development's phase of this project, we recognized a lack of information available on the web especially when we talk about the MATLAB implementation's basic concepts: continued research and development of this paradigm is essential. This surely will include the exploration of machine learning techniques implemented in this field for improved abundance estimation, as well as the adaptation of SU to new data types and sources.

2) **Cross-Disciplinary Collaboration**
   Collaborative efforts and data/knowledge sharing between different kinds of scientist like computer science engineers, remote sensing experts, geoscientists, ecologists, and urban planners can lead to innovative applications and problem-solving in disparate domains.

3) **Data Fusion**
   Since Spectral Unmixing algorithms' evaluations and results strictly depend on the hyperspectral data integrity and solidity, we recognize that collecting as much of them also with different kind of sensors, such as LiDAR and SAR, would hold great potential for enhanced spectral unmixing results. To improve experiments, it's mandatory to overcome standard datasets and merge the key features of different sensors.

4) **Addressing Uncertainty**
   It's crucial to develop methodologies, new paradigms and create new tools to quantify and communicate the uncertainty associated with obtained abundance maps to get truthful results.

5) **Open-Source Reproducibility**
   The essential feature to improve the performance of Spectral Unmixing itself is to promote open-access spectral libraries, data, and open-source software tools.

In conclusion, spectral unmixing represents a new dynamic and evolving field that. Its impact on scientific research, resource management, and environmental stewardship cannot be overstated. By further exploring its potential, addressing challenges, and collaborating across disciplines, we can harness the full power of spectral unmixing to address the complex challenges of our time.

8 Bibliography

- Cerra, D., Pato, M., Alonso, K., Köhler, C., Schneider, M., de los Reyes, R., . . . Müller, R. (2021). DLR HySU—A Benchmark Dataset for Spectral Unmixing. *Remote Sensing*.

- Daniele Cerra, M. P. (2021). DLR HySU—A Benchmark Dataset for Spectral Unmixing. 1-28.

- *Hyperspectral Image Processing*. (s.d.). Tratto da Mathworks: https://it.mathworks.com/help/images/hyperspectral-image-processing.html

- Jiaojiao Wei, X. W. (2020). An Overview on Linear Unmixing of Hyperspectral Data. *Hindawi*.

- Keshava, N. (2003). A Survey of Spectral. *LINCOLN LABORATORY JOURNAL*.

- Keshava, N. J. (2002). Spectral Unmixing. *IEEE Signal Processing Magazine*.

- MathWorks. (s.d.). *estimateAbundanceLS*. Tratto da MathWorks: https://it.mathworks.com/help/images/ref/estimateabundancels.html

- MathWorks. (s.d.). *hypercube*. Tratto da MathWorks: https://it.mathworks.com/help/images/ref/hypercube.html

- Mathworks. (s.d.). *Hyperspectral Image Analysis Using Maximum Abundance Classification*. Tratto da Mathworks: https://it.mathworks.com/help/images/hyperspectral-image-classification-using-abundance-maps.html

- MathWorks. (s.d.). *nfindr*. Tratto da MathWorks: https://it.mathworks.com/help/images/ref/spectralmatch.html

- MathWorks. (s.d.). *ppi*. Tratto da MathWorks: https://it.mathworks.com/help/images/ref/ppi.html

- MathWorks. (s.d.). *spectralMatch*. Tratto da MathWorks: https://it.mathworks.com/help/images/ref/spectralmatch.html

- NV5GeospatialSoftware. (s.d.). *How does ENVI's Pixel Purity Index Work?* Tratto da nv5geospatialsoftware: https://www.nv5geospatialsoftware.com/Support/Maintenance-Detail/ArtMID/13350/ArticleID/19703/1631

- Tuszynski, J. (2023). *read_envihdr*. Tratto da Mathworks: https://www.mathworks.com/matlabcentral/fileexchange/38500-read_envihdr

- Valero, X. C. (2017). Processing Hyperspectral Images. In N. B. Zribi, *Optical Remote Sensing of Land Surface Techniques and Methods* (p. 163-200).

- X. Zhang, X. -h.-l. (2009). "An improved N-FINDR algorithm for endmember extraction in hyperspectral imagery.". *IEEE*.