



Hyperspectral Image Classification Based on Mathematical Morphology and Tensor Decomposition

Mohamad Jouni, Mauro Dalla Mura, Pierre Comon

► To cite this version:

Mohamad Jouni, Mauro Dalla Mura, Pierre Comon. Hyperspectral Image Classification Based on Mathematical Morphology and Tensor Decomposition. *Mathematical Morphology - Theory and Applications*, 2020, 4 (1), pp.1-30. 10.1515/mathm-2020-0001 . hal-02401272v2

HAL Id: hal-02401272

<https://hal.science/hal-02401272v2>

Submitted on 20 Jan 2020 (v2), last revised 30 Apr 2020 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mohamad Jouni¹ *, Mauro Dalla Mura^{1,2} , and Pierre Comon¹

Hyperspectral Image Classification Based on Mathematical Morphology and Tensor Decomposition

<https://doi.org/DOI>, Received ...; revised ...; accepted ..

Abstract: Hyperspectral Image (HSI) classification refers to classifying hyperspectral data into features, where labels are given to pixels sharing the same features, distinguishing the present materials of the scene from one another. Naturally a HSI acquires spectral features of pixels, but spatial features based on neighborhood information are also important, which results in the problem of spectral-spatial classification. There are various ways to account to spatial information, one of which is through Mathematical Morphology, which is explored in this work. A HSI is a **third-order** data block, and building new spatial diversities may increase this order. In many cases, since pixel-wise classification requires a matrix of pixels and features, HSI data are reshaped as matrices which causes high dimensionality and ignores the multi-modal structure of the features. This work deals with HSI classification by modeling the data as tensors of high order. More precisely, multi-modal hyperspectral data is built and dealt with using tensor Canonical Polyadic (CP) decomposition. Experiments on real HSI show the effectiveness of the CP decomposition as a candidate for classification thanks to its properties of representing the pixel data in a matrix compact form with a low dimensional feature space while maintaining the multi-modality of the data.

Keywords: Remote Sensing Image, Hyperspectral Imagery, Mathematical Morphology, Attribute Profiles, Tensor Decomposition, Scene Classification.

1 Introduction

Hyperspectral imagery refers to the acquisition of images of a scene over a wide and almost continuous spectrum, amounting to hundreds of spectral bands, to form at the end what is called a hypercube [34]. HSI is a tool in remote sensing that has found importance in many applications such as monitoring Earth resources, planetary observation, and biomedical image processing [21, 1, 27, 29, 2, 17]. Due to its significance of exploring the spectral properties of a spatial setting, HSI has become a wide area of research with a lot of explored and potential usages, and *image classification* of the scene is one of its most common.

At its core, image classification is the problem of assigning each pixel in the image a semantic label, which at the end allows the identification of materials in the scene into recognisable properties. This kind of procedure is called *pixel-wise classification*, a scenario where generally two main variables are required, the pixels as samples to be classified and the distinctive features (like spectral response or neighborhood information) as characteristics to be classified upon. Moreover, the samples-to-features relationship can be described in a matrix form.

*Corresponding author: Mohamad Jouni¹ , Email: mohamad.jouni@gipsa-lab.fr

Mauro Dalla Mura^{1,2} , Email: mauro.dalla-mura@gipsa-lab.fr

Pierre Comon¹ , Email: pierre.comon@gipsa-lab.fr

¹ Univ. Grenoble Alpes, CNRS, Grenoble INP, Gipsa-Lab, Images-Signal Department, 38000 Grenoble, France

² Tokyo Tech World Research Hub Initiative (WRHI), School of Computing, Tokyo Institute of Technology, Tokyo, Japan

Naturally a HSI acquires the spectral features of the scenes, mapping each pixel position, represented by a couple of positive integers (x, y) , to a vector of positive real values that correspond to the material's spectral response measured by the sensor, resulting in a structure that is often represented as a data cube of third-order¹. When the pixels are rearranged in lexicographic order as samples, i.e. the first two modes merge into one, the spectral mode can then represent the set of features, making it a direct way to classify a HSI. However, due to the high spectral resolution of HSI, pixels are considered to be in a very high dimensional feature space, which calls for dimensionality reduction. One conventional way to solve the problem of dimensionality is through Principal Component Analysis (PCA) [22], which works by reducing the number of features to that of the most significant principal components, then passing the corresponding second-order block of data to the classifier [31].

In fact, real HSI acquisition may contain mixed pixels, presenting challenges in acquiring the exact spectral features of the pixels and limiting the classification accuracy. As a result, the classification map may look noisy. Moreover, due to the advancement in optical sensor technology, spatial information surrounding the pixels is becoming more relevant especially when it comes to identifying relatively small objects. For this reason, it has become an important topic to incorporate spatial information extracted from the pixel's neighborhood in the classification, such that each pixel has features based on both spatial and spectral data [6, 16].

Classifying images based on spatial and spectral features is called *spectral-spatial classification*. Many of such works rely on tools in the context of *Mathematical Morphology* (MM) [32, 33, 28], which is perhaps one of the most popular ways to account to spatial information. Tools in MM incorporate spatial information of the pixel's neighborhood, obtained directly using successive morphological transformations of an image, and stacks them as pixel-wise spatial features. Examples of popular spatial features based on MM are the Morphological Profiles (MP) [5, 4, 37] and Attribute Profiles (AP) [12, 14, 3] depending on the type of morphological transformations that are carried out. For multivariate images (as in the case of HSI), such profiles are called Extended MP (EMP) and Extended AP (EAP), **generally known to be computed using one of two strategies; the conventional marginal strategy** where the profiles of the single grayscale image slices are **computed separately then stacked in one way** [4, 14], **and the vectorial processing strategy where the extended profile is computed by processing the different image slices simultaneously** [3]. If multiple kinds of attributes are considered at once, the structure is referred to as Extended Multi-Attribute Profiles (EMAP) [12]. It is worth noting here that HSI classification techniques that are purely defined in MM may end up in a very high-dimensional feature space caused by the stacking of spectral and spatial features along the same way, so they are usually combined with dimensionality reduction techniques in the form of pre-processing and, in some cases, post-processing [26].

Some works approach the problem of classification using *Composite Kernel* methods [7], by extracting spectral and spatial features and separately learning them through what would be a spectral kernel and a spatial one. Other works include contextual extraction and labeling through *Markov Random Fields* [36]. **Finally, it is worth mentioning that many works have been developed based on deep learning methods** [9, 8], where principal spectral information and spatial neighborhood patches around the pixels are distributed, processed, and learned over different deep neural network architectures (e.g. Applying Encoder and Convolutional Neural Networks). **While deep learning-based methods show significant performances, comparing them is out of the scope of this paper.**

One issue regarding the mentioned methods is that, even though they prove to improve significantly on HSI feature classification with respect to classification of only spectral features, they either tend to treat the different kinds of features as one entity such as stacking spectral and spatial features as one, or they provide separate treatment of the features but don't take into account exploring the natural high-order properties and relationships among the modes of the data blocks that could be obtained. In other words, such data blocks

¹ The order of an array refers to the number of its indices, and sometimes it is defined as the number of *ways* or *modes* in the array. A HSI is a third-order array, which can also be called a *three-way* array, and it is of dimensions $I_1 \times I_2 \times J$ where I_1 and I_2 represent the number of horizontal and vertical pixel positions respectively and J represents the number of spectral bands.

can show interesting relationships between the modes, including those among the features themselves, and these relationships are lost.

Tensor analysis has been increasing in popularity recently especially for applications that deal with such multi-modal data [39, 38, 41], and it is still under development. Tensors are high-order data structures often represented as multi-dimensional arrays. By keeping the features separated in different modes (into what we would call a higher-order arrangement of the features compared to when they are stacked along the same mode), the ordering of the data and the intrinsic connections are respected, and these connections can be accounted to through tools defined in multi-linear algebra, and specifically through *tensor decomposition*.

In fact, *tensor modeling* of HSI for the sake of classification has previously been visited [39, 18, 23, 24]. In [39], which is close to our work, a so-called Tensor-PCA (TPCA) is used as a high-order form of the conventional PCA method following the concept of High-Order Singular Value Decomposition (HOSVD) [15] for tensors. A rough description of the method is visualized in figure 1. Briefly speaking, the singular matrices obtained from the [Singular Value Decomposition \(SVD\)](#) of the matrix-unfoldings in the feature modes are computed and truncated to the most significant principal components. After that, they are multiplied by the original tensor for the sake of post-processing dimensionality reduction in the feature space. However, when it comes to the classification phase, it still boils down to rearranging the data into matrix form and merging the feature modes. In order to avoid breaking the tensor structure at the classification phase, one possible way is to approach the whole problem differently through tensor *Canonical Polyadic Decomposition (CPD)* [11].

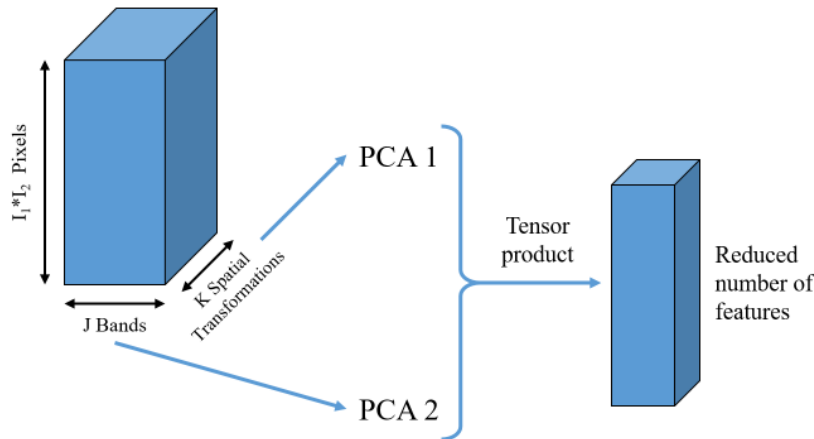


Fig. 1: An illustration of the TPCA framework for dimensionality reduction [39]

Through CPD, our main goal is to *directly* obtain a matrix of samples and features through one tensor decomposition that:

- Intuitively accounts to the high-order relationships.
- Provides the matrix with low-dimensional feature space (i.e. low-rank decomposition).
- Conserves the information found in the tensor.

This goal was explored with preliminary results that can be found in ISMM 2019 [23] and IGARSS 2019 [24] contributions, and the following is an extension of those two. [23] and [24] include preliminary results using AMD and EMAP as morphological settings applied on one dataset. The current version includes more detailed analysis, better modeled results, more morphological settings, an additional dataset, and further discussions regarding the possible use of unconstrained CPD and the potential unmixing based on different types of features.

First, we start by creating high-order tensors from HSI using spatial transformations derived from MM, then we jointly handle the different variety of features by means of nonnegative CPD. For example, in terms of spatial transformations, one might want to create multiple modes, each based on a certain morphological concept, which results in a block of data that is at least of order four: i.e., two for spatial dimensions (image

rows and columns), one for spectral features, and another one for spatial transformations. Normally, the first two modes are merged into one (that is to say that the pixels are rearranged in lexicographic order), which leads to a tensor of order three at least, where each pixel represents at least a matrix of features. For instance, if the tensor is of order four or higher, one pixel starts to represent a tensor of features. This cannot be directly fed to the classifier, and here comes the significance of CPD.

CPD is a tool which decomposes a tensor into several matrices, often called the “factor matrices”, each of which directly represents and corresponds to one of the modes of the tensor. This is to say that CPD directly produces a matrix that represents the mode of pixels in what we would call a “compact matrix form”, which we choose to provide to the classifier, without the need to “matricize” the tensor by rearranging the feature modes. We also take advantage of other properties of this decomposition. For example, CPD enjoys uniqueness under mild conditions, it represents the pixels in a relatively low dimensional feature space and accounts to the data structure as a whole despite its complexity, and it is flexible with constraints (we mainly mention the nonnegativity constraint on the factors which is an important feature since it potentially allows physical interpretation of the decomposed data [38]). Experiments are conducted on two real HSI datasets. For each dataset, we show some experiments done over tensors of order three, having one spectral mode and another spatial one, and other experiments done over tensors of order four, having one spectral mode and two spatial ones. For tensors of order three, we have two kinds of experiments, the first one is based on EMP, and the second one is based on an Additive Morphological Decomposition (AMD) inspired by [39]. For tensors of order four, the experiments are based on an EMAP formed using four different kinds of attribute transformations. We note that in [12], an EMAP was built when multiple EAP of the same HSI are concatenated along the same mode for classification, instead, we stack them as a new mode in the tensor. The experiments show promising results in improving the classification accuracy compared to other methods in the literature. Moreover, some discussion is added in the case of unconstrained CPD.

From here on, this paper is organized as follows. We start by giving the notation in Section 2. Section 3 serves as a background on MM for image processing and talks about the higher-order arrangement of the transformations by adding new modes to the HSI based on MM. In Section 4 we talk about tensors and describe the CP decomposition and its functionality in the framework of classification. In Section 5 we talk about the contributions of this work. In Section 6 we present the experiments and discuss the results as mentioned earlier. Finally, a conclusion is drawn in Section 7.

2 Notation

We assume that a HSI of order three is denoted as $\mathcal{I} \in \mathbb{R}^{I_1 \times I_2 \times J}$. \mathcal{I} can be seen as a mapping from the discrete vector subset $E \subset \mathbb{Z}^2$ to the real vector subspace $F \subseteq \mathbb{R}^J$. In other words, a pixel's coordinates of horizontal and vertical position $(i_1, i_2) \in E$, upper-bounded by (I_1, I_2) , is mapped to a vector $\mathbf{x} \in F$ of J real values, where x_j represents the spectral response of the pixel at the j -th channel ($j \in \{1 \dots J\}$).

In general, scalars are denoted by unformatted letters, e.g. $x \in \mathbb{R}$, $I \in \mathbb{R}$. Vectors are denoted by bold lower-case letters, e.g. $\mathbf{v} \in \mathbb{R}^I$. Matrices are denoted by bold upper-case letters, e.g. $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$. Higher order tensors (say of order N , N -way tensor) are denoted by upper-case calligraphic letters, e.g. $\mathcal{T} \in \mathbb{R}^{I_1 \times \dots \times I_N}$.

Mainly in section 6, we fix some notations to avoid confusion. A three-way HSI cube is referred to as $\mathcal{I} \in \mathbb{R}^{I_1 \times I_2 \times J}$. A tensor of order four or more built upon the three-way HSI is referred to as $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times J \times K \times \dots}$. A tensor of order three or more, created from \mathcal{T} by rearranging the pixels in lexicographic order, is referred to as $\mathcal{Y} \in \mathbb{R}^{I \times J \times K \times \dots}$ where $I = I_1 \times I_2$ is the number of pixels in the scene. A tensor of order three or more, created by compressing \mathcal{Y} as will be explained in section 4.2, is referred to as \mathcal{G} . The factor matrices of the CPD are referred to as $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \dots$, in the order of the modes, except in section 4.1 since it provides a general overview on CPD, where they will be referred to as \mathbf{H}_i where i indicates the order of the mode. Finally, the core tensor of the CPD is referred to as $\mathbf{\Lambda}$.

3 Background on Mathematical Morphology

In this section, we start by introducing the basic concepts and notions of MM as the set of transformations in image processing; we recall some definitions and properties of morphological and attribute filters that are used in this project, then we explain briefly the concept of morphological and attribute profiles. After that, we move on to talk about the case examples for building high-order multi-modal data; the EMP-based example and the ones inspired by the AMD and the EMAP methods.

3.1 Basics of Mathematical Morphology

In image processing, morphological operators are image-to-image neighborhood-based transformations. Let's assume that Ψ is a morphological transformation, then Ψ maps an image to another where the output value of a pixel is **dependent on the values of its neighboring pixels**. For a given 2D image \mathbf{I} , we define some properties of image-based transformations:

- Idempotence: $\Psi(\Psi(\mathbf{I})) = \Psi(\mathbf{I})$
- Extensivity: $\Psi(\mathbf{I}) \geq \mathbf{I}$
- Anti-extensivity: $\Psi(\mathbf{I}) \leq \mathbf{I}$

where inequalities are understood entry-wise.

Such transformations are important in image processing for instance to extract features of interest in the image based on spatial characteristics. Depending on the desired spatial characteristics, we focus on two families of filters defined in the framework of MM, morphological filters based on structuring elements (SE) [35] and attribute filters [13].

3.1.1 Morphological Filters Based on Structuring Elements (SE)

This family of operators is mostly used for geometrical features of the image. Such operators are defined by the type of transformation and the size and shape of the neighborhood, with the latter being the major key point, known as the structuring element, **which is a mask of predefined shape and size that is meant to interact with the image and govern the interaction of pixels with their neighborhood**. Suppose for $\text{SE}_2 \subseteq \text{SE}_1$, we say that Ψ follows the absorption law if $\Psi_{\text{SE}_1}(\Psi_{\text{SE}_2}) = \Psi_{\text{SE}_1}$.

For example, Erosion (ϵ_{SE}) and Dilation (δ_{SE}) are two basic morphological operators; Erosion shrinks regions with values that are greater than those of their surrounding pixels by means of the SE while Dilation expands them. Another two types of such operators are Opening (γ_{SE}) and Closing (ϕ_{SE}), Opening is the Dilation of the Erosion of an image \mathbf{I} by means of the same SE while Closing goes in reverse. As a result, we can say that Opening removes white spaces that are smaller than the SE while Closing fills dark spaces that are smaller than the SE. In many cases, we are interested in conserving the details of objects that are larger than the SE (objects which remain unfiltered). This leads us to the family of operators by reconstruction, where an image transformation is repeated until idempotence is reached under the constraint of a mask (\mathbf{M}) that is the original image itself ($\mathbf{M} = \mathbf{I}$), we denote such transformations by $\Psi_{\text{SE}}^{\infty}(\mathbf{M}, \mathbf{I})$ [40]. Opening by Reconstruction $\gamma_{\text{SE}}^{\infty}(\mathbf{M}, \mathbf{I})$ and Closing by Reconstruction $\phi_{\text{SE}}^{\infty}(\mathbf{M}, \mathbf{I})$ are examples of that.

We note that Opening by reconstruction and Closing by reconstruction follow the absorption law, and they are widely used for Morphological Profiles (MP), which is to be described in section 3.2.

3.1.2 Attribute Filters

Attribute filters (attribute thinning and thickening) were introduced as a new technique to extract distinctive types of spatial features, called attributes, while addressing some limitations of filters based on SE [12]. Such

filters can be geometric (analogous to morphological Opening and Closing), or based on characteristics of the data distribution. The concept of attribute filtering is not based on SE, but totally on connected regions of the image sharing similar *attribute* characteristics and rearranged as component trees. In other terms, attribute filtering is defined by a criterion that evaluates the desired attribute according to a specified threshold. Any property that applies to image regions can be considered as an attribute, [examples of that](#) are the area of the regions, the moment of inertia, the standard deviation, *etc.*

Attribute thinning and thickening can be used for building Attribute Profiles (AP), which is to be described in section 3.2.

3.2 Morphological and Attribute Profiles

Before we dive into the definition of such profiles, we note that attribute thinnings and thickenings are analogous to morphological openings and closings by reconstruction respectively. In terms of properties, thinning and opening by reconstruction are anti-extensive, while thickening and closing by reconstruction are extensive, but, conversely to opening and closing, thinning and thickening are not increasing transformations [12, 14]. As such, we proceed to explain the concept of profiles considering the case of MP noting that the transition to explaining the case of AP is obvious.

For a set of extensive and anti-extensive dual operators, respectively $\{\bar{\Psi}^i, \underline{\Psi}^i\}_{i=1\dots m}$, where m denotes the size of the used filter (sometimes denoted by λ for AP), the MP of a grayscale image \mathbf{I} can be defined as:

$$\text{MP}(\mathbf{I}) := \{\underline{\Psi}^m(\mathbf{I}), \dots, \underline{\Psi}^1(\mathbf{I}), \mathbf{I}, \bar{\Psi}^1(\mathbf{I}), \dots, \bar{\Psi}^m(\mathbf{I})\}, \quad (1)$$

More precisely, to create a MP, these morphological transformations of the same image are stacked and rearranged along the third mode in increasing order of extensivity, that is, the elements of $\text{MP}(\mathbf{I})$ are ordered such that $\text{MP}(\mathbf{I})_j < \text{MP}(\mathbf{I})_{j+1}$ where $\text{MP}(\mathbf{I})_j$ refers to the j -th element of $\text{MP}(\mathbf{I})$, with the original image stored at the middle. For multivariate images such as HSI, the concept is extended but the same idea still holds. In this case, where we focus on the marginal strategy, the profiles of each grayscale image (whether the image is a principal component or the original band itself) are concatenated along the third mode forming what is referred to in the literature as an “extended profile”, which is then given as an input to the classifier. In the case of MP, extended profiles are referred to as EMP [4]. Figure 2 shows an illustration of conventionally creating an EMP of a multivariate image using Opening and Closing by reconstruction.

Additionally, we note that creating morphological diversity for multivariate images is not limited to extended profiles. Works in the literature extended this concept to other possible rearrangements and derivations such as the Differential Morphological Profile (DMP) [5], which is created by stacking images obtained from the differences between the successive transformations and thus can be seen as the derivative of MP. Another example is AMD, which we will [talk about in more detail in the next subsection](#).

In the case of AP, extended profiles are referred to as EAP, and, as mentioned at the beginning of this subsection, creating an EAP is analogous to that of an EMP, but using successive values of the attribute parameter. Furthermore, the work on EAP for image classification was extended to EMAP, which is to be discussed in the next subsection.

3.3 High-Order Tensors Using MM

Having talked about the various ways to build spatial features using MM for image classification, we mention three examples that we adopt in our experiments; the first one is the EMP method which is direct and simple, the second one is a method inspired by AMD because it provides significant features and it has been previously used as a tensor model for multi-modal feature space reduction, and the third one is inspired by EMAP as a way to create more than one spatial diversity, all of which are explained in the following.

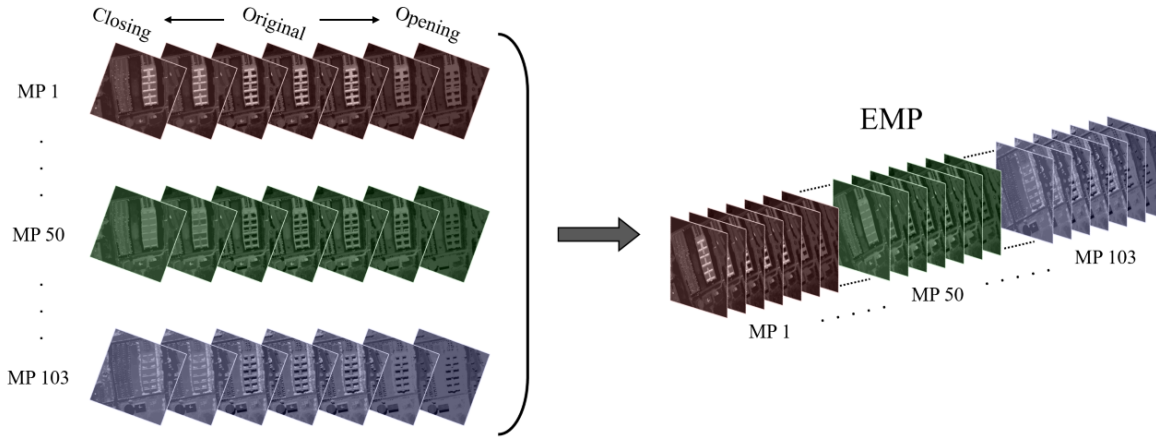


Fig. 2: An illustration showing one way of creating an EMP of a multivariate image. On the left, it shows sample Morphological Profiles (MP) of some spectral bands, numbered by the index of the spectral band, of a cropped portion of the HSI of Pavia University. On the right, it shows the stacking of the MPs along the third mode to form the Extended Morphological Profile (EMP). The transformations were carried out using three different SE; disks with sizes [1,6,11].

3.3.1 Fourth-Order Tensor Using EMP

As mentioned earlier, the conventional way to build the EMP of a HSI is by stacking the MP of the single spectral bands along the spectral mode, forming a third-order block of data. This process considers single morphological transformations ($\Psi(\mathcal{I})$) only as a mapping from a 2D image to another. Instead, we choose to consider the transformation of a HSI ($\Psi(\mathcal{I})$) as another third-order cube of data, i.e. as a mapping from a 3D image to another, under the same assumption that the transformation of a HSI is equivalent to applying the same transformation on its bands one by one. As a result, the EMP of a HSI can be seen as a higher-order MP arrangement, or in other words, a fourth-order analogy to the MP of a grayscale image, by stacking the third-order transformations of the HSI as follows:

$$\text{MP}(\mathcal{I}) := \{\underline{\Psi}^m(\mathcal{I}), \dots, \underline{\Psi}^1(\mathcal{I}), \mathcal{I}, \bar{\Psi}^1(\mathcal{I}), \dots, \bar{\Psi}^m(\mathcal{I})\}, \quad (2)$$

The structure in (2) brings the same values as those found in the conventional EMP, except that the former is rearranged to separate spatial features from spectral ones. The result is a fourth-order block of data of size $I_1 \times I_2 \times J \times K$, where $I_1 \times I_2$ is the number of pixels, J is the number of spectral bands, and $K = 2m + 1$ is the number of morphological transformations. Figure 3a shows an illustration of a fourth-order tensor built through EMP.

3.3.2 Fourth-Order Tensor Using AMD

An AMD is formed by decomposing an image in an additive manner such that the morphological transformations sum up to give back the image itself. What is stacked in this case is not directly the original image and its transformations, but the components of the additive decomposition that are composed of one “structure component” and the “residuals”. Following the same notation of $\{\bar{\Psi}^i, \underline{\Psi}^i\}_{i=1 \dots m}$ from Section 3.2, we give the formulas of the consecutive residual components as defined in [39]:

$$\mathbf{R}_i^+ = \bar{\Psi}^i(\bar{\Psi}^{i-1}(\mathcal{I})) - \bar{\Psi}^{i-1}(\bar{\Psi}^{i-2}(\mathcal{I})) \geq 0 \quad (3)$$

$$\mathbf{R}_i^- = \underline{\Psi}^{i-1}(\underline{\Psi}^{i-2}(\mathcal{I})) - \underline{\Psi}^i(\underline{\Psi}^{i-1}(\mathcal{I})) \geq 0 \quad (4)$$

where the inequalities are understood entry-wise. The AMD of an image \mathbf{I} can then be written as:

$$\begin{aligned}\mathbf{I} &= \frac{\bar{\Psi}^m(\bar{\Psi}^{m-1}(\mathbf{I})) + \Psi^m(\Psi^{m-1}(\mathbf{I}))}{2} + \sum_{i=1}^m \frac{\mathbf{R}_i^- - \mathbf{R}_i^+}{2} \\ &= \mathbf{S} + \sum_{i=1}^m \mathbf{R}_i,\end{aligned}\quad (5)$$

where the term \mathbf{S} is considered to be the structure component because it contains the unfiltered components in the decomposition, and the terms $\{\mathbf{R}_i\}_{i=1\dots m}$ are called the residuals because they contain the residues obtained from the differences between each two successive morphological transformations. In the case of HSI, the structure and the residuals are of order three, and at the end they are stacked along the fourth mode to create a fourth-order block of data.

In our approach, we choose to build the morphological decomposition using the set $\{\mathbf{S}, \mathbf{R}_i^-, \mathbf{R}_i^+\}_{i=1\dots m}$ instead of $\{\mathbf{S}, \mathbf{R}_i\}_{i=1\dots m}$ to preserve the nonnegativity of the data, which will be relevant for physical interpretation in the future. Thus, for the sake of consistency, we refer to it as Positive-AMD (PAMD). As such, the obtained fourth-order hyperspectral data set is denoted by \mathcal{T} and arranged in such an order: $[\mathbf{S}, \mathbf{R}_1^-, \mathbf{R}_1^+, \dots, \mathbf{R}_m^-, \mathbf{R}_m^+]$, which is represented by an $I_1 \times I_2 \times J \times K$ array, and $K = 2m + 1$ is the number of terms along the morphological decomposition. Figure 3b shows an illustration of a fourth-order tensor built through EMP.

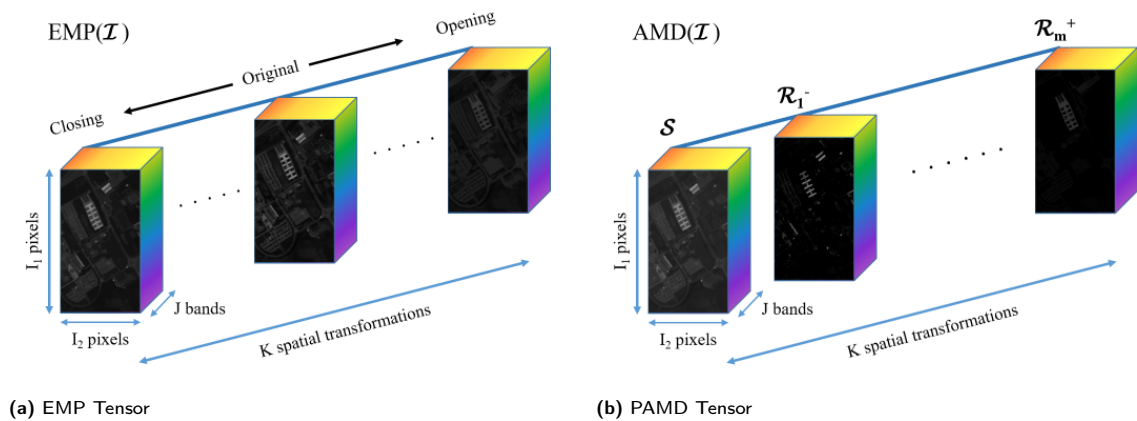


Fig. 3: An illustration of fourth-order tensors using EMP and PAMD

3.3.3 Fifth-Order Tensor Using EMAP

Similarly to EMP, an EAP of a multivariate image is normally a third-order block of data where the spectral and spatial features are stacked along the same mode. We also choose to separate the features to form a block of data of size $I_1 \times I_2 \times J \times K$, where K is the number of attribute transformations. The concept of EMAP is to stack multiple EAP of the image so we would have a variety of spatial information for each pixel depending on different types of attributes before passing it to the classifier. In [12], EMAP was used by concatenating four kinds of EAP on a multivariate image composed of the principal components of the original HSI. We note that an EMAP is normally a third-order block of data, which means that the spectral features are mixed with the spatial ones, and that the multiple attribute profiles are mixed as well.

In our approach, we compute more than one type of EAP, each of which are of order four, then we stack them in parallel to form a fifth mode that can be seen as an acquisition of different types of EAP. The obtained fifth-order block of data is denoted by $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times J \times K \times L}$, with L being the number of attributes taken into consideration.

4 Tensor Decomposition

After obtaining the data set as a result of morphological transformations, we now deal with it as a tensor of high-order. In fact, each pixel now is composed of at least a matrix of features which is indeed complex. Tensor CP decomposition is a powerful tool to break the complexity of tensors into data forms (matrices) that are easy and intuitive to deal with without the need to alter the structure of the tensor. In other words, we show that decomposing the high-order image data using CP decomposition provides a direct and simple way to represent the pixels in matrix form that is to be classified.

In this section, we address the tensor decomposition aspect of this work from a theoretical point of view. We start by explaining the CP decomposition with nonnegative constraints involved, then we talk about High-Order Singular Value Decomposition (HOSVD) as means of compressing the data. Following this in section 5, we first dive into the algorithmic part and our implementation of this work, then we talk about the importance of using tensor decomposition when multiple types of features are involved.

Previously, we explained how tensor \mathcal{T} can be of fourth and fifth order. In fact, since a 2D image is usually of full rank which normally equals either minimum of its two dimensions, then higher order data sets built upon this image should be of higher rank. For that, we merge the first two modes corresponding to pixel positioning by rearranging them in lexicographic order, which in turns reduces the rank of the data and directly works in the favor of classification since the latter requires one mode of pixels (seen as samples). As a result, our tensors to be decomposed will be at least of order three, denoted by \mathcal{Y} with a first mode of dimension $I = I_1 \times I_2$.

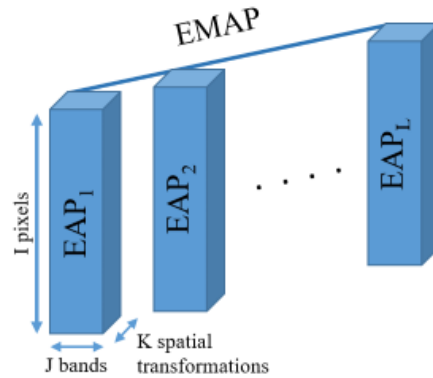


Fig. 4: An illustration of fourth-order tensor using EMAP after merging the first two modes of pixel positioning

For the sake of consistency, explanations in this section will be on general tensors. Only in this section, the notation of the dimensions of a tensor will have the same symbol indexed by the order of the mode. For example, an N -th order tensor \mathcal{Y} is of size $I_1 \times \dots \times I_N$.

4.1 CP Decomposition

An N th-order decomposable tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is a tensor that can be written as the outer product of N vectors $\mathbf{h}^{(d)} \in \mathbb{R}^{I_d}$ (for $d \in \{1 \dots N\}$):

$$\mathcal{D} \stackrel{\text{def}}{=} \mathbf{h}^{(1)} \otimes \dots \otimes \mathbf{h}^{(N)} \quad (6)$$

Any tensor \mathcal{Y} can be written as a weighted sum of decomposable tensors \mathcal{D}_r (supposing that \mathcal{D}_r are normalized) such that:

$$\mathcal{Y} = \sum_{r=1}^R \lambda_r \mathcal{D}_r, \quad (7)$$

where the tensor rank, usually denoted by R , is the least number of terms such that the CP decomposition (7) holds exact. Starting from the fact that each decomposable tensor \mathcal{D}_r can be written as in (6), the set of vectors in the outer products of $\mathcal{D}_{r=1\dots R}$ for a fixed mode $d \in \{1 \dots N\}$, denoted by $\mathbf{h}_{r=1\dots R}^{(d)}$, can be rearranged as the columns of a matrix such that $\mathbf{H}_d = [\mathbf{h}_1^{(d)}, \dots, \mathbf{h}_R^{(d)}] \in \mathbb{R}^{I_d \times R}$. Along the same line, the weights of λ_r can be rearranged in a diagonal tensor $\mathbf{\Lambda} \in \mathbb{R}^{I_1 \times \dots \times I_N}$.

Consequently, the CP decomposition can be written using an alternative notation:

$$\mathcal{Y} = (\mathbf{H}_1, \dots, \mathbf{H}_N) \cdot \mathbf{\Lambda} \quad (8)$$

The set of \mathbf{H}_d are usually referred to as the factor matrices, and their column vectors are sometimes called the components. $\mathbf{\Lambda}$ is called the core tensor of the decomposition which governs the interactions between the components of the factor matrices. Since $\mathbf{\Lambda}$ is diagonal in the CP decomposition, only the components that share the same index r can interact with each others, and R can be seen as directly related to the degrees of freedom in the decomposition.

Furthermore, it is important that the rank is relatively small (compared to the dimensions of the tensor) for the decomposition to be identifiable, up to scaling and permutation ambiguities. A generic bound on the rank would be to start from the inequality that the number of equations should be greater than the number of unknowns, leading to:

$$R \leq \frac{\prod_{d=1}^N I_d}{\sum_{d=1}^N I_d - N + 1}, \quad (9)$$

which is almost always verified since the value $\frac{\prod_{d=1}^N I_d}{\sum_{d=1}^N I_d - N + 1}$ is usually large, especially when one of the dimensions is the total number of pixels. Another bound is that of Kruskal [25], which is based on the Kruskal rank of the factor matrices. Kruskal's bound states that if the rank is small enough and the components of the factor matrices don't show collinearity, then the decomposition is identifiable:

$$R \leq \frac{1}{2} \left(\sum_{d=1}^N k_{\mathbf{H}_d} - N + 1 \right), \quad (10)$$

where $k_{\mathbf{H}_d}$ is the Kruskal rank of factor matrix \mathbf{H}_d .

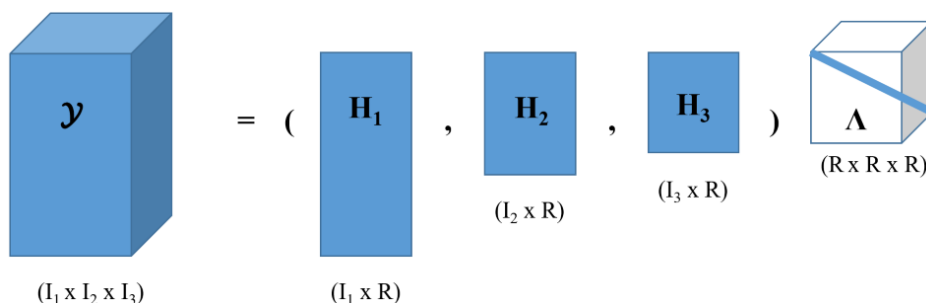


Fig. 5: An illustration of a third-order tensor CP decomposition

In its computation, the CP decomposition is practically approximated by minimizing the cost function:

$$\frac{1}{2} \|\mathcal{Y} - (\mathbf{H}_1, \dots, \mathbf{H}_N) \cdot \mathbf{\Lambda}\|^2 \quad (11)$$

However, in many cases where \mathcal{Y} is nonnegative, it is desirable that the factors of the decomposition are nonnegative especially when it comes to physical interpretation where negative factors don't have any meaning.

In such cases, the rank is called the nonnegative rank and is usually denoted by R_+ . It is also shown that a best nonnegative rank- r approximation exists [30]. We write the cost function to be minimized incorporating the nonnegativity constraints on the factor matrices as follows:

$$\begin{aligned} & \frac{1}{2} \|\mathbf{Y} - (\mathbf{H}_1, \dots, \mathbf{H}_N) \cdot \mathbf{\Lambda}\|^2 \\ & \text{s.t. } \mathbf{H}_1 \succeq 0, \dots, \mathbf{H}_N \succeq 0 \end{aligned} \quad (12)$$

An interesting aspect that comes as a result of computing the CP decomposition is the relationship between the tensor and its factor matrices. For a start, the factor matrices are directly related to the modes of the tensor in such a way that \mathbf{H}_d represents the d -th mode of \mathbf{Y} . Generally speaking, an element y_{i_1, \dots, i_N} of \mathbf{Y} is decomposed into R_+ terms, and it can be recovered from the respectively-indexed rows in each of the corresponding factor matrices (i.e. the i_d -th row of \mathbf{H}_d).

4.2 High-Order Singular Value Decomposition (HOSVD)

Because of the huge dimensions of our data especially that of the first mode (the high number of pixels), directly applying the CP decomposition is exhaustive. In order to deal with this issue, one would want to compress the data without losing information. One way to do so is by compressing the data into the minimal dimensions such that all information is conserved. For example, for a tensor of order **four** of size $I_1 \times I_2 \times I_3 \times I_4$, the dimensionality of the mode of pixels can be very big compared to the product of those of the other modes (i.e. $I_1 \gg I_2 I_3 I_4$). Thanks to HOSVD, such tensor can be compressed to a size of $(I_2 I_3 I_4) \times I_2 \times I_3 \times I_4$ supposing that $R_+ \leq \prod_{d=2}^4 I_d$.

The method is explained as follows. Let's assume we have a fourth-order tensor $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$, and we denote the d -mode unfolding (matricization) of the tensor by $\mathbf{Y}_{(d)}$, created when the tensor is reshaped into matrix form where one mode is of dimension $\prod_{k=1}^4 I_k |_{k \neq d}$ (taking the first mode) and the other is of dimension I_d (taking the second mode). The economic SVD of each unfolding is computed such that:

$$\begin{aligned} \mathbf{Y}_{(d)} &= \mathbf{U}_d \mathbf{\Sigma}_d \mathbf{V}_d^T \\ \forall d &= \{1, 2, 3, 4\} \end{aligned} \quad (13)$$

This decomposition reveals the ranks of the unfoldings, denoted by R_d ($d = \{1, 2, 3, 4\}$), and the right singular matrices \mathbf{V}_d are orthogonal matrices of dimensions $I_d \times R_d$. In our case where the number of pixels is huge, $R_1 \leq I_2 I_3 I_4$, and for $d = \{2, 3, 4\}$ $R_d \leq I_d$. The set of right singular matrices are then taken as the new basis, upon which \mathbf{Y} is projected and in turns compressed into a new tensor \mathbf{G} . Hence the HOSVD in this case can be written as follows:

$$\mathbf{Y} = \mathbf{G} \bullet_1 \mathbf{V}_1 \bullet_2 \mathbf{V}_2 \bullet_3 \mathbf{V}_3 \bullet_4 \mathbf{V}_4 = (\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_4) \cdot \mathbf{G} \quad (14)$$

Since the d -mode unfoldings are almost always full-rank, then for $d = \{2, 3, 4\}$ $R_d = I_d$, so only the first mode shall be compressed in order to reduce the complexity of computations, which means that only \mathbf{V}_1 is computed from SVD while the others are chosen as Identity matrices in (14). \mathbf{G} is then of size $(I_2 I_3 I_4) \times I_2 \times I_3 \times I_4$, without loss of information. Algorithm 1 shows our implementation to compress an N -th order tensor using HOSVD; we note that **way_c** is a vector containing the modes that we want to compress and **dim_c** is a vector of the compressed dimension size (in case there is a desire to truncate the singular vectors of \mathbf{V}_d).

5 Contributions

Computing the CP decomposition of a compressed tensor when nonnegative constraints are involved can be tricky to implement, and the issue has been addressed in [10]. Recently, Alternating Optimization - Alternating Direction Method of Multipliers (AO-ADMM) was introduced in [20] as a powerful algorithm to compute the

Algorithm 1 COMPRESS**Require:** \mathcal{Y} , way_c , dim_c **for** $d \in \text{way}_c$ **do** Unfold \mathcal{Y} into $\mathbf{Y}_{(d)}$ such that the mode d takes the second way of the matrix; Compute the right singular matrix from the SVD of $\mathbf{Y}_{(d)}$, denoted by \mathbf{V}_d ; Truncate the columns of \mathbf{V}_d by $\text{dim}_c(d)$;**end for****for** $d \notin \text{way}_c$ **do** $\mathbf{V}_d = \mathbf{I}_{\text{dim}_c(d)}$; (Identity matrix)**end for** $\mathcal{G} = \mathcal{Y} \bullet_1 \mathbf{V}_1^\top \bullet_2 \dots \bullet_N \mathbf{V}_N^\top$;**return** \mathcal{G} and $\mathbf{V}_1, \dots, \mathbf{V}_N$

CPD and a flexible one with constraints. This being said, in the first part of this section, we talk about our implementation of these constraints in the framework of AO-ADMM.

In the second part, we dive into our specific application and talk about the advantages of combining spectral and spatial features using tensor decomposition in the framework of classification.

5.1 Alternating Optimization - Alternating Direction Method of Multipliers (AO-ADMM)

Alternating Least Squares (ALS) is one of the most popular algorithms to compute the CP decomposition (8). As the name suggests, the point is to break down the non-convex objective function (11) into multiple convex least-squares sub-problems by fixing all the factor matrices in the minimization except for \mathbf{H}_d , and alternating over them, such that:

$$\mathbf{H}_d = \underset{\mathbf{H}_d}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{Y}_{(d)} - \mathbf{W} \mathbf{H}_d^\top\|^2 \quad (15)$$

$$\forall d = 1 \dots N,$$

where \mathbf{W} contains the Khatri-Rao product of all the factor matrices excluding \mathbf{H}_d . When constraints are imposed on the factor matrix, supposedly represented by function $r(\mathbf{H}_d)$, then (15) is to be modified as such:

$$\mathbf{H}_d = \underset{\mathbf{H}_d}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{Y}_{(d)} - \mathbf{W} \mathbf{H}_d^\top\|^2 + r(\mathbf{H}_d) \quad (16)$$

$$\forall d = 1 \dots N,$$

In our case, we note that the input tensor is the one obtained after compression, that is denoted by \mathcal{G} , which allows to contain nonnegative entries. However, the original tensor is nonnegative and its factor matrices are constrained to be so. Following our description in section 4.2, only \mathbf{H}_1 is compressed, so assuming $\mathbf{H}_1 = \mathbf{V}_1 \mathbf{H}_{1_c} \succeq 0$ where \mathbf{H}_{1_c} is the compressed version of \mathbf{H}_1 , our optimization problem can be reformulated such that:

$$\underset{\mathbf{H}_{1_c}, \dots, \mathbf{H}_N, \mathbf{\Lambda}}{\operatorname{argmin}} \frac{1}{2} \|\mathcal{G} - (\mathbf{H}_{1_c}, \dots, \mathbf{H}_N) \cdot \mathbf{\Lambda}\|^2 \quad (17)$$

$$\text{s.t. } \mathbf{V}_1 \mathbf{H}_{1_c} \succeq 0, \mathbf{H}_2 \succeq 0, \dots, \mathbf{H}_N \succeq 0, \mathbf{\Lambda} \succeq 0$$

In order to cope with the constraints of compression and nonnegativity at once, we present a solution that is inspired by [10], with AO-ADMM being the adopted algorithm for its efficiency and flexibility with constraints [20]. One can assume that ALS is at the base of AO-ADMM, in which the alternating process is at the level of the least-squares sub-problems (16), each of which is solved through the ADMM algorithm.

Using ADMM, it is possible to solve the sub-problem of each factor matrix with the corresponding constraints and alternate accordingly. Hence, solving (17) boils down to the following updates, defined for $d = \{2, \dots, N\}$:

$$\left. \begin{aligned} \tilde{\mathbf{H}}_{1_c} &\leftarrow (\mathbf{W}^\top \mathbf{W} + \rho \mathbf{I})^{-1} (\mathbf{W}^\top \mathbf{G}_{(1)} + \rho (\mathbf{U} + \mathbf{H}_{1_c})^\top) \\ \mathbf{H}_{1_c} &\leftarrow \mathbf{V}_1^\top \max(0, \mathbf{V}_1 (\tilde{\mathbf{H}}_{1_c}^\top - \mathbf{U})) \\ \mathbf{U} &\leftarrow \mathbf{U} + \mathbf{H}_{1_c} - \tilde{\mathbf{H}}_{1_c}^\top \end{aligned} \right\} \quad (18)$$

$$\left. \begin{aligned} \tilde{\mathbf{H}}_d &\leftarrow (\mathbf{W}^\top \mathbf{W} + \rho \mathbf{I})^{-1} (\mathbf{W}^\top \mathbf{G}_{(d)} + \rho (\mathbf{U} + \mathbf{H}_d)^\top) \\ \mathbf{H}_d &\leftarrow \max(0, \tilde{\mathbf{H}}_d^\top - \mathbf{U}) \\ \mathbf{U} &\leftarrow \mathbf{U} + \mathbf{H}_d - \tilde{\mathbf{H}}_d^\top \end{aligned} \right\} \quad (19)$$

We show through algorithms 2, 3, and 4 how we implement the AO-ADMM method. In Algorithm 3, \mathbf{Y} is a matrix updated by $\mathbf{Y}_{(d)}$, \mathbf{W} is the Khatri-Rao product defined in Algorithm 2, k is the input rank of the decomposition, we refer to [20] for μ , ϵ is the tolerance for r and s which we refer to [20] for their updates, *constraint* is a string defining the type of proximity update, i_{max} is the number of inner ADMM iterations (0 if no limit), and \mathbf{V}_d is the compression matrix. Finally, we recall that the dimensions of $\mathbf{Y}_{(d)}$ are $\prod_{j \neq d} I_j \times I_d$, those of \mathbf{W} are $\prod_{j \neq d} I_j \times R$, and those of $\tilde{\mathbf{H}}_d$, \mathbf{H}_d^\top and \mathbf{U}^\top are $R \times I_d$.

Computational complexity

We refer to paper [20] for the detailed explanations concerning the CPD by using AO-ADMM. Let us consider a third-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ of relatively low rank R , and we consider the complexity as per ADMM iterations (algorithm 3). Also, it is important to note that the matrices \mathbf{W} and \mathbf{Y} are independent from the inner-ADMM updates, so they can be used only once to compute the Cholesky decomposition and the product $\mathbf{W}^\top \mathbf{Y}$, whose values can be cached before the $\tilde{\mathbf{H}}$ update allowing to save a lot of repetitive computations. Now, we split the cases into four:

1. Unconstrained CPD: In this case, the complexity of the algorithm is dominated only by the updates of $\tilde{\mathbf{H}}$. Hence, the complexity is $\mathcal{O}(I_d R^2)$.
2. Nonnegative CPD: Nonnegative constraints on the factor matrices require only element-wise projection, i.e. a complexity of $\mathcal{O}(I_d R)$, which is negligible compared to $\mathcal{O}(I_d R^2)$. Hence, the same complexity of unconstrained CPD still dominates.
3. Unconstrained CPD of the compressed tensor (compression along the mode of pixels, Section 4.2): This case requires one step after the whole AO-ADMM framework to decompress the first factor matrix back into \mathbf{H}_1 , which is the final product before the classification phase. However, the ADMM updates by themselves remain unchanged with a gain on the complexity $\mathcal{O}(I_1 R^2)$ since I_1 here is compressed to a much lower value, i.e. $I_2 I_3$. The complexity of the ADMM updates generally remains dominated by $\mathcal{O}(I_2 I_3 R^2)$ per iteration, and that of the decompression step is $\mathcal{O}(I_1 I_2 I_3 R)$.
4. Nonnegative CPD of the compressed tensor (compression along the mode of pixels, Section 4.2): This case affects only the ADMM updates of the first factor matrix, specifically at the level of \mathbf{H}_{1_c} in equation (18) due to the decompression and compression steps. The complexity increases by $\mathcal{O}(I_1 I_2 I_3 R)$ per iteration for the first factor matrix \mathbf{H}_{1_c} , but remains unchanged for the others (i.e. $\mathcal{O}(I_d R^2)$, $d = 2, 3$).

In the case of TPCA, a PCA is carried out for every matrix $\mathbf{Y}_{(d)}$ that corresponds to a feature mode in order to finally perform the tensor product that would reduce the dimensionality of the feature modes separately. This means that for the same third-order tensor considered earlier, two PCA steps are carried out. Each PCA holds as much complexity to it as that of an SVD, that is $\mathcal{O}(mn^2)$ for a matrix of dimensions $m \times n$ where $m \geq n$. Considering in this case that the PCA is carried out for the second and third matrix-unfoldings, i.e. $\mathbf{Y}_{(2)}$ and $\mathbf{Y}_{(3)}$, then the complexities are $\mathcal{O}(I_1 I_2 I_3 I_2)$ and $\mathcal{O}(I_1 I_2 I_3 I_3)$ respectively. After that, in order to obtain the tensor with reduced feature dimensions, say $\mathcal{G} \in \mathbb{R}^{I_1 \times R_2 \times R_3}$, through tensor product, the complexity is as much as $\mathcal{O}(I_1 R_2 R_3 I_2 I_3)$.

For the sake of comparison, this means that for each additional feature mode of dimension I_N in the tensor, the complexity of the CPD in the first three cases will only add by $\mathcal{O}(I_N R^2)$ (one additional matrix update). In the fourth case, the complexity of the decompression and compression step increases to $\mathcal{O}(I_1 \dots I_N R)$ (multiplies once by the new dimension). In the case of TPCA, it increases by $\mathcal{O}(I_1 \dots I_N I_d)$ for every $d = \{1, \dots, N\}$, with an additional load at the tensor product phase, i.e. $\mathcal{O}(I_1 R_2 \dots R_N I_2 \dots I_N)$. In general, R is much smaller than, or at most comparable to, $\prod_{d=2}^N R_d$. Finally, the computational complexity is better in the case of CPD for most cases, but the choice of the parameters can affect this conclusion.

Algorithm 2 Alternating Optimization (Least Squares loss)

Require: $\mathcal{Y}, \mathbf{H}_1, \dots, \mathbf{H}_N, \mathbf{U}_1, \dots, \mathbf{U}_N$
 $\mathbf{H}_1, \dots, \mathbf{H}_N$ are initialized; $\mathbf{U}_1, \dots, \mathbf{U}_N$ are initialized to zero matrices;
for $d = 1, \dots, N$ **do**
 Store the different unfoldings $\mathbf{Y}_{(d)}$;
end for
repeat
 for $d = 1, \dots, N$ **do**
 $\mathbf{W} = \odot_{j \neq d} \mathbf{H}_j$;
 update \mathbf{H}_d and \mathbf{U}_d using Algorithm 3;
 end for
 update μ if necessary; (refer to [20] for the update of μ)
until some termination criterion is reached (number of iterations)
 Normalize the columns of the factor matrices and store the weights in $\mathbf{\Lambda}$;
return $\mathbf{H}_1, \dots, \mathbf{H}_N, \mathbf{\Lambda}$

Algorithm 3 ADMM of a mode d

Require: $\mathbf{Y}, \mathbf{W}, \mathbf{H}, \mathbf{U}, k, \mu, \epsilon, \text{constraint}, i_{max}, \mathbf{V}$
 \mathbf{H} and \mathbf{U} are already initialized;
 $\mathbf{G} = \mathbf{W}^\top \mathbf{W}$;
 $\rho = \text{trace}(\mathbf{G})/k$;
Calculate \mathbf{L} from Cholesky decomposition such that
 $\mathbf{G} + (\rho + \mu)\mathbf{I}_k = \mathbf{L}\mathbf{L}^\top$;
 $\mathbf{F} = \mathbf{W}^\top \mathbf{Y}$;
 $\mathbf{H}_f = \mathbf{H}$;
repeat
 $\tilde{\mathbf{H}} \leftarrow (\mathbf{L}^\top)^{-1} \mathbf{L}^{-1} (\mathbf{F} + \rho(\mathbf{H} + \mathbf{U})^\top + \mu \mathbf{H}_f^\top)$; (See [20] for μ and \mathbf{H}_f)
 $\mathbf{H} \leftarrow \text{proximity}(\text{constraint}, \tilde{\mathbf{H}}^\top, \mathbf{U}, \mathbf{V})$; (refer to Algorithm 4)
 $\mathbf{U} \leftarrow \mathbf{U} + \mathbf{H} - \tilde{\mathbf{H}}^\top$;
 Update r and s ; (refer to [20] for the updates of r and s)
until ($r < \epsilon$ and (s is undefined or $s < \epsilon$)) or ($i_{max} > 0$ and $i \geq i_{max}$)
return \mathbf{H} and \mathbf{U}

Algorithm 4 Proximity Update of \mathbf{H} in ADMM

Require: $\text{constraint}, \mathbf{H}_t, \mathbf{U}, \mathbf{V}$
switch (constraint)
 case Nonnegativity:
 $\mathbf{H} \leftarrow \mathbf{H}_t - \mathbf{U}$;
 $\mathbf{H} \leftarrow \mathbf{H}^+$;
 case Compression and Nonnegativity:
 $\mathbf{H}_u \leftarrow \mathbf{V}(\mathbf{H}_t - \mathbf{U})$;
 $\mathbf{H} \leftarrow \mathbf{V}^\top (\mathbf{H}_u^+)$;
end switch
return \mathbf{H}

5.2 Spectral and Spatial Features Using Tensor Decomposition

Now, in our particular case of HSI tensors, and based on what was discussed in section 3, we recall that the first mode spans the pixels, the second mode spans the spectral bands, the third mode spans morphological transformations, and the potential fourth mode spans different kinds of morphological concepts. For simplicity, we keep the following explanation short to third-order tensors, then higher-order tensors follow suite.

We show an illustration of some relationships between a third-order tensor and the components of its CPD in figure 6. Here, we expect that \mathbf{H}_1 represents the mode of pixels, \mathbf{H}_2 represents that of spectral bands, and \mathbf{H}_3 represents the corresponding spatial diversity. Accordingly, each row in \mathbf{H}_1 describes the composition of a high-order pixel in \mathcal{Y} of the same index (i.e. by fixing the pixel's index in the tensor as illustrated in figure 6 in green):

$$y_{\mathbf{p}, i_2, i_3} = \sum_{r=1}^{R_+} h_{\mathbf{p}, r}^{(1)} \cdot h_{i_2, r}^{(2)} \cdot h_{i_3, r}^{(3)} \cdot d_r \quad (20)$$

$$1 \leq i_2 \leq I_2, 1 \leq i_3 \leq I_3$$

Additionally, the coefficients found in one row in \mathbf{H}_1 (across the columns) are related to their counterparts in \mathbf{H}_2 and \mathbf{H}_3 , both of which represent feature information (figure 6 highlights in orange similarly indexed columns from each of the factor matrices along with the corresponding coefficient in Λ). In other words, these coefficients in \mathbf{H}_1 describe the spectral and spatial information that spread out in \mathbf{H}_2 and \mathbf{H}_3 respectively. Consequently, \mathbf{H}_1 can be seen as a matrix of samples and features where spectral and spatial features are combined.

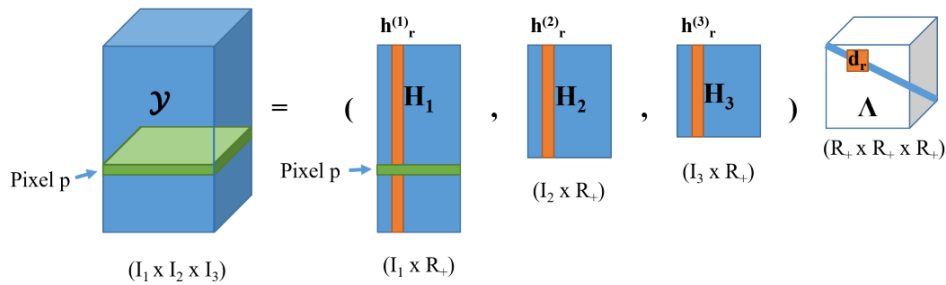


Fig. 6: An illustration of some relationships between the tensor and its CP decomposition. “Pixel p” refers to a pixel at index p . $h_r^{(1)}$, $h_r^{(2)}$, and $h_r^{(3)}$, are the r -th columns of matrices \mathbf{H}_1 , \mathbf{H}_2 , and \mathbf{H}_3 respectively. d_r is the r -th diagonal element of Λ .

Thanks to the nonnegative constraints in the decomposition, the components of the factor matrices can hold physical interpretation. For instance, the columns of \mathbf{H}_2 can be seen as spectral signatures, and those of \mathbf{H}_1 , when reshaped, can be seen as corresponding grayscale images. This can find its uses in Hyperspectral Unmixing, which is presently not our concern.

Briefly speaking, CP decomposition directly provides a simple and low dimensional representation of the data. First, the complexity of the tensor is reduced to simple matrices, each linked to one of its modes, with an intuitive approach to account to the high-order relationships in the data and sometimes with barely any loss of information. Second, the high-order feature aspect of the pixels in \mathcal{Y} boils down to row vectors of R_+ elements in \mathbf{H}_1 . Third, there are often redundancies in the tensor, which makes R_+ relatively small and corresponds to the low dimensionality of the feature space. Finally, since classification is our main concern, our target is to classify the matrix representing the pixel mode, \mathbf{H}_1 . We think of the rows as samples and the components as features in the classification. For that, we conduct some experiments that are to be shown in the following section.

6 Experiments

In this section, we talk about the experiments conducted in this work. First, we give a description of the data sets that we use. Then we present the experimental set-ups and their corresponding results for further discussion. In order to demonstrate our method, we use two real HSI datasets as shown in section 6.1.

In the results sections, we demonstrate the results on Pavia University first, then we show those of DFC. For each HSI, we show three different kinds of morphological applications:

- EMP: Based on morphological operators. The decomposed tensor is [three-way](#). The spatial way is the stacking of the transformations as they are.
- PAMD: Based on morphological operators. The decomposed tensor is [three-way](#). The spatial way is a form of a successive differential stacking of the transformations.
- EMAP: Based on attribute operators. The decomposed tensor is [four-way](#). One spatial way corresponds to attribute indexing (the kind of attribute), and the other one is the stacking of the transformations as they are.

Moreover, we make some remarks on different aspects of the used decomposition. As a start, we notice that there are two main variables that can influence the decomposed data; the number of AO-ADMM iterations and the rank of the decomposition, which are usually provided as inputs to the CPD. For instance, the higher the number of iterations or the value of the rank is, the less the reconstruction error, so we expect the decomposed data to better represent the original tensor. Another aspect would be the constraints. Since we use nonnegativity constraints on the original factor matrices while carrying the CPD on the compressed versions, the decomposition can take a lot of time compared to its unconstrained counterpart, so we carry out some experiments using unconstrained CPD. At the end, we show a sample result of a decomposed component from the nonnegative CPD and explain how it can be made possible to potentially interpret the results in terms of unmixing based on spectral-spatial features.

6.1 Description of Data-Set

6.1.1 Pavia University

The first HSI, known as Pavia University, was taken over the University of Pavia and acquired by the ROSIS sensor. The image has a spatial size of 610×340 pixels with a geometric resolution of 1.3 meters, and consists of 103 spectral bands. The groundtruth image is included in the data set and it consists of [nine](#) classes: trees, asphalt, bitumen, gravel, metal sheets, shadows, self-blocking bricks, meadows, and bare soil. Additionally, 40002 pixels are available as test set, and 3921 pixels are available as training set. The training set is fixed. Figure 7 shows the HSI in false colors (by choosing the bands 58, 34, and 17 as Red, Green and Blue components) and the training set [and ground-truth](#) used for classification.

6.1.2 Data Fusion Contest (DFC) image for IEEE GRSS 2013

The second HSI was acquired over the University of Houston campus and the neighboring urban area by the ITRES-CASI 1500 sensor. The image has a spatial size of 349×1905 pixels with a geometric resolution of 2.5m, and consists of 144 spectral bands. The groundtruth image is included in the data set and it consists of 15 classes: grass healthy, grass stressed, grass synthetic, tree, soil, water, residential, commercial, road, highway, railway, parking lot 1, parking lot 2, tennis court, and running track. Additionally, 12197 pixels are available as test set, and 2832 pixels are available as training set. The training set is fixed. Figure 7 shows the HSI in false colors (by choosing the bands 65, 40, and 22) and the training set [and ground-truth](#) used for classification.

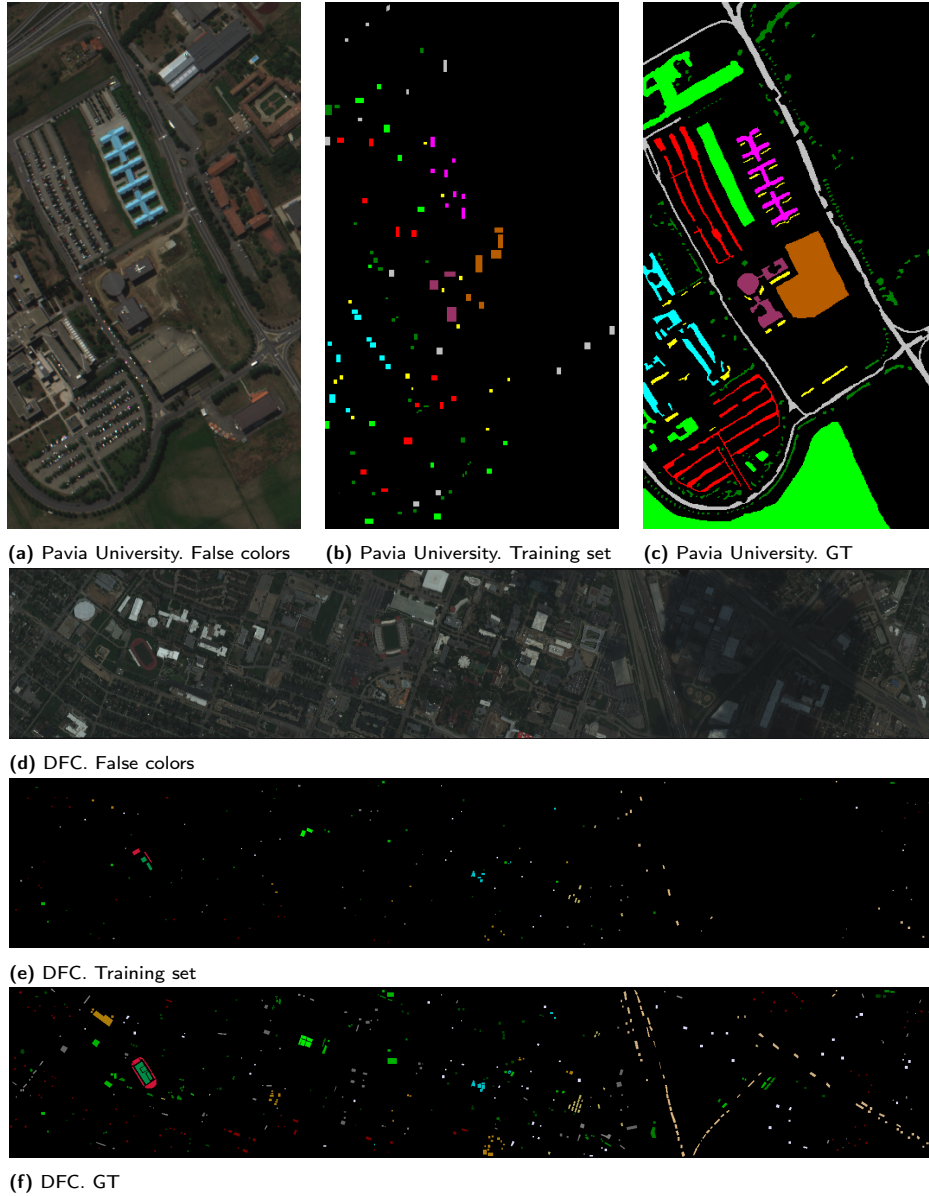


Fig. 7: Pavia University HSI (a-c), and DFC HSI (d-f) in false colors with their training sets and ground-truths (GT)

6.1.3 Some Experimental Parameters

In the following, to avoid repetition, we describe some common parameters used in the morphological methods.

Concerning the choice of the parameters of the morphological and attribute transformations, they were set such that enough distinctive variations or features can be observed in the profiles.

As for the CPD, in some of the following experiments, we would like to see the effect of the number of iterations and the choice of the rank on the results. Mainly, the number of iterations is set to 50, and is decided by looking at the plot of the reconstruction error of the decomposition with respect to the number of iterations. However, some experiments were carried out with 30 and 50 iterations. As for the value of the rank, first, we note that it sets the value of the reduced dimension of the feature space. Second, in most of the experiments, we start with a value that is very close to the number of predefined classes in the groundtruth image, and the other values are chosen to keep the results comparable to those of TPCA in terms of dimension

of the feature space. Note that the expression $\text{CPD}_{(i,R)}$ refers to a CPD carried out with i iterations and rank R .

TPCA is one of the methods that we compare our results to. In this regard, the number of principal components of the modes are decided based partly on the elbow rule of the graphs of the singular values obtained from the SVD of the corresponding mode-unfolding of the tensor, and partly according to how much the reduced mode dimensions can explain the original data. Note that the expression $\text{TPCA}_{(pc_3, pc_4)}$ refers to a TPCA carried out with the dimensions of the third and the fourth modes reduced to pc_3 and pc_4 components respectively.

In the classification phase, Support Vector Machine (SVM) with a Gaussian kernel was used. The hyper-parameters of SVM were optimized using 5-fold cross-validation as mentioned in the guide of [19]. The training and testing sets that are available with the two data-sets were used to train and test the classifier.

6.2 Experiments with Pavia University

6.2.1 EMP

First, we start by showing the results of the EMP method. In this part, we use Opening and Closing by reconstruction as the operators ($\Phi = \gamma_{SE}^\infty, \underline{\Phi} = \phi_{SE}^\infty$). In the following, we fix the parameters of the transformations to $m = 6$ disk-shaped structuring elements of different sizes, in pixels: $\{1, 6, 11, 16, 21, 26\}$. This means that the fourth mode of the tensor is of dimension $K = 13$ with the arrangement shown in (2). The first two modes are then merged to give tensor $\mathcal{Y} \in \mathbb{R}^{207400 \times 103 \times 13}$.

Before applying CP decomposition, since the data is huge, we attempt to preprocess the tensor by compressing its spatial mode as explained in section 4.2. After the compression step, we obtain a tensor $\mathcal{G} \in \mathbb{R}^{1339 \times 103 \times 13}$, which is then the input to be directly decomposed through CP decomposition.

Now, we compute the CP decomposition with nonnegativity constraints as mentioned in the previous sections. We also use different values of the rank in order to see the variations in the results such that $R = \{10, 20, 30, 40\}$. The factor matrices are initialized randomly as the absolute value of the i.i.d standard Gaussian distribution, and only the first factor matrix is compressed afterwards.

For TPCA, [four to six](#) principal components were chosen for the third-mode tensor unfolding explaining the data by 98.97%, 99.28%, and 99.47% respectively, and [four to six](#) components were chosen for the fourth-mode tensor unfolding explaining the data by 98.95%, 99.37%, and 99.57% respectively. As a result, this maps to [nine](#) different classification results noting that the total number of features is the product of the number of the principal components chosen from the latter [two](#) modes.

Discussion. In table 1, we show some tests with their reconstruction errors. Indeed, we notice that the more the iterations or the higher the rank, the less the reconstruction error, and generally, the better the classification accuracy. Increasing the number of iterations allows AO-ADMM to converge further, but at some point there will be only slight changes. Increasing the rank allows for more degrees of freedom for the data to be spread in the decomposition, but first we need the rank to be relatively small, second we notice that eventually there will be some kind of a limit for how much the reconstruction error and the classification accuracy can improve, and third if the rank is high then noisy structures may start to appear in the decomposition.

In addition to the influence of the number of iterations and the rank, nonnegative CPD was able to bring better accuracy results starting at only 10 features compared to 25 features for TPCA. Figures 9a-9e show some classification maps for the EMP setup, with an indication on the methods used and their parameters.

Method	Features	Rec. Error %	OA %	AA %
EMP + TPCA _(5,5)	25	-	93.20	91.16
EMP + CPD _(30,10)	10	6.19	91.82	91.18
EMP + CPD _(50,10)	10	6.16	<u>93.25</u>	<u>91.58</u>
EMP + CPD _(30,20)	20	4.20	<u>93.24</u>	<u>92.33</u>
EMP + CPD _(50,20)	20	4.06	<u>93.99</u>	<u>92.36</u>
EMP + CPD _(30,30)	30	3.18	<u>94.34</u>	<u>93.54</u>
EMP + CPD _(50,30)	30	3.05	<u>95.63</u>	<u>95.41</u>
EMP + CPD _(30,40)	40	2.68	<u>97.73</u>	<u>97.71</u>
EMP + CPD _(50,40)	40	2.59	<u>97.68</u>	<u>97.71</u>

Tab. 1: Pavia University. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMP set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The CPD cases are grouped by the same value of the rank. The cases when CPD does better are underlined.

6.2.2 PAMD

Now we show some of the results using the PAMD method. Similarly to EMP, Opening and Closing were chosen as the morphological operators with the same type of the structuring element and same size values. At the end we obtain a tensor $\mathcal{Y} \in \mathbb{R}^{207400 \times 103 \times 13}$, which is then compressed into $\mathcal{G} \in \mathbb{R}^{1339 \times 103 \times 13}$.

We use the same parameters of the nonnegative CPD as those in the EMP setup. For TPCA, **three to five** principal components were chosen for the third-mode tensor unfolding explaining the data by 99.67%, 99.75%, and 99.82% respectively, and **four to six** components were chosen for the fourth-mode tensor unfolding explaining the data by 97.72%, 98.65%, and 99.47% respectively.

Discussion. We show some of the results in table 2. We notice the same pattern of reconstruction error and classification accuracy results as those found in table 1 for increasing number of iterations and values of the rank though the reconstruction errors are higher in table 2. Nonnegative CPD was able to bring better accuracy results starting at 30 features compared to 30 features for TPCA. Figures 9f-9j show some classification maps for the PAMD setup, with an indication on the methods used and their parameters.

Method	Features	Rec. Error %	OA %	AA %
PAMD + TPCA _(5,6)	30	-	92.23	90.94
PAMD + CPD _(30,10)	10	18.02	87.21	88.46
PAMD + CPD _(50,10)	10	17.98	87.77	88.28
PAMD + CPD _(30,20)	20	11.21	91.35	88.78
PAMD + CPD _(50,20)	20	11.10	92.16	89.77
PAMD + CPD _(30,30)	30	8.76	<u>94.52</u>	<u>91.77</u>
PAMD + CPD _(50,30)	30	8.71	<u>94.59</u>	<u>91.72</u>
PAMD + CPD _(30,40)	40	7.24	<u>96.96</u>	<u>96.44</u>
PAMD + CPD _(50,40)	40	7.23	<u>96.91</u>	<u>96.49</u>

Tab. 2: Pavia University. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the PAMD set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The CPD cases are grouped by the same value of the rank. The cases when CPD does better are underlined.

6.2.3 EMAP

In this setting, we start by showing the **four** types of attributes and their corresponding threshold values that we used. **Some attribute thresholds depend on the size of objects in the image, and others depend on the range of values in the pixels, so they were chosen based on observed changes between transformations:**

- Area of the regions; $\lambda_a = \{100, 500, 1000, 5000\}$.
- Diagonal of the bounding box; $\lambda_d = \{10, 25, 50, 100\}$.
- Standard deviation; $\lambda_s = \{20, 30, 40, 50\}$.
- Moment of inertia; $\lambda_i = \{0.2, 0.3, 0.4, 0.5\}$.

For each attribute, we fix **four** different thresholds, corresponding to **eight** thinnings and thickenings, which means that the fourth mode is of dimension $K = 9$ (including the original image). After we consider the different attributes together in one data block, the fifth mode becomes of dimension $L = 4$. After merging the first two modes, we obtain a tensor $\mathcal{Y} \in \mathbb{R}^{207400 \times 103 \times 9 \times 4}$.

We notice that, practically, compressing or decomposing \mathcal{Y} can be computationally very demanding, so we tend to reduce the spectral dimension in the original HSI, \mathcal{I} . In order to do that and conserve both the nonnegativity and the information in \mathcal{I} , we use Nonnegative Matrix Factorization (NMF) on its first matrix unfolding, call it \mathbf{I} , where one mode represents pixels arranged in lexicographic order, and the other represents spectral bands. NMF decomposes \mathbf{I} into **two** other matrices with nonnegative entries, e.g. $\mathbf{I} = \mathbf{W}\mathbf{H}^T$, where \mathbf{W} represents pixel information, and the number of columns in \mathbf{W} and \mathbf{H} is defined by the rank of the NMF, call it R_{NMF} , which is usually relatively small. \mathbf{W} is then chosen and rearranged into a data cube, \mathcal{W} , with reduced spectral dimension, to be used in order to form \mathcal{Y} . Suppose that we note by R_{NMF} as the size of the reduced spectra, then in the case of Pavia HSI, \mathcal{Y} is of dimensions $207400 \times R_{NMF} \times 9 \times 4$. Now, following what we do with EMP and PAMD, \mathcal{Y} is decomposed using compressed nonnegative CPD. The steps are visualized in a flowchart as seen in figure 8.

In a similar manner, in the case of TPCA, PCA of \mathcal{I} was computed as means of reducing the dimension of the spectral mode before forming the EMAP on the principal components. After that, \mathcal{Y} is formed. The PCA and EMAP part was done in [12].

For NMF, we choose a value 40 of the rank which corresponds to a reconstruction error of 0.65% compared to the original HSI. For PCA, we choose **four** principal components which explains the data by 99.16%. For CPD, we choose one value of the rank, which is $R = 40$. For TPCA, **four** principal components were chosen for the third-mode tensor unfolding explaining the HSI by 100%, **four** components were chosen for the fourth-mode unfolding explaining the data by 99.28%, and **three** components were chosen for the fifth-mode unfolding explaining the data by 99.98%.

Discussion. Table 3 shows some results related to EMAP. It is important to note that in some cases, the data was rescaled to fit the threshold values of the attribute profiles such that we see enough distinctive features between the images of the attribute profiles. With a lower number of 40 features compared to 48 for TPCA, nonnegative CPD showed better accuracy results. This shows that the method can still bring good results when going from an order-4 tensor to a matrix using nonnegative CPD. Figures 9l and 9m show some corresponding classification maps.

6.3 Experiments with DFC Image

6.3.1 EMP

In the case of the DFC image, the set-up and the parameters are almost the same as those of Pavia University. Only the differences are mentioned in the following.

The structuring elements are disk-shaped of different sizes, in pixels: $\{2, 7, 12, 17, 22, 27\}$. This also means that the fourth mode of the tensor is of dimension $K = 13$ with the arrangement shown in (2). After merging the first two modes, we obtain the tensor $\mathcal{Y} \in \mathbb{R}^{664845 \times 144 \times 13}$. After that we obtain the compressed tensor $\mathcal{G} \in \mathbb{R}^{1872 \times 144 \times 13}$.

Method	Features	Rec. Error %	OA %	AA %
PCA ₍₄₎	4	-	68.30	80.28
NMF ₍₄₀₎	40	0.65	77.01	87.65
EMAP	144	-	91.59	92.92
EMAP + PCA ₍₁₅₎	15	-	94.46	94.12
EMAP + TPCA _(4,4,3)	48	-	95.44	94.86
EMAP + CPD _(50,40)	40	2.03	97.06	98.60

Tab. 3: Pavia University. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMAP set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The cases when CPD does better are underlined.

Method	EMP %		PAMD %		EMAP %	
	TPCA _(5,5)	CPD _(50,40)	TPCA _(5,6)	CPD _(50,40)	TPCA _(4,4,3)	CPD _(50,40)
Asphalt	94.21	98.00	96.58	98.63	94.59	98.36
Meadow	94.77	97.11	93.05	97.47	97.60	94.88
Gravel	58.4	94.32	59.55	89.58	77.85	98.84
Tree	99.17	98.14	99.17	98.83	98.04	97.35
Metal Sheet	99.46	99.55	99.73	99.64	99.28	99.46
Bare Soil	89.78	99.71	84.44	92.34	88.01	99.80
Bitumen	99.08	99.28	99.49	99.59	99.79	99.69
Brick	98.84	98.42	98.15	98.12	99.22	99.25
Shadow	86.79	94.84	88.30	94.21	99.37	99.74
OA	93.20	97.68	92.23	96.91	95.44	97.06
AA	91.16	97.71	90.94	96.49	94.86	98.6

Tab. 4: Pavia University. Some per-class accuracy records including those of Overall and Average Accuracies for the EMP, PAMD, and EMAP set-ups.

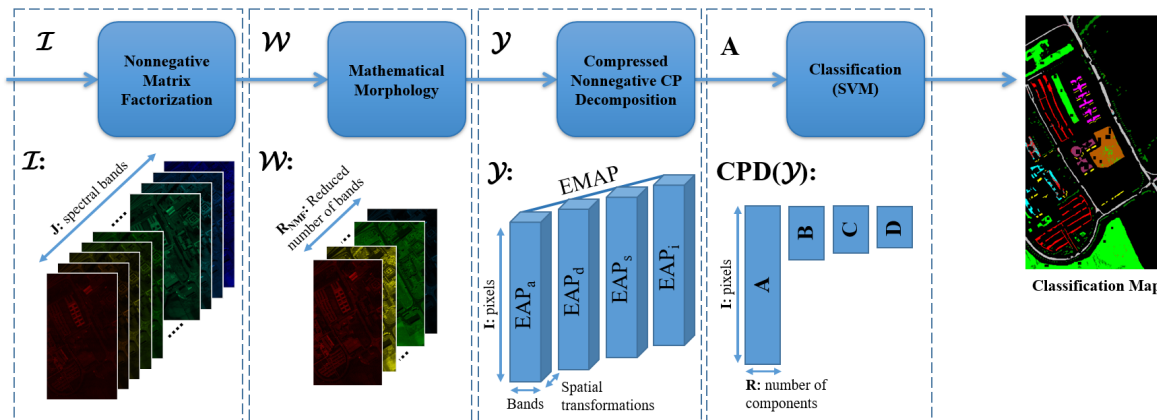


Fig. 8: A flow chart of the EMAP procedure which includes a preprocessing step to reduce the number of spectral bands.

As for the nonnegative CPD, we set the values of the rank to $R = \{15, 20, 30, 40\}$. The factor matrices and the parameters of the CPD are initialized similarly to the case in Pavia University. For TPCA, three to five principal components were chosen for the third-mode tensor unfolding explaining the data by 99.60%, 99.72%, and 99.81% respectively, and four to six components were chosen for the fourth-mode tensor unfolding explaining the data by 99.28%, 99.52%, and 99.69% respectively.

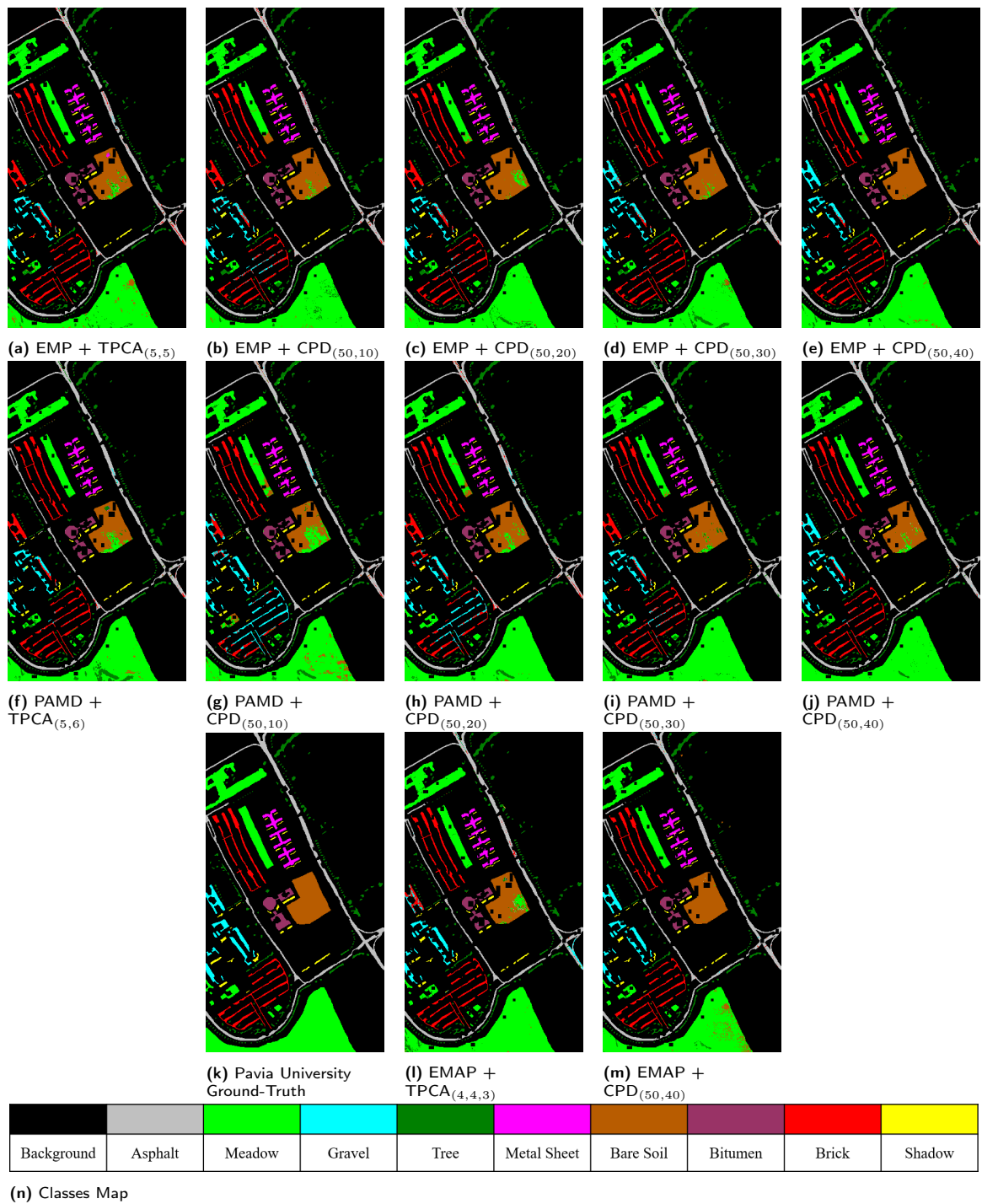


Fig. 9: Classification maps for Pavia University HSI.

Discussion. In table 5, we show some of the results with their reconstruction errors. In this case, we only show the changes in the rank of the CPD, where we notice that as the value of the rank goes higher, the reconstruction error improves, while the classification accuracy doesn't follow the same pattern as before even though it exceeds the values given by TPCA in all cases. The best two overall accuracy values obtained by TPCA are shown in the table in order to keep it consistent with table 6. Additionally, CPD could bring better accuracy results starting at 15 features compared to 30 features for TPCA. Figures 10b, 10d, 10f, 10h, and 10j show the corresponding classification maps.

Method	Features	Rec. Error %	OA %	AA %
EMP + TPCA _(4,5)	20	-	83.20	85.64
EMP + TPCA _(5,6)	30	-	83.38	85.93
EMP + CPD _(50,15)	15	4.40	<u>84.97</u>	<u>87.04</u>
EMP + CPD _(50,20)	20	3.35	<u>84.78</u>	<u>86.87</u>
EMP + CPD _(50,30)	30	2.35	<u>84.70</u>	<u>86.80</u>
EMP + CPD _(50,40)	40	2.01	<u>84.79</u>	<u>86.97</u>

Tab. 5: DFC. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMP set-up. The features column indicates the size of the feature space in the classification. The best two values given by TPCA are shown. The cases when CPD does better are underlined.

6.3.2 PAMD

In the case of PAMD, for TPCA, [three to five](#) principal components were chosen for the third-mode tensor unfolding explaining the data by 99.76%, 99.83%, and 99.87% respectively, and [four to six](#) components were chosen for the fourth-mode tensor unfolding explaining the data by 95.86%, 98.04%, and 99.21% respectively.

Discussion. In table 6, we show some of the results. In this case, we notice that as the value of the rank goes higher, both the reconstruction error and the classification accuracy improve. Since the best classification accuracy given by TPCA showed a better classification than the ones given by CPD, we included the second best as well. In this case, the classification accuracy results given by CPD are around the range of those given by TPCA. Figures 10c, 10e, 10g, 10i, and 10k show the corresponding classification maps.

Method	Features	Rec. Error %	OA %	AA %
PAMD + TPCA _(5,6)	30	-	83.76	86.10
PAMD + TPCA _(3,6)	18	-	<u>84.55</u>	<u>86.83</u>
PAMD + CPD _(50,15)	15	15.05	80.79	83.65
PAMD + CPD _(50,20)	20	10.77	81.16	83.93
PAMD + CPD _(50,30)	30	7.68	82.46	85.16
PAMD + CPD _(50,40)	40	5.94	<u>84.07</u>	<u>86.49</u>

Tab. 6: DFC. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the PAMD set-up. The features column indicates the size of the feature space in the classification. The best two values given by TPCA are shown. The cases when CPD does better are underlined.

6.3.3 EMAP

Here, we used the same [four](#) types of attributes, [where the choice of the threshold values was also based on observed changes between transformations](#):

- Area of the region; $\lambda_a = \{300, 2000, 6000, 9000\}$
- Diagonal of the bounding box; $\lambda_d = \{20, 45, 75, 150\}$
- Standard deviation; $\lambda_s = \{20, 30, 40, 50\}$
- Moment of inertia; $\lambda_i = \{0.2, 0.3, 0.4, 0.5\}$

Consequently, we obtain a tensor $\mathcal{Y} \in \mathbb{R}^{664845 \times 144 \times 9 \times 4}$.

Following the same steps taken in the EMAP part of Pavia's HSI, and considering the larger block of data in the case of DFC, NMF is performed with a rank of 40, chosen with a reconstruction error of 0.48%. As for the PCA part of the original image, [five](#) principal components were chosen, explaining the original HSI with 99.87%.

As for the CPD, we choose a rank of 40. For TPCA, **three** principal components were chosen for the third-mode tensor unfolding explaining the HSI by 99.15%, **six** components were chosen for the fourth-mode unfolding explaining the data by 99.68%, and **three** components were chosen for the fifth-mode unfolding explaining the data by 99.97%.

Table 7 shows some of the results. CPD showed a better classification accuracy with 40 features compared to 54 features in the case of TPCA. Figures 10l and 10m show the corresponding classification maps.

Method	Features	Rec. Error %	OA %	AA %
PCA₍₅₎	5	-	74.12	77.22
NMF₍₄₀₎	40	0.48	78.83	81.82
EMAP	180	-	83.39	85.78
EMAP + PCA₍₁₅₎	15	-	82.28	84.96
EMAP + TPCA_(3,6,3)	54	-	81.32	84.30
EMAP + CPD_(50,40)	40	1.95	83.19	85.76

Tab. 7: DFC. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMAP set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The cases when CPD does better are underlined.

Method	EMP %		PAMD %		EMAP %	
	TPCA _(5,6)	CPD _(50,15)	TPCA _(3,6)	CPD _(50,40)	TPCA _(3,6,3)	CPD _(50,40)
Grass Healthy	81.86	79.58	80.43	79.58	82.24	81.76
Grass Stressed	82.80	84.02	85.05	98.21	82.70	84.30
Grass Synthetic	100	99.60	100	100	100	100
Tree	84.28	90.81	79.45	96.11	87.87	79.92
Soil	98.95	98.67	100	99.52	97.44	99.14
Water	95.10	95.10	95.10	95.10	95.10	95.10
Residential	88.71	79.47	90.67	95.33	78.26	80.69
Commercial	71.03	83.09	70.65	43.96	41.40	75.21
Road	81.01	85.08	82.62	84.79	85.64	75.16
Highway	68.33	68.43	68.24	68.53	68.05	67.56
Railway	82.35	83.87	82.54	84.81	83.20	85.76
Parking Lot 1	73.87	79.73	87.22	69.83	82.99	81.94
Parking lot 2	80.70	78.59	80.70	82.45	79.64	80.35
Tennis Court	100	100	100	100	100	100
Running Track	100	99.57	99.78	99.15	100	99.57
OA	83.38	84.97	84.55	84.07	81.32	83.19
AA	85.93	87.04	86.83	86.49	84.30	85.76

Tab. 8: DFC. Some per-class accuracy records including those of Overall and Average Accuracies for the EMP and PAMD set-ups.

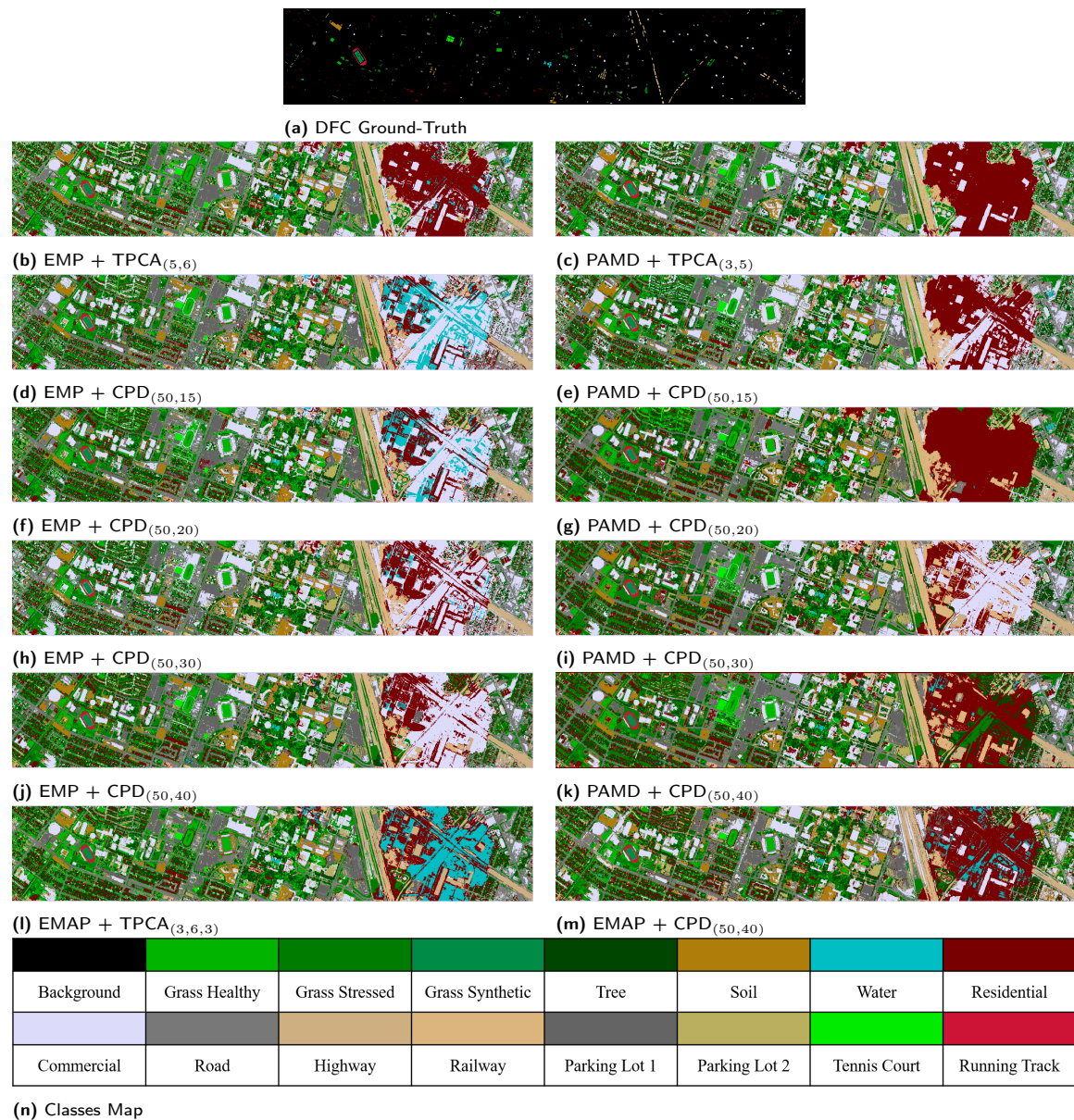


Fig. 10: Classification maps for DFC HSI.

6.4 Further Discussion

6.4.1 Unconstrained CPD

The emphasis in this work was on Nonnegative CPD, which is more intuitive when it comes to physical interpretation of the data. However, we think that classifying the data is still possible without the nonnegativity constraints. The advantage in this case is that the decomposition without constraints is faster than that with constraints.

Considering that CPD without constraints is another way of rewriting the data in a tensor decomposition format, it was worth showing some of the results even though they might have no physical interpretation. In this case, the original tensor is still compressed and the compressed version becomes the input to AO-ADMM. Since there are no constraints on the uncompressed factors of the decomposition, AO-ADMM becomes pretty much the same as computing Alternating Least Squares. We computed 20 realizations of unconstrained CPD

for each value of the rank, $R = \{10, 20, 30, 40\}$ in the case of Pavia University and $R = \{15, 20, 30, 40\}$ in the case of DFC. We note that, practically, computing the CPD yields some variations between the factor matrices that are obtained from different initializations, which are drawn randomly as described in the previous section (except that in the case of unconstrained CPD, we don't force the initializations to be in the positive quadrant).

First, we show, for each of the setups, the overall accuracy and reconstruction error results of 20 realizations in the box plots of figures 11 and 12 representing the HSI of Pavia University and DFC respectively, followed by the tables 9 and 10 respectively that show specific values of overall accuracy from the box plots; the lower adjacent, median and upper adjacent of the boxes, and the overall accuracy corresponding to the minimal reconstruction error among the 20 realizations. We notice in most of the cases a general increase in the overall accuracy and a likewise decrease in the reconstruction error as the rank increases, and in most of the cases the results are comparable to, or higher than, those found in tables 1, 2, 5 and 6 especially with higher values of the rank. The only exception to the latter is in the AMD setup of the DFC image where the values of the overall accuracy fluctuate a bit, which could be caused by a bad modeling of the classifier since cross-validation doesn't always find the most optimized solution for the hyper-parameters of SVM, but by looking at table 10, we notice that the range of the values is still comparable to a state-of-the-art method such as TPCA.

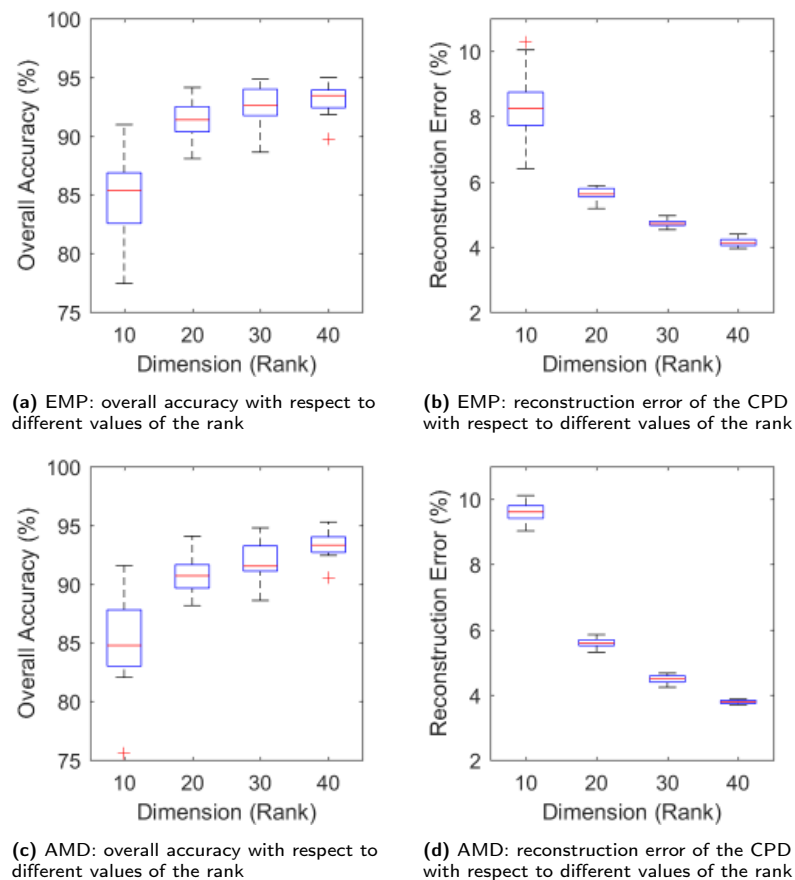


Fig. 11: Pavia University. Box plots of overall accuracy (left) and reconstruction error (right) with respect to the rank of the CPD. Figures 11a-11b correspond to EMP and figures 11c-11d correspond to AMD. Each box represents 20 decompositions, each carried with a different initialization along with the corresponding value of the rank. Remarkable values of overall accuracy are shown in table 9.

EMP	$R = 10$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	77.42	88.06	88.61	91.82
Median	85.34	91.38	92.62	93.43
Upper Adjacent	90.96	94.13	94.83	94.98
Minimal Rec. Error	85.25	94.13	92.60	94.42
AMD	$R = 10$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	82.03	88.16	88.59	92.43
Median	84.76	90.70	91.51	93.31
Upper Adjacent	91.56	94.07	94.78	95.28
Minimal Rec. Error	91.56	89.47	93.96	95.28

Tab. 9: Pavia University. Specific values of overall accuracy from the box plots of figure 11.

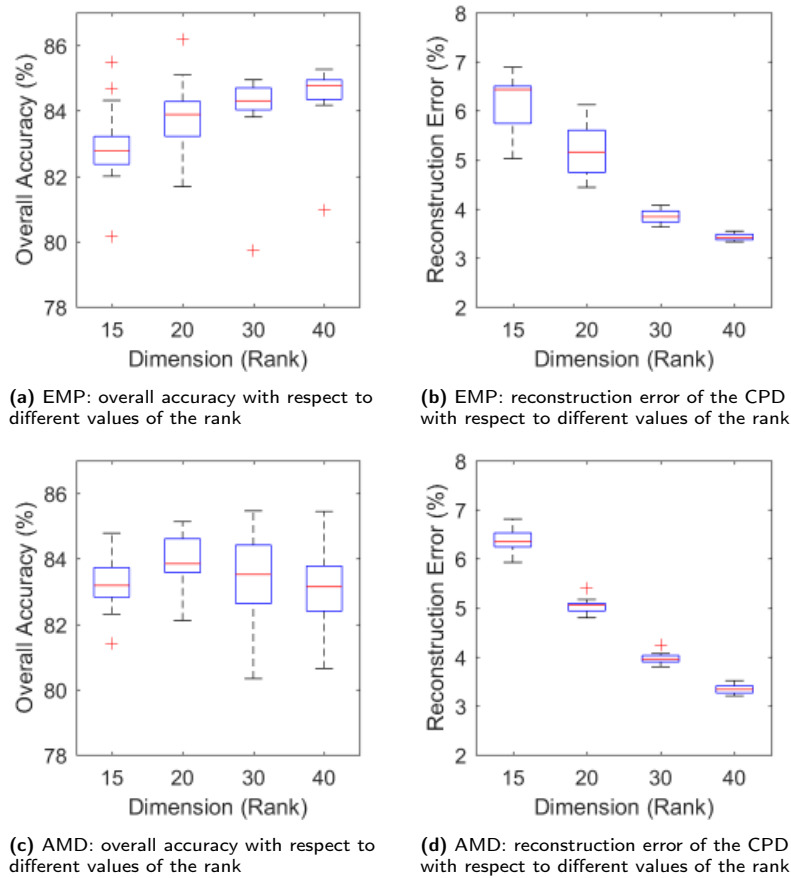


Fig. 12: DFC. Box plots of overall accuracy (left) and reconstruction error (right) with respect to the rank of the CPD. Figures 12a-12b correspond to EMP and figures 12c-12d correspond to AMD. Each box represents 20 decompositions, each carried with a different initialization along with the corresponding value of the rank. Remarkable values of overall accuracy are shown in table 10.

EMP	$R = 15$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	81.99	81.67	83.80	84.16
Median	82.77	83.88	84.30	84.75
Upper Adjacent	84.31	85.09	84.94	85.26
Minimal Rec. Error	85.49	83.95	84.16	84.16
AMD	$R = 15$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	82.30	82.11	80.33	80.46
Median	83.18	83.84	83.52	83.14
Upper Adjacent	84.78	85.13	85.46	85.44
Minimal Rec. Error	83.47	83.57	82.69	82.97

Tab. 10: DFC. Specific values of overall accuracy from the box plots of figure 12.

7 Conclusion

In this paper, CPD was proposed as a technique in the framework of spectral-spatial classification of hyperspectral data with spatial information being added using Mathematical Morphology as an application. In general, CPD provides an intuitive approach to deal with such multi-modal feature data where the results in the decomposition can be used for pixel-wise classification in a low dimensional feature space without loss of information. The focus here was on nonnegative CPD (with compression of the data), which further provides a better understanding of the results that are distributed over the decomposed factors, and where each factor describes one of the modes of the tensor in an interpretable way. Experiments on [two](#) datasets and [three](#) different morphological settings were carried out to explore the classification aspect of the proposed technique, compared to results obtained through TPCA as a base algorithm in the framework of tensor modeling. The effect of some parameters such as the number of iterations, the rank of the decomposition, the reconstruction error, and the use of constraints was explored. The classification results were promising to say that the data found in matrix \mathbf{A} can be seen as pixel data and hence classified.

This kind of application allows to go further. In the perspectives, one way would be to explore the potential unmixing aspect of using nonnegative CPD of hyperspectral data based on multi-modal feature study (spectral and spatial features for instance) in the framework of unsupervised classification or blind separation, which would highlight both the use of CPD compared to other decomposition methods that have been used, and that of nonnegativity constraints. However, there are some challenges for using the nonnegative compressed CPD. First, it may suffer from long-time execution due to per-iteration compression and decompression steps. Second, due to the iterative nature of its approximation, it may fall into local minima.

References

- [1] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. Sousa. Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing*, 9(11):1110, 2017.
- [2] H. Akbari, K. Uto, Y. Kosugi, K. Kojima, and N. Tanaka. Cancer detection using infrared hyperspectral imaging. *Cancer science*, 102(4):852–857, 2011.
- [3] E. Aptoula, M. Dalla Mura, and S. Lefèvre. Vector attribute profiles for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3208–3220, 2016.
- [4] J. Benediktsson, J. Palmason, and J. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, 2005.
- [5] J. Benediktsson, M. Pesaresi, and K. Amason. Classification and feature extraction for remote sensing

- images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9):1940–1949, 2003.
- [6] J. A. Benediktsson and P. Ghamisi. *Spectral-spatial classification of hyperspectral remote sensing images*. Artech House, 2015.
- [7] G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351–1362, 2005.
- [8] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.
- [9] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6):2094–2107, 2014.
- [10] J. Cohen, R. Farias, and P. Comon. Fast decomposition of large nonnegative tensors. *IEEE Signal Processing Letters*, 22(7):862–866, 2015.
- [11] P. Comon. Tensors: a brief introduction. *IEEE Sig. Proc. Magazine*, 31(3):44–53, May 2014. hal-00923279.
- [12] M. Dalla Mura, J. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762, 2010.
- [13] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute filters for the analysis of very high resolution remote sensing images. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 3, pages III–97. IEEE, 2009.
- [14] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *International Journal of Remote Sensing*, 31(22):5975–5991, 2010.
- [15] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [16] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2012.
- [17] L. Gao and R. T. Smith. Optical hyperspectral imaging in microscopy and spectroscopy—a review of data acquisition. *Journal of biophotonics*, 8(6):441–456, 2015.
- [18] Y. Gu, T. Liu, and J. Li. Superpixel tensor model for spatial-spectral classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.
- [19] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. *A practical guide to support vector classification*, 2003.
- [20] K. Huang, N. Sidiropoulos, and A. Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, 2016.
- [21] J. R. Jensen. *Remote sensing of the environment: An earth resource perspective 2/e*. Pearson Education India, 2009.
- [22] I. Jolliffe. *Principal component analysis*. Springer, 2011.
- [23] M. Jouni, M. Dalla Mura, and P. Comon. Classification of hyperspectral images as tensors using nonnegative cp decomposition. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 189–201. Springer, 2019.
- [24] M. Jouni, M. Dalla Mura, and P. Comon. Hyperspectral image classification using tensor CP decomposition. In *2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2019, Yokohama, Japan, July 28 - August 2, 2019*, pages 1164–1167, 2019.
- [25] J. B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions. *Linear Algebra and Applications*, 18:95–138, 1977.
- [26] P. R. Marpu, M. Pedergnana, M. Dalla Mura, S. Peeters, J. A. Benediktsson, and L. Bruzzone. Classification of hyperspectral data using extended attribute profiles based on supervised and unsupervised feature extraction techniques. *International Journal of Image and Data Fusion*, 3(3):269–298, 2012.
- [27] S. Murchie, R. Arvidson, P. Bedini, K. Beisser, J.-P. Bibring, J. Bishop, J. Boldt, P. Cavender, T. Choo, R. Clancy, et al. Compact reconnaissance imaging spectrometer for mars (crism) on mars reconnaissance orbiter (mro). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

- [28] L. Najman and H. Talbot. *Mathematical morphology: from theory to applications*. John Wiley & Sons, 2013.
- [29] C. Pilorget and J.-P. Bibring. Nir reflectance hyperspectral microscopy for planetary science: Application to the micromega instrument. *Planetary and Space Science*, 76:42–52, 2013.
- [30] Y. Qi, P. Comon, and L. H. Lim. Uniqueness of nonnegative tensor approximations. *IEEE Trans. Inf. Theory*, 62(4):2170–2183, Apr. 2016. arXiv:1410.8129.
- [31] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surveying and Land Information Science*, 62(2):115–122, 2002.
- [32] J. Serra. *Image Analysis and Mathematical Morphology*. Number v. 1 in Image Analysis and Mathematical Morphology. Academic Press, 1984.
- [33] J. Serra. *Image analysis and mathematical morphology: Theoretical advances*. Image Analysis and Mathematical Morphology. Academic Press, 1988.
- [34] G. A. Shaw and H. K. Burke. Spectral imaging for remote sensing. *Lincoln laboratory journal*, 14(1):3–28, 2003.
- [35] P. Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.
- [36] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson. Svm-and mrf-based method for accurate classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 7(4):736–740, 2010.
- [37] D. Tuia, F. Pacifici, M. Kanevski, and W. J. Emery. Classification of very high spatial resolution imagery using mathematical morphology and support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 47(11):3866–3879, 2009.
- [38] M. A. Veganzones, J. E. Cohen, R. C. Farias, J. Chanussot, and P. Comon. Nonnegative tensor cp decomposition of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(5):2577–2588, 2016.
- [39] S. Velasco-Forero and J. Angulo. Classification of hyperspectral images by tensor modeling and additive morphological decomposition. *Pattern Recognition*, 46(2):566–577, 2013.
- [40] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE transactions on image processing*, 2(2):176–201, 1993.
- [41] Y. Xu, Z. Wu, J. Chanussot, P. Comon, and Z. Wei. Nonlocal coupled tensor cp decomposition for hyperspectral and multispectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.