






Tensor-based Factorization Algorithms for Pixel-wise Classification of Hyperspectral Data Using Deep Convolutional Networks

Josué López ^{1,*} , Deni Torres ¹ , Clement Atzberger ² , Andrea González ¹  and Israel Yañez ³ 

¹ Center for Research and Advanced Studies of the National Polytechnic Institute, Telecommunications Group, Av del Bosque 1145, Zapopan 45017, Mexico; deni.torres@cinvestav.mx; andrea.gonzalez@cinvestav.mx

² University of Natural Resources and Life Science, Institute of Geomatics, Peter Jordan 82, Vienna 1180, Austria; clement.atzberger@boku.ac.at

³ Polytechnic University Juventino Rosas, Networks and Telecommunications Ingengering Department, Miguel Hidalgo 102, Comunidad de Valencia 38253 Juventino Rosas, Guanajuato, Mexico; jyanezptc@upjr.edu.mx

* Correspondence: josue.lopez@cinvestav.mx

Version December 7, 2020 submitted to Remote Sens.

Abstract: Tensor-based decomposition for compression of high dimensionality datasets have been widely used in recent years in several research areas, including Multi- and Hyperspectral Images (MSI and HSI) processing. On the other hand, Convolutional Neural Networks (CNNs) are specialized kind of Artificial Neural Networks (ANNs) for processing data that has a known grid-like topology and belongs to the set of natural numbers, such as image data. Compression of the input data of a CNN induce poor performance in tasks, such as classification and semantic segmentation. In this paper, Tucker-based models are employed to reduce the dimensionality in the spectral domain of MSIs and HSIs to reduce the complexity of a pixel-wise classification CNN, while preserving high performance. We propose a framework, based on information theory, that performs a characterization of a spectral dataset, by computing the entropy and the probability distribution function of the spectral bands, as well as a quantification of orthogonality and divergence of the compressed data, to define the dimensionality of the CNN input tensor. Besides, we propose an alternative Tucker approximation with non-negativity and integer constraints called Integer Approximation Non-negative Tucker Decomposition (IANTD). Experimental results demonstrate...

Keywords: entropy; hyperspectral imagery; tensor decomposition

1. Introduction

Reducing the dimensionality of the input dataset for Artificial Intelligence (AI) algorithms has been one of the most active research areas in recent years []. The introduction of tensor-based models for these kind of tasks inspired a change in several areas, such as image processing. [].

Particularly, Remote Sensing (RS) image processing is focused in detecting and monitoring physical features about areas of interest by measuring its reflectance. Generally, remote sensors use filters to separate the reflectance of an object in different wavelength ranges []. Hence the concept spectral images. Multi- (MSI) and Hyperspectral Images (HSI) lead to high performace in image processing tasks such as detection, classification and segmentation []. Besides, in the last few years the use of spectral data has grown exponentially in other fields as medical analysis [], biomedical [], and, in RS fields as agriculture [],natural disaster prediction [], security affairs [], among others [].

In the last decade, many supervised classification and segmentation algorithms were developed, with the aim of taking advantage of the spatial and spectral data features, provided by RS MSI and

HSI. Support Vector Machines (SVM) [], k-Nearest Neighbors (k-NN) [] and Convolutional Neural Networks (CNN) [] are examples of the aforementioned. Spectral image processing in artificial intelligence algorithms increase drastically the execution time []. The foregoing requires having robust computer equipment to achieve time competitive results.

Several works opted for matrix factorization algorithms to reduce the high-dimensionality of spectral images []. More recently, the development of tensor-based factorization algorithms [] show advantages over those based on matrices []. Nevertheless, changing the domain of the input data of a machine learning model could lead to a drop in its performance []. Other authors developed dimensionality reduction strategies by measuring the saliency of the spectral bands []. This approach has the advantage of preserving the original domain of MSIs and HSIs.

This work proposes a framework, based on information theory, that characterize MSI and HSI, by computing the entropy and the probability distribution function of the spectral bands. Besides, degree of orthogonality and Jensen Shannon divergence of the compressed data is computed, to reduce the number of bands of the CNN input tensor. On the other hand, we propose an Integer Approximation Non-negative Tucker Decomposition (IANTD), alternative to TKD and NTD but with non-negativity and integer constraints.

1.1. Previous works

In recent years, several works have developed frameworks to reduce computational load of machine learning algorithms []. Specially, for hyperspectral dataset classification and segmentation through Deep Learning (DL) ANN []. The crucial factor addressed in this work is, to achieve compression of the input data to reduce the high number of computations, without decreasing performance classification.

Before the introduction of tensor decomposition algorithms, the way to use hyperspectral images as input for supervised classification algorithms was by band selection [23] and [22]. Later, matrix decomposition methods were used, such as PCA in [?], and even non-negative matrix decomposition methods [?]. In 2015 Zhang et al. [24] were pioneers in experimenting with multilinear algebra-based decompositions on hyperspectral images.

On the other hand, instead of HSI, MSIs was a good alternative due to the small number of spectral bands, which also produce competitive classification performance [11], [18], [21] and [?]. However, the need to increase performance forces researchers to use data with larger number of spectral bands, which alleviate the classification of classes difficult to differentiate [26], [27], [29], [?] and [?].

Recently, Sayeh [?] published a work close to our research. They proposed a non-negative tensor decomposition of hyperspectral images but, different to the framework proposed in this work, they try to preserve certain spatial-spectral features into the so called abundance maps, i.e. the projection matrices, while the framework proposed in this work preserve the architecture of the image by compressing the spectral domain in the non-negative core tensor.

Table 1 summarizes some of the most cited related papers, which deal with the compression-classification issue.

Table 1. Related work in spectral imagery semantic segmentation.

Reference	Input	Decomposition	Reduction	Classifier
Li, S. et al. [23] (2014)	HSI	-	Band selection	SVM
Zhang, L. et al. [24] (2015)	HSI	TKD	Spatial-Spectral	-
Wan, Q. et al. [22] (2016)	HSI	-	Band selection	SVM/kNN/CART
Kemker, R. et al. [11] (2017)	MSI	-	-	CNN
Tong L. et al. [] (2017)	HSI	NMF	Unmixing	-
Hamida, A. et al. [21] (2017)	MSI	-	-	CNN
Chien, J. et al. [] (2017)	RGB	TFNN	Spatial-Spectral	TFNN
Dewa, M. et al. [] (2018)	HSI	PCA	Spectral	PCA
Xu, Y. et al. [] (2018)	HSI	-	-	CNN
Li, J. et al. [28] (2019)	MSI	NTD-CNN	Spatial-spectral	-
An, J. et al. [27] (2019)	HSI	T-MLRD	Spatial-spectral	SVM/1NN
An, J. et al. [29] (2019)	HSI	TDA	Spatial-spectral	SVM/1NN
Lopez, J. et al. [] (2020)	MSI	TKD	Spectral	FCN
Sayeh, M. [] (2019)	HSI	NTD	Spatial-Spectral	3D-CNN
Our framework	MSI/HSI	iNTD/NTD	Spectral	CNN

1.2. Motivation

Some data compression strategies have favorably reduced MSIs and HSIs dimensionality. Decomposition methods based on matrix and tensor approaches have been applied as pre-processing of ANNs input. In tensor decompositions, data processing in its original format, i.e., N-th order tensors, improves the factorization because it considers the correlation of the data in its different modes. It is also important to note that the inappropriate selection of some decomposition parameters could lead to information loss and would penalize the performance of a CNN.

With the aim of keeping high classification performance after a tensor-based decomposition, this work presents a strategy, based on information theory metrics, to reduce the computational complexity of CNNs for spectral image pixel-wise classification.

1.3. Contribution

Unlike previous works, this work seeks to adapt the data in a more efficient way to the DL-ANNs input. CNN models are designed to extract and interpret all the spatial properties of an image by moving the kernels over the input data []. Therefore, producing uncorrelated data in space and spectrum, would make harder the interpretation of the data in the convolutional network [?]. Thus, we address the problem of setting the n-rank of the MSI and HSI Tucker-based decomposition, specifically the 3rd-rank, by an entropy and Jensen-Shannon divergence analysis. In addition, we proposed an approximation that preserves the integer and non-negativity tensor original format of the spectral images, as well as the spatial dimensionality.

The main contributions of this work can be summarized by the following three points:

1. This work proposes a semi-empirical strategy, based on information theory, for defining the 3rd-rank of compression models based on the TKD.
2. Furthermore, an integer non-negative Tucker-based approximation, called IANTD, was developed to improve performance of pixel-wise classification CNNs. This approximation is restricted to preserve the spatial domain while compressing the spectral domain and, in turn, decreasing computational load.
3. This work also presents an exhaustive performance evaluation analysis measuring and comparing performance by metrics as Pixel Accuracy (PA) in function of the number of new tensor bands, F1-score, Mathews Correlation Coefficient (MCC), Kohen's Kappa coefficient, orthogonality degree of the factor matrices, as well as the core tensor, reconstruction error, and execution time.

The remainder of this work is organized as follows. Section 2 introduces tensor algebra notation and basic concepts to familiarize the reader with the symbology used in this paper. Section 3 describes the TKD model and its non-negative constrained version. Section 4 presents the problem statement of this work and the mathematical definition. In Section 5 it is described the framework proposed for compression and pixel-wise classification of spectral images. Experimental results are presented in Section 6. Finally, Sections 7 and 8 present the discussions, comparisons and conclusions based on the results obtained in the experiments.

2. Notation and definitions

Matrix-based factorizations, such as SVD [], and dimensionality reduction approaches as PCA [] have been significant and useful tools for data compression and other approaches. Nevertheless, they are limited to data representations in 2-dimensional spaces. Most of current applications have data structures often as higher-order arrays, e.g. dimensions of space, time, and frequency. This 2-way view in matrix factorizations may be inadequate and it is natural to use tensor decomposition approaches [?].

A tensor can be defined as a multi-way or multidimensional array. The order of a tensor is the number of dimensions, also known as modes, i.e., an N -order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is an N -dimensional array, which elements x_{i_1, i_2, \dots, i_n} are indexed by $i_n \in 1, 2, \dots, I_n$ for $1 \leq n \leq N$.

Throughout this paper, the mathematical notation used by Kolda et al. [17] has been adopted. Table 2 summarize this notation.

It is also necessary to introduce some tensor algebra operations and basic concepts used in later explanations.

2.1. Matricization

The mode- n matricization is the process of reordering the elements of a tensor into a matrix along axis n and it is denoted as $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$.

2.2. Inner Product

The inner product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is the sum of the products of their entries, i.e., $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} a_{i_1 \dots i_N} b_{i_1 \dots i_N}$.

2.3. N -Mode Product

It means the multiplication of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ or vector $\mathbf{u} \in \mathbb{R}^{I_n}$ in mode n , i.e., along axis n . It is represented by $\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$, where $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N}$ [17].

2.4. Rank-One Tensor

A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is rank one if it can be written as the outer product of N vectors, i.e.,

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \dots \circ \mathbf{a}^{(N)} \quad (1)$$

where \circ denotes the outer product and $\mathbf{a}^{(n)}$ denotes a vector in a sequence of N vectors. Each element of the tensor is the product of the corresponding vector elements, i.e.,

$$x_{i_1 i_2 \dots i_N} = a_{i_1}^{(1)} \dots a_{i_N}^{(N)}.$$

2.4.1. N -Rank

The n -rank of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ denoted $\text{rank}_n(\mathcal{X})$, is the column rank of $\mathbf{X}_{(n)}$, i.e., the dimension of the vector space spanned by the mode- n fibers. Hence, if $R_n \equiv \text{rank}_n(\mathcal{X})$ for $n = 1, \dots, N$, then \mathcal{X} has a rank- (R_1, \dots, R_N) tensor [17].

Table 2. Tensor algebra notation summary

$\mathcal{A}, \mathbf{A}, \mathbf{a}, a$	Tensor, matrix, vector and scalar respectively
$\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$	N -order tensor of size $I_1 \times \dots \times I_N$.
$a_{i_1 \dots i_N}$	An element of a tensor
$\mathbf{a}_{:i_2 i_3}, \mathbf{a}_{i_1 : i_3},$ and $\mathbf{a}_{i_1 i_2 :}$	Column, row and tube fibers of the third order tensor \mathcal{A}
$\mathbf{A}_{i_1 ::}, \mathbf{A}_{:i_2 :}, \mathbf{A}_{::i_3}$	Horizontal, lateral and frontal slices of the third order tensor \mathcal{A}
$\mathbf{A}^{(n)}, \mathbf{a}^{(n)}$	A matrix/vector element from a sequence of matrices/vectors
$\mathbf{A}_{(n)}$	Mode- n matricization of a tensor. $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$
$\mathbf{a}^{(1)} \circ \dots \circ \mathbf{a}^{(N)}$	Outer product of N vectors
$\langle \mathcal{A}, \mathcal{B} \rangle$	Inner product of two tensors.
$\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$	n -mode product of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ along axis n .

3. Tensor decompositions (TDs)

As an extension of the SVD [], two main specific tensor decompositions can be considered, Tucker Decomposition (TKD) [] and CANDECOMP/PARAFAC (CP) []. There are many other tensor decompositions, INDSCAL, PARAFAC2, CANDELINC, DEDICOM, PARATUCK2, among others [17]. Furthermore, there are also nonnegative variants of all of the above. With the aim of preserving particular characteristics of hyperspectral images for pixel-wise classification, this study is limited to use decompositions based on the Tucker model.

3.1. Tucker Decomposition (TKD)

The TKD [17], for the particular case of third-order tensors, can be formally formulated as follows [?]. Given a third-order data tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and three positive indices J_1, J_2 and J_3 , find a core tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ and three component matrices called factor matrices $\mathbf{U}^1 \in \mathbb{R}^{I_1 \times J_1}$, $\mathbf{U}^2 \in \mathbb{R}^{I_2 \times J_2}$ and $\mathbf{U}^3 \in \mathbb{R}^{I_3 \times J_3}$ which perform the following approximate decomposition:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} + \mathcal{E} \quad (2)$$

where \mathcal{E} denotes the approximation error tensor. The core tensor \mathcal{G} preserves the level of interaction for each factor or projection matrix $\mathbf{U}^{(n)}$. The factor matrices are commonly considered orthogonal, but in Tucker models with non-negativity constraints, that is not necessarily imposed [?]. These matrices can be seen as the principal components in each mode [17] (see Figure 1). J_n represents the number of components in the decomposition, i.e., the rank – (R_1, R_2, R_3) .

The TKD can also be denoted by the matricization approach and expressed as

$$\mathbf{X}_{(1)} = \mathbf{U}^{(1)} \mathbf{G}_{(1)} (\mathbf{U}^{(3)} \otimes \mathbf{U}^{(2)})^T \quad (3a)$$

$$\mathbf{X}_{(2)} = \mathbf{U}^{(2)} \mathbf{G}_{(2)} (\mathbf{U}^{(3)} \otimes \mathbf{U}^{(1)})^T \quad (3b)$$

$$\mathbf{X}_{(3)} = \mathbf{U}^{(3)} \mathbf{G}_{(3)} (\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})^T \quad (3c)$$

where \otimes denotes the Kronecker product and $\mathbf{X}_{(n)}$ and $\mathbf{G}_{(n)}$ are the n -mode matricized versions of tensor \mathcal{X} and \mathcal{G} respectively.

Starting from (2), the reconstruction of an approximated tensor can be given by

$$\hat{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \quad (4)$$

where $\hat{\mathcal{X}}$ is the reconstructed tensor. Then, the core tensor \mathcal{G} can be acquired by the multilinear projection

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times_3 \mathbf{U}^{(3)T} \quad (5)$$

161 where $\mathbf{U}^{(n)T}$ denotes the transpose matrix of $\mathbf{U}^{(n)}$ for
 162 $n = 1, \dots, N$. The reconstruction error ξ can be computed as

$$\xi(\hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 \quad (6)$$

163 and $\|\cdot\|_F$ represents the Frobenius norm. To compute the best 3-rank approximation of a tensor, it can
 164 be used an iterative algorithm as ALS, HALS, HOOI after a HOSVD initialization [?].

165 HOOI initializes the factors matrices using HOSVD and assumes that othogonal matrices are
 166 known, so that the core tensor is obtained with (5). Then, it maximizes the cost function

$$\max_{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}} \|\mathbf{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times_3 \mathbf{U}^{(3)T}\|_F^2 \quad (7)$$

167 with $\mathbf{U}^{(n)}$ unknown. Fixing all factor matrices but one, tensor \mathbf{X} can be projected onto the
 168 $\{R_1, \dots, R_{n-1}, R_{n+1}, \dots, R_N\}$ -dimensional space as

$$\mathbf{W}^{(-n)} = \mathbf{X} \times_1 \mathbf{U}^{(1)T} \dots \times_{n-1} \mathbf{U}^{(n-1)T} \times_{n+1} \mathbf{U}^{(n+1)T} \dots \times_N \mathbf{U}^{(N)T} \quad (8)$$

169 and the orthogonal matrices can be estimated as an orthonormal basis for the dominant subspace of
 170 the projection by applying the standard matrix SVD for n -mode unfolded matrix $\mathbf{W}_{(n)}^{(-n)}$ for $n = 1, 2, 3$
 171 [?].

172 3.1.1. Non-negative Tucker Decomposition (NTD)

173 The NTD is a decomposition based on the Tucker model. It is a new tensor factorization method
 174 with nonnegativity constraints []. For the third-order case, the NTD, as defined by Cichocky [15], can
 175 be formulated as follows. Given a third-order tensor $\mathbf{X} \in \mathbb{R}_+^{I_1 \times I_2 \times I_3}$ find a core tensor $\mathbf{G} \in \mathbb{R}_+^{J_1 \times J_2 \times J_3}$
 176 and the factor matrices $\mathbf{U}_1 \in \mathbb{R}_+^{I_1 \times J_1}$, $\mathbf{U}_2 \in \mathbb{R}_+^{I_2 \times J_2}$ and $\mathbf{U}_3 \in \mathbb{R}_+^{I_3 \times J_3}$ which performs the approximation
 177 given in Eq. (2). As well as for the TKD model, the best 3-rank approximation of a nonnegative tensor
 178 can be computed by an iterative algotihm as HOOI, maximizing the cost function given in equation 7.
 179 Algorithm 1 shows the HOOI algorithm for a NTD.

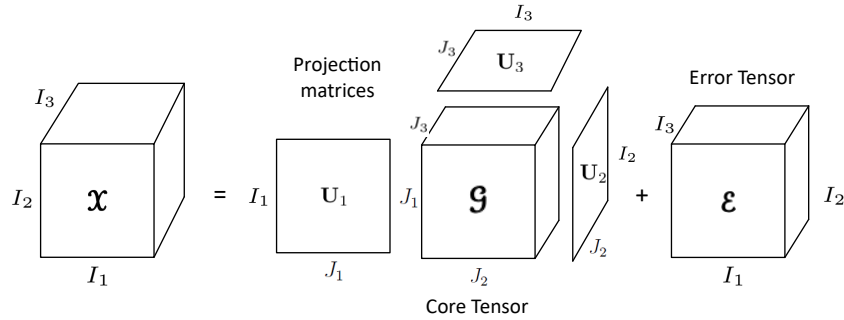


Figure 1. Tucker decomposition for a third-order tensor.

Algorithm 1: HOOI algorithm to compute a rank- (R_1, \dots, R_N) NTD for an N th-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$.

Function HOOI ($\mathcal{X}, J_1, \dots, J_N$):
 initialize $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ for $n = 1, \dots, N$ using HOSVD or random
repeat
 for $n = 1, \dots, N$ **do**
 $\mathcal{W}^{(-n)} \leftarrow \mathcal{X} \times_{-n} \{\mathbf{U}^{(T)}\}$
 $[\mathbf{U}^{(n)}, \boldsymbol{\Sigma}^{(n)}, \mathbf{V}^{(n)}] \leftarrow \text{svds}(\mathcal{W}_{(n)}^{(-n)}, J_n, \text{'LM'})$
 $\mathbf{U}^{(n)} \leftarrow [\mathbf{U}^{(n)}]_+$
 end
until fit ceases to improve or maximum iterations exhausted;
 $\mathcal{G} \leftarrow \mathcal{W}^{(-N)} \times_N \mathbf{U}^{(N)T}$
Output: $\mathbf{U}^{(n)} \in \mathbb{R}_+^{I_n \times J_n}, \mathcal{G} \in \mathbb{R}_+^{J_1 \times J_2 \times \dots \times J_N}$

4. Problem phenomenology

4.1. Spectral Imagery

Multi- or Hyper-spectral images are by nature multidimensional integer nonnegative arrays. A spectral image can be sorted and represented as a third-order tensor $\mathcal{X} \in \mathbb{N}^{I_1 \times I_2 \times I_3}$, where \mathbb{N} denotes the space of natural numbers, I_1 , I_2 and I_3 represent the height, width and spectral bands respectively. In RS image processing, spectral images are frequently used for classification of different material in a scene of interest. However, due to the low spatial resolution produced by the distance between the sensor and the target, spatial features are not sufficient to discern certain classes. That is why spectral resolution plays an important role in this type of task.

The separation into spectral bands allows perception of reflectance at different wavelengths. This helps to better characterize various materials, in order to simplify the process of discernment between classes. The effort to obtain these spectral features generates a greater amount of data, which increases the processing complexity. This is where the spectral decomposition task becomes relevant.

4.2. Problem Statement

Let $\mathcal{X} \in \mathbb{N}^{I_1 \times I_2 \times I_3}$ be a spectral image represented as a third-order tensor, and $\mathbf{Y} \in \mathbb{N}^{I_1 \times I_2}$ its corresponding ground truth matrix for a specific number of classes C . Find the $\text{rank}_3(\mathcal{X})$ of a Tucker-based approximation, by analyzing information and statistical properties of the spectral data, to produce a core tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times J_3}$ used as the input of a pixel-wise classification CNN. The output is the prediction matrix $\hat{\mathbf{Y}}$ intended to achieve competitive pixel-wise classification performance while decreasing computational load in the classification process.

4.3. Mathematical Definition

The problem statement described above can be mathematically defined as the following optimization problem

$$\begin{aligned}
& \min_{\mathbf{g}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}} \|\mathbf{X} - \mathbf{g} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}\|_F^2 \\
& \text{subject to} \quad \mathbf{U}^{(n)} \in \mathbb{R}_+^{I_n \times J_n} \quad \text{for } n = 1, 2, 3 \quad \text{and} \quad \mathbf{g} \in \mathbb{R}_+^{J_1 \times J_2 \times J_3} \\
& \quad J_1 = I_1, J_2 = I_2 \quad \text{no compression in the spatial domain,} \\
& \quad J_3 < I_3 \quad \text{reduced spectral domain at the core tensor,} \\
& \quad \zeta(\hat{\mathbf{X}}) \leq \zeta_s \quad \text{measure of representativity} \\
& \quad D(\mathbf{g}_{j_3}) - D(\mathbf{g}_{j_3+1}) < D_s \quad \text{divergence stop criterion}
\end{aligned} \tag{9}$$

5. Methodology

The following subsections describe the methodology of the framework proposed in this work. The big picture is summarized in three steps: the HSI modeling, the tensor decomposition and the classification process.

5.1. HSI modeling

Consider an input dataset $\mathbf{X} \in \mathbb{N}^{I_1 \times I_2 \times I_3}$ with $I_1 \times I_2 \times I_3$ samples in the set of the natural numbers \mathbb{N} , where a fiber $\mathbf{x}_{i_1 i_2}$ represents the spectral signature of pixel i_1, i_2 and can be represented by the Linear Mixing Model (LMM) as follows

$$\mathbf{x}_{i_1 i_2} = \alpha_{i_1 i_2} \mathbf{M} + \eta_{i_1 i_2} \tag{10}$$

where $\alpha_{i_1 i_2}$ represents the abundance vector at pixel $i_1 i_2$, \mathbf{M} denotes the endmember matrix, and $\eta_{i_1 i_2}$ represents an additive noise vector. The abundance vectors $\alpha_{i_1 i_2}$ must always satisfy two constraints, i) the non-negativity, $\alpha_{i_1 i_2 c} \geq 0$ for all $c = 1, \dots, C$, and ii) the sum-to-one restriction, $\sum_{c=1}^C \alpha_{i_1 i_2 c} = 1$.

5.2. Tensor factorization

Consider $\mathbf{Y} \in \mathbb{C}^{I_1 \times I_2}$ as the matrix of actual classes corresponding to our dataset \mathbf{X} , and $\hat{\mathbf{Y}} \in \mathbb{C}^{I_1 \times I_2}$ as the prediction matrix, where \mathbb{C} defines the set of C different classes. In order to reduce data dimensionality of the input dataset \mathbf{X} while keeping classifier performance, we propose to use a restricted Tucker-based decomposition, to transform the input image into a core tensor $\mathbf{g} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ and n factors matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, denoted by

$$\mathbf{X} \xrightarrow{\mathcal{T}} (\mathbf{g}, \mathbf{U}^{(n)}) \tag{11}$$

where the decomposition \mathcal{T} is restricted to preserve the spatial domain and to reduce only the 3rd-mode domain by the Tucker-1 model, mathematically expressed as

$$\mathbf{X} = \mathbf{g} \times_1 \mathbf{I} \times_2 \mathbf{I} \times_3 \mathbf{U}^{(3)} \tag{12}$$

Hence, each fiber of the core tensor $\mathbf{g}_{i_1 i_2}$ takes a new representation in the tensor bands domain and can be mathematically defined as follows

$$\mathbf{g}_{i_1 i_2} = \beta_{i_1 i_2} \mathbf{S} + \eta_{i_1 i_2} \tag{13}$$

where $\beta_{i_1 i_2}$ is the contribution of each material at pixel i_1, i_2 and \mathbf{S} denotes the endmember matrix in the new tensor bands domain.

We also propose an approximation based on the NTD, which computes an integer decomposition, the Integer Non-negative Tucker decomposition (INTD). The INTD follows the same Tucker model described in Section 3.1. It considers the additional restriction of decomposing a tensor in the set of the natural numbers.

5.3. Classifier

The tensor decompositions based on the Tucker1 model produce a core tensor, where the first tensor bands provide a signature enough to differentiate the classes of interest of the input dataset. Then, the core tensor $\mathcal{G} \in \mathbb{N}^{I_1 \times I_2 \times J_3}$, with $J_3 < I_3$, and its corresponding ground truth \mathbf{Y} form the input tuple of the classifier Θ , which produce a predicted label for each element of the input, i.e.,

$$(\mathcal{G}, \mathbf{Y}) \xrightarrow{\Theta} \hat{\mathbf{Y}} \quad (14)$$

The performance of our classification model can be measured by the cross-entropy loss, whose output is a probability value. The cross-entropy loss increases as the predicted probability diverges from the actual label and it is computed as

$$J(\mathcal{W}) = -\mathbb{E}_{\mathcal{Y} \sim p} \log p(\mathbf{Y}|\mathcal{G}) \quad (15)$$

where $J(\mathcal{G})$ represents the loss function. For a multiclass probability distribution, the cross entropy cost function can be written as

$$H(y, p) = - \sum_{c=1}^C y_c \log(p_c) \quad (16)$$

where $H(y, p)$ denotes the cross entropy of targets y with a probability p .

The softmax function is used as the output of the classifier, to represent the probability distribution over C different classes. Formally, the softmax function is given by

$$\delta(\mathbf{z})_c = \frac{e^{z_c}}{\sum_{l=1}^L e^{z_l}} \quad (17)$$

where $\delta(\mathbf{z})_c$ denotes the softmax function of vector \mathbf{z} , which is each 3rd-mode fiber of the activation maps at the last convolutional layer. Hence, the softmax function produces a normalized probability distribution for every input pixel, which can be seen as the contribution parameter in the LMM (Eq. 10).

In this paper, we aim to feed supervised classifiers, based on 3D-CNN, with a lower dimensionality tensor than the original dataset. This has three particular motivations: 1) to avoid overfitting the DCNN, 2) to reduce the computational complexity, and 3) keep the classifier performance while reducing the execution time. Figure 2 shows the big picture of the framework proposed.

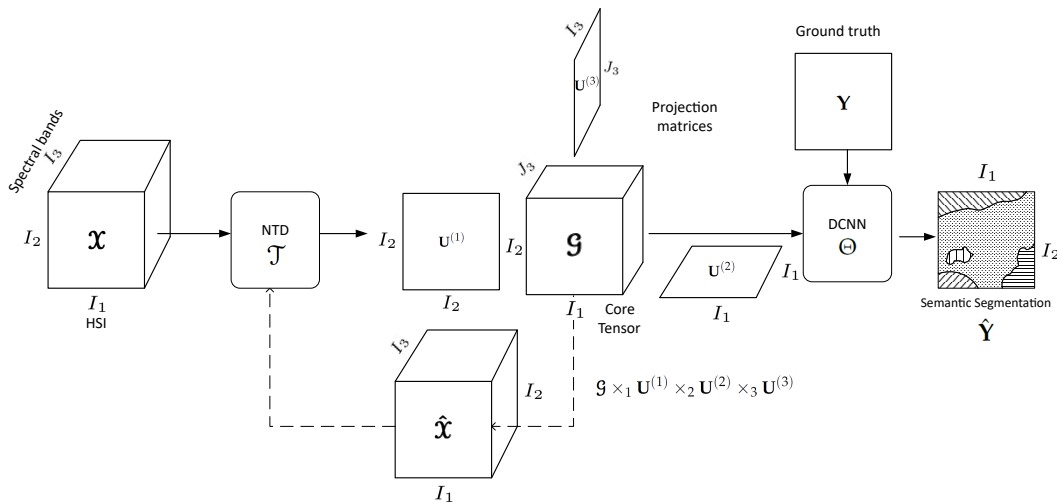


Figure 2. Big picture of the framework proposed.

6. Experimental Results

6.1. Algorithms metrics

6.1.1. Relative Mean Square Error (RMSE)

To compute the reconstruction error of any decomposition, it can be used the relative Mean Square Error, given by

$$\zeta(\hat{\mathbf{X}}) = \frac{\|\hat{\mathbf{X}} - \mathbf{X}\|_F^2}{\|\mathbf{X}\|_F^2}, \quad (18)$$

where \mathbf{X} represents the MSI and $\hat{\mathbf{X}}$ its corresponding reconstruction computed by (4).

6.1.2. Loss function

The error for the current state of an ANN must be computed each epoch as part of the optimization algorithm. Loss functions can be used to estimate the difference between the labels \mathbf{Y} and its corresponding prediction $\hat{\mathbf{Y}}$ of a model so that the weights are updated to reduce the error on the next evaluation. The Frobenius norm is one of the typically used functions to evaluate the loss of an ANN. For image classification it can be written as

$$J(\hat{\mathbf{Y}}) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \|y_{i_1, i_2} - \hat{y}_{i_1, i_2}\|_F^2 \quad (19)$$

where $J(\hat{\mathbf{Y}})$ denotes the loss of the classifier.

6.1.3. Jensen-Shannon Divergence

For this work the Jensen-Shannon divergence (JSD) is used as a metric to quantify how different the core tensor of a decomposition is from the input dataset from an information theory point of view. This method measures the difference between two probability distributions. The JSD can be computed as

$$D_{JS}(X\|G) = \frac{1}{2}D_{KL}(X\|M) + \frac{1}{2}D_{KL}(G\|M) \quad (20)$$

where $D_{JS}(X\|G)$ represents the JS divergence of the probability distributions X and G , $M = \frac{X+G}{2}$ is the mean of the probability distributions, and D_{KL} denotes the Kullback-Leibler divergence, which is a asymmetric version of the JSD and it is computed as

$$D_{KL}(X\|G) = \sum_{i=1}^I p(x_i) \log \frac{p(x_i)}{p(g_i)} \quad (21)$$

where $X(x_i)$ and $G(x_i)$ represent the probability of the i -th element at distributions X and G respectively.

6.1.4. Entropy

The divergence study is reinforced by an entropy analysis to quantify the uncertainty of each band from the original spectral image and from the core tensor of each decomposition. The entropy H of an image is computed as

$$H(\mathbf{X}) = - \sum_{i=1}^I p(x_i) \log_2 p(x_i) \quad (22)$$

6.2. Classification evaluation metrics

The datasets used for experiments in this work are considerable imbalanced. For this reason, we selected the metrics based on the works of Luque et al. [], Chicco et al. [] and Grandini et al. [], where they assess the impact of the imbalance and propose a set of metrics with lower bias in function of the imbalance and depending on the confusion matrix.

In order to fair evaluate performance of the classifier and to compare our work with others of the state of the art, we selected four main performance evaluation metrics. Pixel Accuracy (PA) is used to compute a ratio between the amount of correctly classified pixels and the total number of pixels. Despite this metric is highly biased for multiclass imbalanced dataset, it is one of the most popularly used in the state-of-art. Given a confusion matrix $\mathbf{M} \in \mathbb{N}^{C \times C}$ relating the True Positive (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN), PA is computed by Eq. 23. Cohen's Kappa coefficient and Matthews Correlation Coefficient (MCC) are alternative measures unaffected by the unbalanced datasets issue [?]. Kappa and MCC are computed by 24 and 25 respectively. Last, F1 score is employed as complementary metric to further weight the percentage of TP, since F1 is independent from TN (See Eq. 26).

Table 3. Table of metrics used to evaluate CNN classification performance.

Metric	Formula
Pixel Accuracy	$PA = \frac{TP + TN}{TP + TN + FP + FN} \quad (23)$
Cohen's Kappa Coefficient	$\kappa = \frac{\rho_o - \rho_e}{1 - \rho_e} \quad (24)$
Matthews Correlation Coefficient	$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (25)$
F1-score	$F_1 = \frac{2TP}{2TP + FP + FN} \quad (26)$

6.3. CNN Specifications

The model used to evaluate the framework proposed in this work is Segnet []. Table 4 shows the hyperparameters of the CNN set by cross-validation and the hardware specifications.

Table 4. Experiments' software and hardware specifications.

Hyperparameters	Software/Hardware
learning rate: 1×10^{-3} epochs: 100 optimizer: Adam [] initialization: Xavier [] kernel dimensions: 3×3 Activation Function: ReLU / Softmax	Platform: Python 3.7 AI Framework: Tensorflow 1.13 GPU: NVIDIA GeForce GTX 1050 Ti Processor: Intel core i7 RAM: 8GB SSD: 128GB / HDD: 1TB

6.4. Cases of study

For this work, one multispectral dataset and three popular hyperspectral dataset were selected. For a fair comparison with methodologies cited in Section 1.1, the dataset was processed with the original information from the European Space Agency Sentinel-2 database and from the [Hyperspectral Remote Sensing Scenes](#) web page.

Table 6 summarized the datasets used in this work as well as their spatial and spectral characteristics, the number of classes and their samples.

Table 5. Summary of the different dataset used for experiments in this work.

Dataset	Spatial dimensions	Bands	Classes	Samples
Sentinel-2 CNNMSI	128×128	9	5	1,802,240
Indian Pines	145×145	220	16	10,249
Salinas	512×217	224	16	53,785
Pavia University	610×340	103	9	40,076

6.4.1. Case A: Sentinel-2 dataset

This dataset proposed by Lopez et al. [?] is composed of 110 RS Sentinel-2 scenarios from central Europe. It has 100 scenarios as the training space and 10 scenarios for testing, all of them with 128×128 pixels with spatial resolution of $20m^2$ and 9 spectral bands in the range $490 - 2190nm$. The labels are semi-manually assigned for five classes of interest: vegetation, soil, water, clouds and shadows. Data are available in the link [Sentinel-2 Dataset](#).

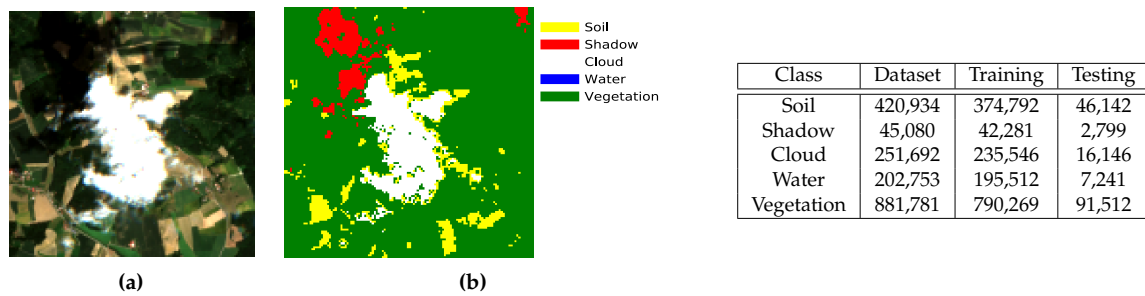


Figure 4 & Table 5. Sentinel-2 dataset a) True color image and b) Ground truth. Table) Samples per class.

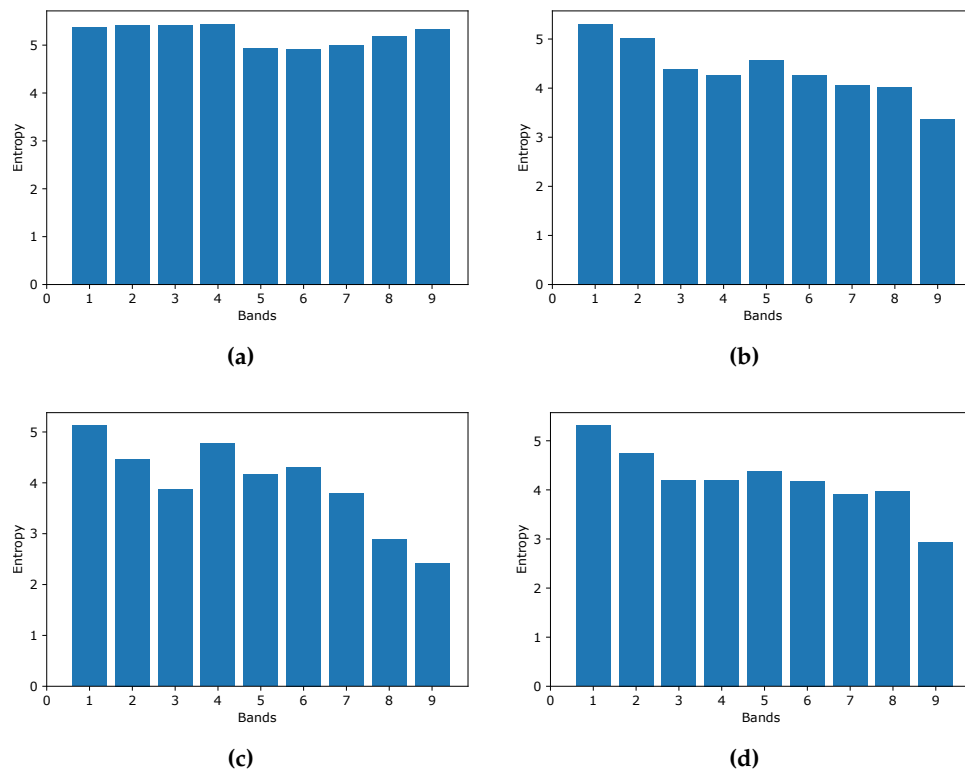


Figure 5. Sentinel-2 dataset entropy, with nbins = 54, computed using the Freedman–Diaconis' criterion. a) Original dataset, b) TKD, c) NTD, d) INTD.

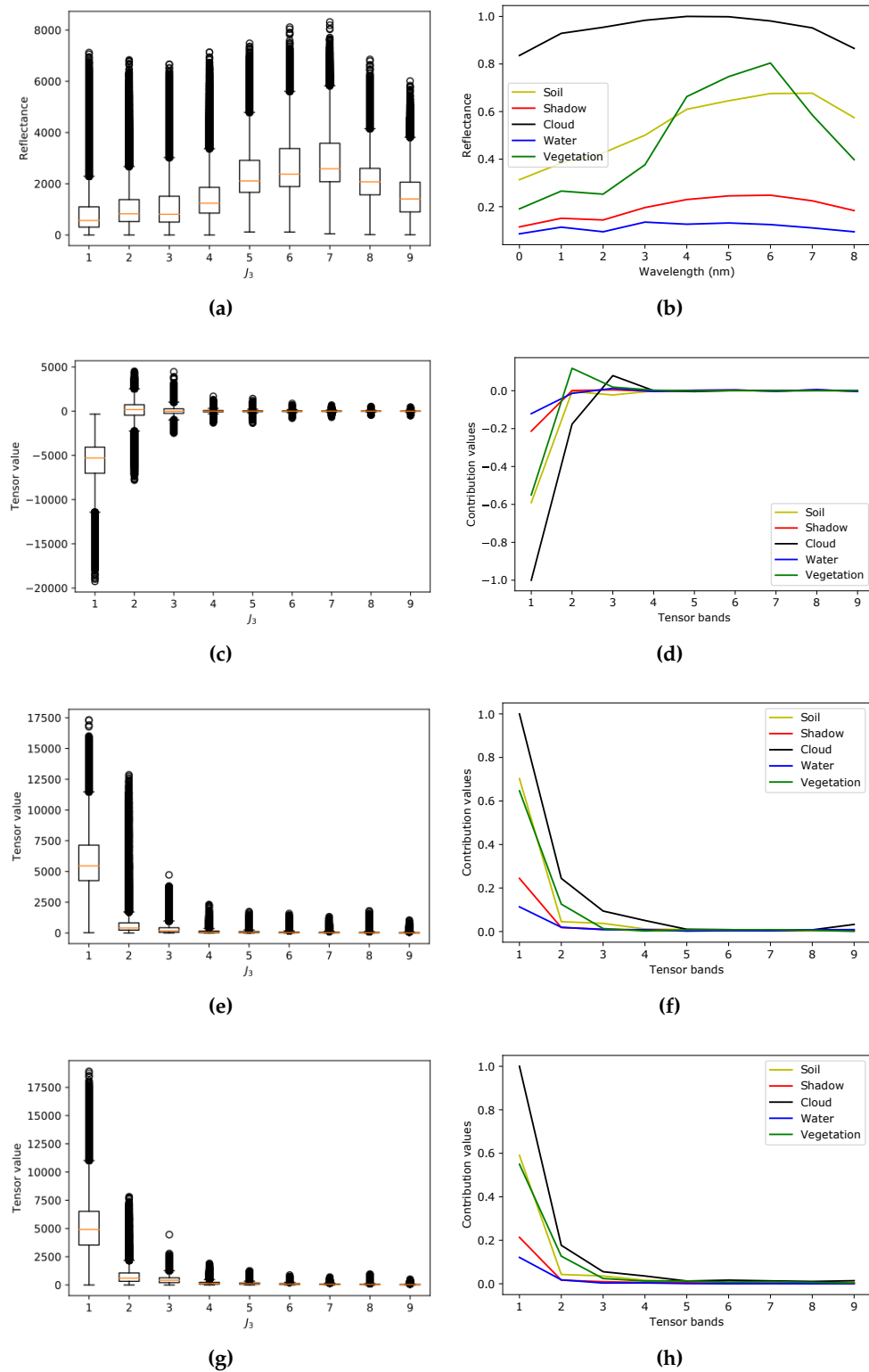


Figure 6. Box and whiskers plot and spectral signatures for a) and b) original spectral bands, c) and d) tensor bands of the NTKD and e) and f) tensor bands of the TKD

6.4.2. Case B: Indian Pines

This dataset is a scene produced by AVIRIS in North-western Indiana and consists of 145×145 pixels and 224 spectral bands in the wavelength range $0.4 - 2.5 \mu\text{m}$. The Indian Pines scene contains

two-thirds agriculture, and one-third forest or other natural perennial vegetation. There are two major dual lane highways, a rail line, as well as some low density housing, other built structures, and smaller roads. Since the scene is taken in June some of the crops present, corn, soybeans, are in early stages of growth with less than 5% coverage. The ground truth available is designated into sixteen classes and is not all mutually exclusive. Indian Pines data are available at [Indian Pines dataset](#). Figure 7a shows the true color, Figure 7b the ground truth and Table 4 shows the number of samples for each class.

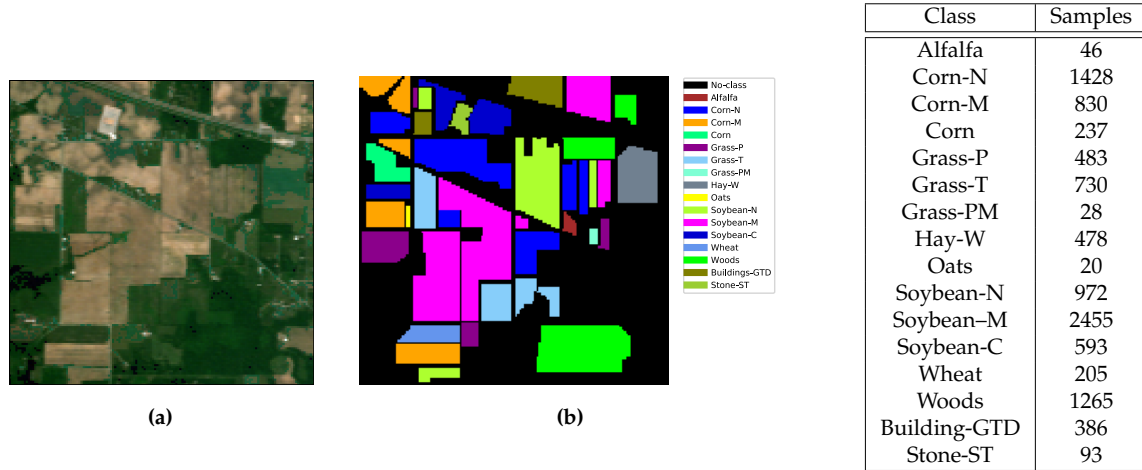


Figure 7 & Table 6. Indian Pines dataset, a) True color image and b) Ground truth. Table) Samples per class

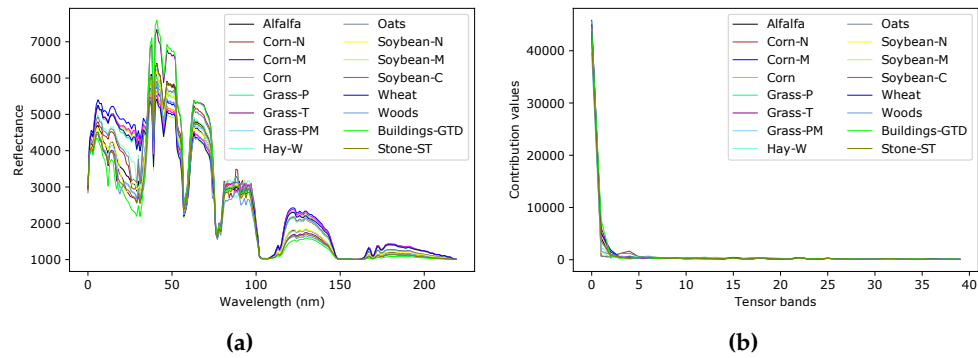


Figure 8. Behavior of the 16 classes of the Indian Pines dataset, a) in the spectral domain (spectral signatures) and b) in the the tensor bands domain after IANTD.

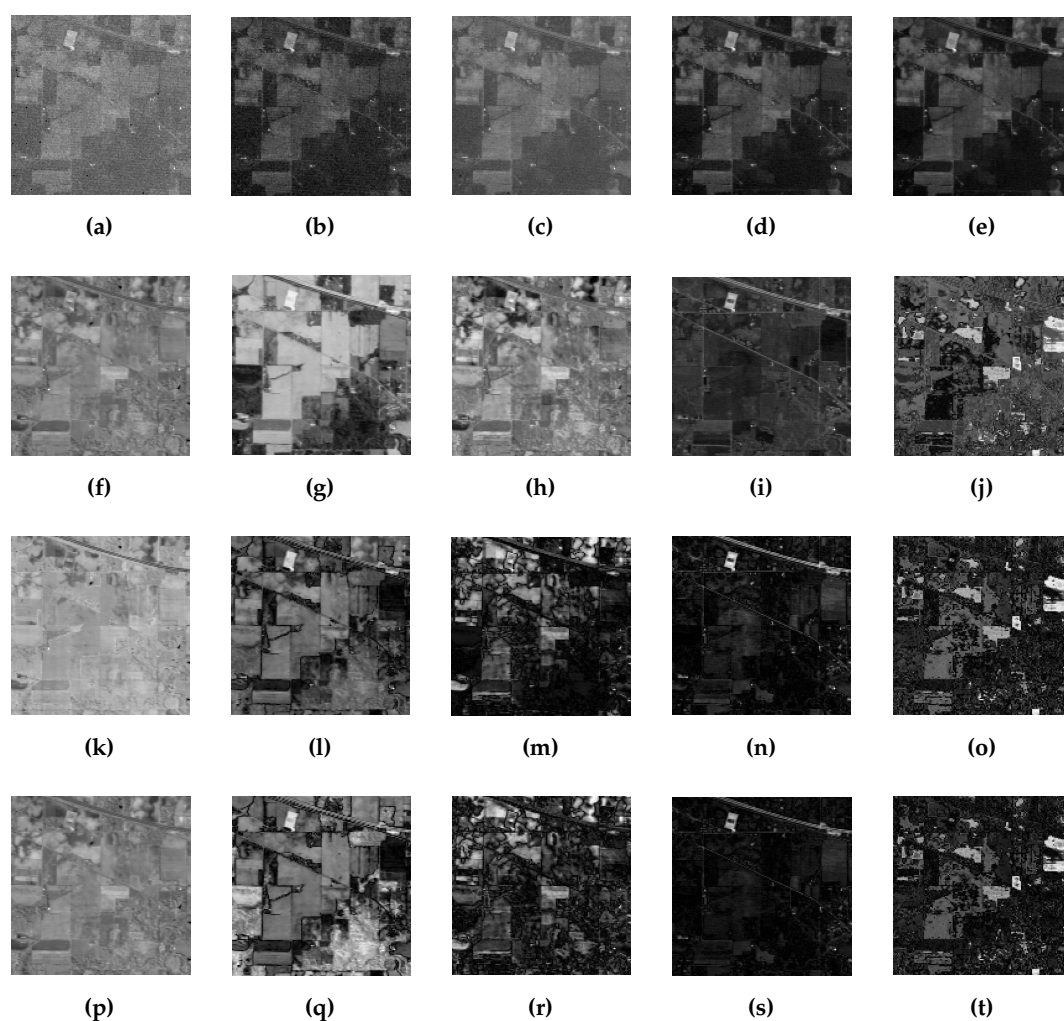


Figure 9. Visualisation of Indian Pines 1st to 5th [9a](#) to [9e](#) original spectral bands, [9f](#) to [9j](#) tensor bands with TKD, [9k](#) to [9o](#) tensor bands with NTD, and [9p](#) to [9t](#) tensor bands with INTD.

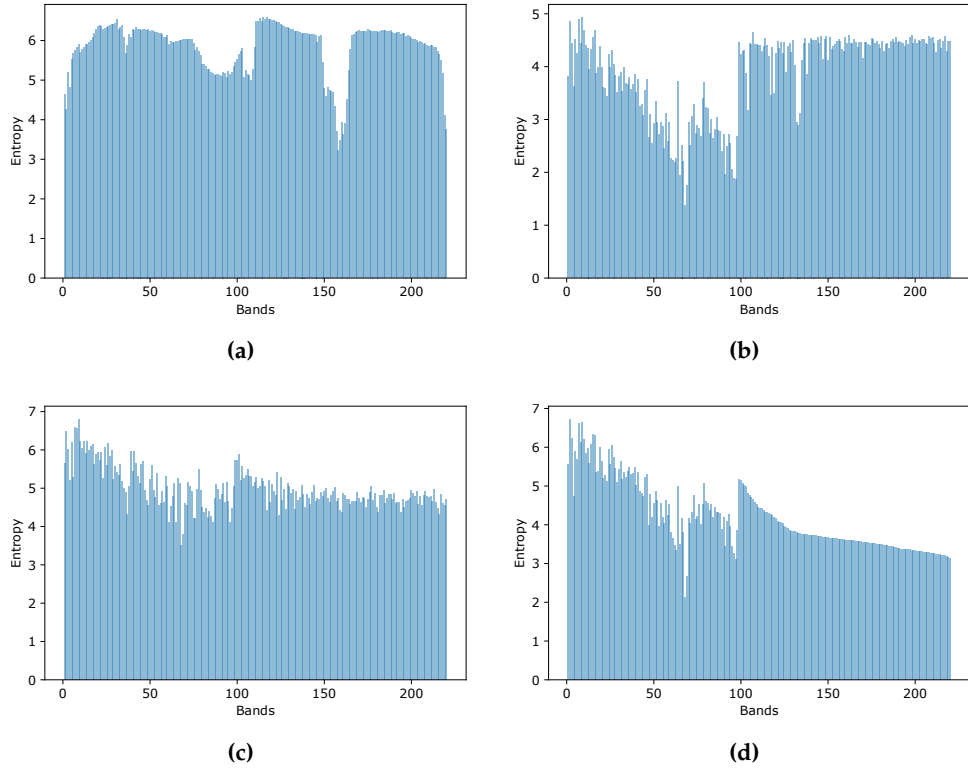


Figure 10. Indian Pines dataset entropy, with nbins = 43, computed using the Freedman–Diaconis’ criterium. a) Original dataset, b) TKD, c) NTD, d) INTD.

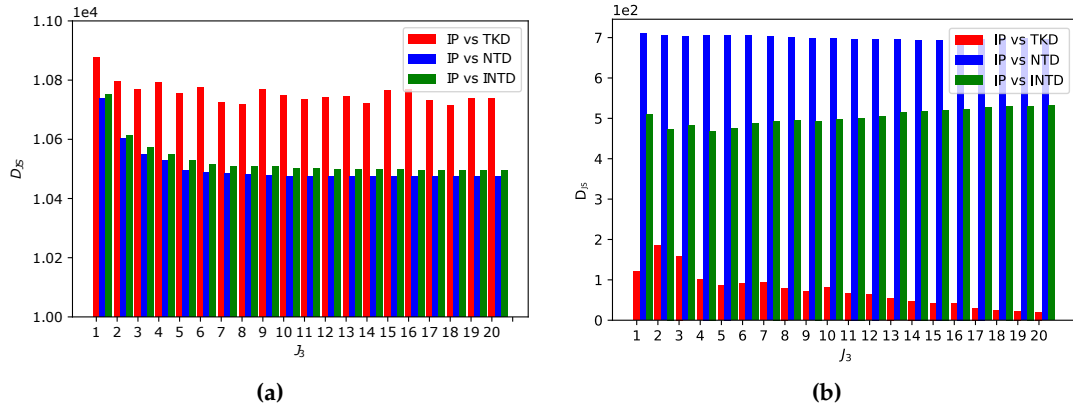


Figure 11. JSD between a) TKD / NTD / INTD core tensor vs input dataset, and b) TKD / NTD / INTD reconstruction vs input dataset.

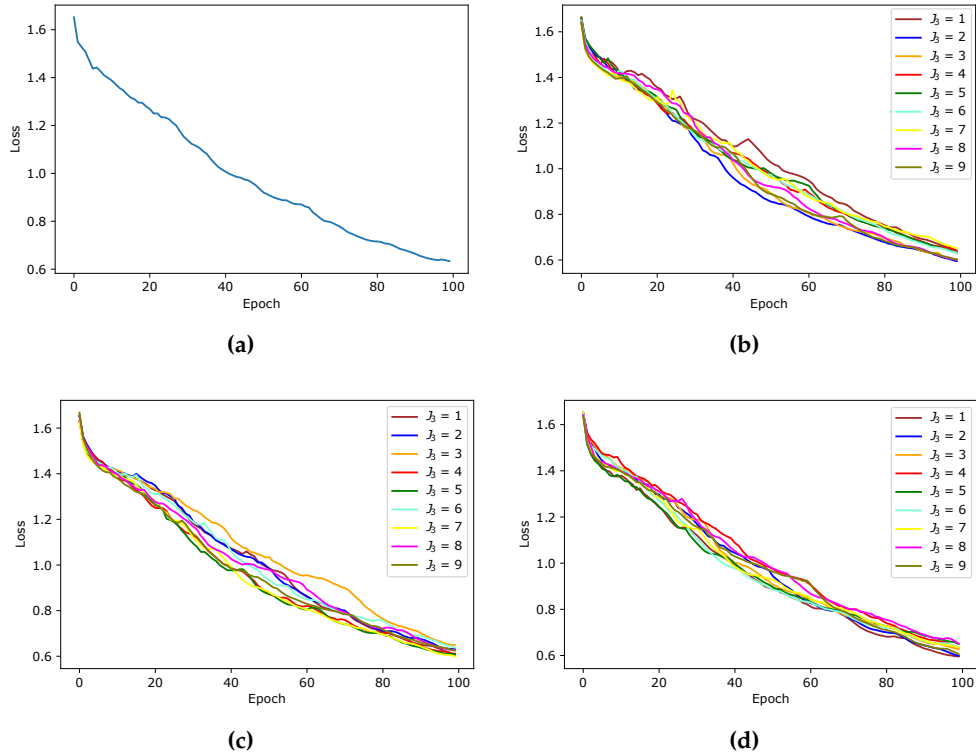
Table 7. Experimental results of the three decomposition models. Comparison of reconstruction error, execution time² and Frobenius norm of each core tensor band¹.

J_3	TKD			NTD			INTD		
	ζ	Time (s)	$\ \mathcal{G}\ _F$	ζ	Time (s)	$\ \mathcal{G}\ _F$	ζ	Time (s)	$\ \mathcal{G}\ _F$
1	398.95	9.05	1	398.95	10.02	1	398.95	9.52	1
2	169.84	17.86	0.1454	329.92	19.85	0.1461	409.54	18.32	0.1825
3	106.04	32.54	0.0531	310.10	35.79	0.0584	413.29	32.96	0.0875
4	73.80	59.10	0.0304	309.29	60.02	0.0251	414.55	60.05	0.0496
5	55.63	105.72	0.0194	304.49	105.83	0.0211	414.92	106.82	0.0395
6	36.85	174.19	0.0166	303.61	180.04	0.0145	415.82	179.89	0.0210
7	26.65	282.97	0.0101	293.72	283.35	0.0175	416.41	284.12	0.0187
8	12.24	351.40	0.0094	280.50	350.02	0.0187	414.42	350.74	0.0109
9	9.06	415.69	0.0049	280.38	408.21	0.0047	414.35	407.05	0.0097

¹ Relative to the first tensor band norm.² For the original Sentinel-2 dataset: Time = 397.21 s.

As shown by the results obtained by measuring the divergence between the original data set and its reconstruction for each decomposition model, non-negative decompositions produce a reconstruction error greater than the TKD, see Table 7. Furthermore, the fall of the error is very slow as the 3-rank of the decomposition increases.

Figure 12 shows the behavior of the loss function for the original Sentinel-2 as input dataset, as well as for the core tensor of each decomposition. Figure 12d highlights the robustness of the classifier to integer non-negative input data. This can be inferred based on the standard deviation of the INTD loss functions.

**Figure 12.** CNN Loss function with a) Sentinel-2 original dataset, b) TKD, c) NTD, and d) INTD.

Actual		Predicted					
		Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	41059	524	1475	341	2743	0.8898	
Shadow	14	2711	0	45	29	0.9685	
Cloud	1935	383	13043	439	346	0.8078	
Water	7	21	0	7213	0	0.9961	
Vegetation	12815	970	873	332	76522	0.8361	
Recall	0.7354	0.5881	0.8474	0.8617	0.9608	0.8578	

(a)

Actual		Predicted					
		Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	40602	1213	564	119	3644	0.8799	
Shadow	13	2718	2	10	56	0.9710	
Cloud	1037	454	14269	41	345	0.8837	
Water	3	82	0	7149	7	0.9872	
Vegetation	5778	1056	739	299	83640	0.9139	
Recall	0.8559	0.4921	0.09162	0.9384	0.9537	0.9056	

(b)

Actual		Predicted					
		Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	41875	266	1880	179	1942	0.9075	
Shadow	43	2694	12	21	29	0.9624	
Cloud	1616	85	13414	694	337	0.8307	
Water	6	26	4	7204	1	0.9948	
Vegetation	11488	637	949	524	77914	0.8514	
Recall	0.7609	0.7265	0.8250	0.8355	0.9712	0.8734	

(c)

Actual		Predicted					
		Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	41767	695	918	190	2572	0.9051	
Shadow	101	2605	4	8	81	0.9306	
Cloud	1426	209	14122	21	368	0.8746	
Water	1	26	2	7212	0	0.9959	
Vegetation	6441	526	2561	240	81744	0.8932	
Recall	0.8397	0.6414	0.8020	0.9401	0.9643	0.8999	

(d)

Table 8. Confusion matrices of the CNN classifier with different input data, a) Sentinel-2 original dataset, b) TKD, c) NTD and d) IANTD core tensors.

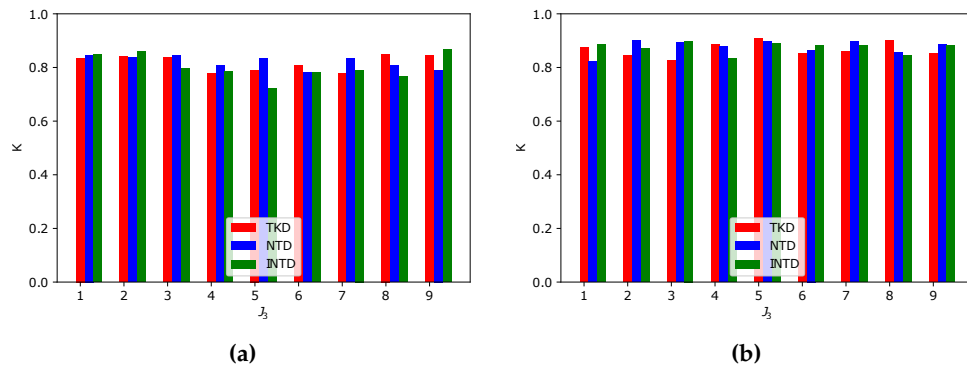


Figure 13. Kappa coefficient for TKD, NTD and INTD, a) Sentinel-2 dataset and b) Indian Pines

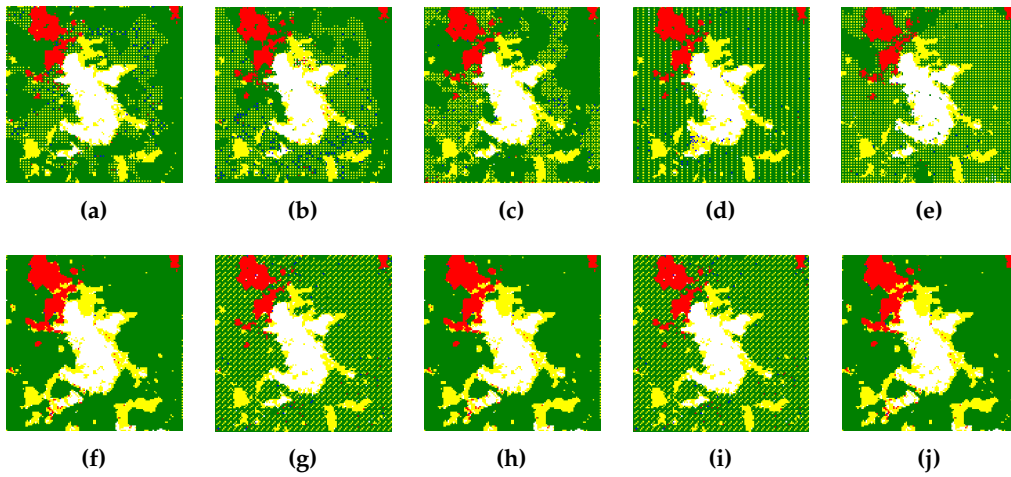


Figure 14. Qualitative results. Visualization of the predicted matrix of a testing scene with abundant vegetation and clouds, and presence of shadows and soil. Prediction after 100 epochs in the CNN used for this work a) with the original dataset without data compression, b) with TKD compressing to $J_3 = 5$, c) with NTD and no compression, $J_3 = 9$, d) with NTD to $J_3 = 5$, e) with NTD and no compression, $J_3 = 9$, f) with INTD compressing to $J_3 = 5$ and g) with INTD and no compression, $J_3 = 9$.

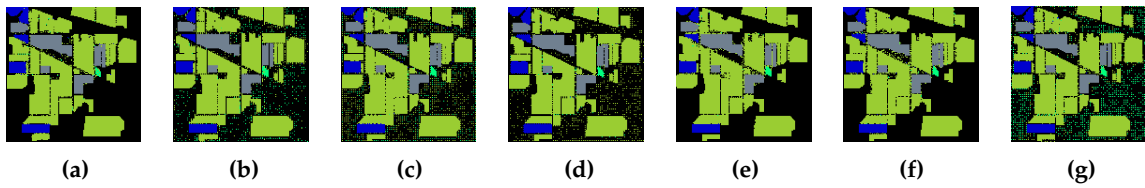


Figure 15. Qualitative results. Visualization of the predicted matrix of the Indian Pines dataset. Prediction after 100 epochs in the CNN used for this work a) with the original dataset without data compression, b) with TKD compressing to $J_3 = 5$, c) with NTD and no compression, $J_3 = 9$, d) with NTD to $J_3 = 5$, e) with NTD and no compression, $J_3 = 9$, f) with INTD compressing to $J_3 = 5$ and g) with INTD and no compression, $J_3 = 9$.

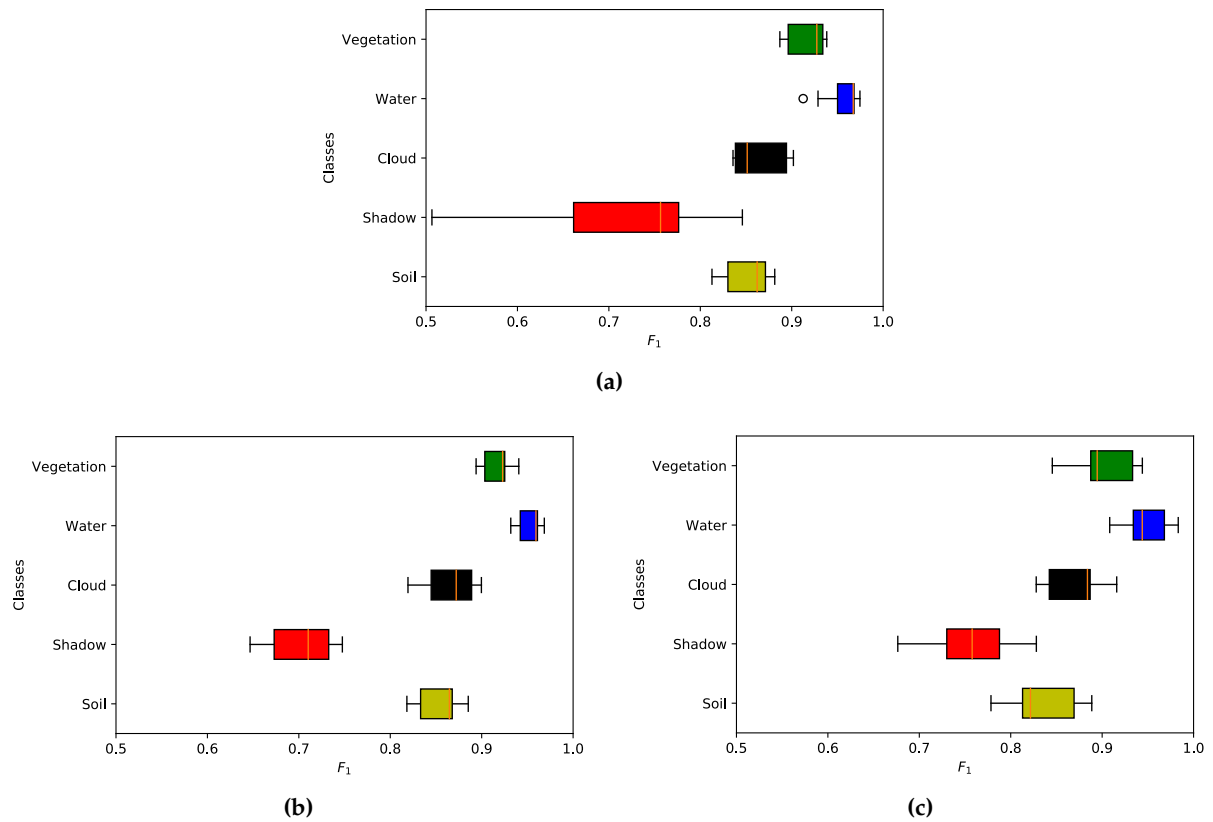


Figure 16. F1 score for Sentinel-2 dataset with a) TKD, b) NTD, and c) INTD.

Table 9. Quantitative results¹ for the Sentinel-2 test dataset running in a NVIDIA GeForce GTX 1050 Ti GPU, Intel core i7 processor, 8 Gb RAM, SSD 128 Gb, and HDD 1 Tb. Decomposition reconstruction error, average processing time per scenario, PA and Kappa's coefficient results for $J_3 = 1, \dots, 9$.

J_3	TKD			NTD			INTD		
	PA	κ	MCC	PA	κ	MCC	PA	κ	MCC
1	0.9069	0.8466	0.8481	0.9076	0.8493	0.8511	0.9002	0.8355	0.8365
2	0.9005	0.8364	0.8376	0.9155	0.8610	0.8626	0.9043	0.8424	0.8439
3	0.9056	0.8446	0.8454	0.8734	0.7956	0.8010	0.8999	0.8366	0.8384
4	0.8822	0.8081	0.8113	0.8675	0.7863	0.7926	0.8615	0.7776	0.7844
5	0.8968	0.8326	0.8346	0.8228	0.7222	0.7409	0.8703	0.7905	0.7956
6	0.8635	0.7805	0.7857	0.8637	0.7814	0.7875	0.8821	0.8084	0.8122
7	0.8973	0.8332	0.8357	0.8699	0.7905	0.7964	0.8635	0.7782	0.7820
8	0.8795	0.8061	0.8122	0.8544	0.7647	0.7709	0.9088	0.8503	0.8512
9	0.8696	0.7908	0.7983	0.9197	0.8675	0.8680	0.9057	0.8450	0.8461

¹ For the original MSI: PA = 0.8578, κ = 0.7709 and MCC = 0.7768.

Table 10. Quantitative results¹ for the Indian Pines dataset running in a NVIDIA GeForce GTX 1050 Ti GPU! (GPU!), Intel core i7 processor, 8 Gb RAM, SSD 128 Gb, and HDD 1 Tb. Decomposition reconstruction error, average processing time per scenario, PA and Kappa's coefficient results for $J_3 = 1, \dots, 10$.

J_3	TKD				NTD				INTD			
	ζ	Time (s)	PA (%)	κ	ζ	Time (s)	PA (%)	κ	ζ	Time (s)	PA (%)	κ
1	375.365	15.83	82.93	0.8207	375.36	16.21	86.41	0.8252	2965.49	15.72	86.75	0.7799
2	140.6	32.83	92.51	0.9020	67.53	31.55	92.87	0.8844	2965.49	39.63	91.82	0.8972
3	116.63	56.56	88.03	0.8946	343.33	57.32	92.31	0.9074	2957.91	62.31	93.25	0.8427
4	105.57	92.13	91.76	0.8766	343.43	97.10	90.83	0.8534	2951.48	98.94	88.39	0.8855
5	98.85	156.21	88.53	0.8981	343.33	151.23	92.75	0.8597	2938.82	164.32	89.91	0.8478
6	92.52	298.80	83.99	0.8653	339.64	301.09	89.90	0.8990	2929.61	313.21	92.36	0.7913
7	87.41	520.13	89.21	0.8973	337.89	515.63	92.74	0.8523	2895.02	535.08	88.94	0.8540
8	79.53	715.69	88.33	0.8561	335.90	704.21	89.86	0.8599	2876.32	732.12	92.15	0.8469
9	76.15	881.21	89.97	0.8853	335.91	876.36	92.03	0.8891	2866.45	901.35	92.01	0.8603
10	72.67	934.78	88.61	0.8871	335.74	910.84	92.93	0.8864	2854.12	978.54	91.95	0.8462

¹ For the original Indian Pines dataset: Time = 878.09 s, PA = 91..22%, κ = 0.9040.

7. Discussion and Comparison

In this work, the hyperspectral input dataset is decomposed by a Tucker-based decomposition model to transform them from the spectral bands domain (wavelength) to a new tensor bands domain. The decompositions are restricted to preserve the spatial domain and to compress the spectral domain. Figure 8 it can be seen how the endmembers of the materials of interest behave in a way that, from a salient band point of view, the first new tensor bands are able to provide enough information to a CNN to differentiate diverse materials. On the other hand, From the information theory point of view, the entropy computed for each original and core tensors band reinforce this assertion (See Figure 10).

Unlike previous works, the introduction of information metrics in this work aids to trade off the empirical setting of the multirank TKD parameters. Although the process is still semi-empirical, it is based on metrics that quantify the amount of information and the divergence from the original data. It is worth noting that, in this work, the compression is developed only in the spectral domain, but the basis of the proposal can also be applicable for other kinds of decomposition.

Qualitative results (Figures 14–15) and quantitative results in Figure ?? present the performance evaluation of the CNN, based on PA, comparing the three models based on TKD. Comparing with results shown in previous works [?], [?], [?], the proposed INTD overcomes unsupervised classification algorithms, as well as decomposition without non-negativity and integer restrictions. While it is true that the PA metric is not the best for an unbalanced dataset, it is a good starting point for a general comparison. Nevertheless, the kappa coefficients results, shown in Figure 13, show greater stability in classification for the TKD, but as the value of J_3 increases, the NTD and INTD improve their performance, while the TKD fall. this can be attributed to the phenomenon of overfitting.

Tables 9 and 10 allow us to make a fair comparison among the Tucker-based decompositions. First of all, as expected, the TKD reconstruction error decrease faster than the approximation with non-negativity and integer constraints. On the other hand, the analysis of PA and the Kappa's coefficient, in combination with the entropy, give a measure linked to the diminution of execution time in favor of the NTD and the proposed INTD approximation.

8. Conclusions

In this work, we had the purpose of improving the features of a multi- or hyperspectral image, while reducing dimensionality and, in turn, the computational complexity of a classification CNN. From the results presented above and the analysis of each metric, it has been shown that the constraints imposed to a decomposition model, as the NTD and INTD, produce an improvement in classification metrics of CNN. It is worth noting that, depending on the model of the classifier, the TD should be limited to provide characteristics that aids the classifier to improve its performance.

Results shown in Figure 11 we can conclude that the proposed integer non-negative approximation

8.1. Open issues

-
-
-

Author Contributions: Conceptualization, J.L.; formal analysis, D.T.; investigation, J.L.; methodology, J.L., D.T., and C.A.; resources, C.A.; software, J.L.; supervision, D.T. and C.A.; validation, D.T. and C.A.; writing—original draft, J.L. and D.T.

Funding: This work was supported by the National Council of Science and Technology CONACYT of Mexico under grant XXXXXXXX.

Acknowledgments:

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
CNN	Convolutional neural network
CPD	Canonical Polyadic Decomposition
DL	Deep Learning
FCN	Fully Convolutional Network
HOOI	Higher-Order Orthogonal Iteration
HOSVD	Higher-Order Singular Value Decomposition

References

- Tempfli, K.; Huurneman, G.; Bakker, W.; Janssen, L.; Feringa, W.; Gieske, A.; Grabmaier, K.; Hecker, C.; Horn, J.; Kerle, N.; et al. *Principles of Remote Sensing: An Introductory Textbook*, 4th ed.; ITC: Geneva, Switzerland, 2009.
- He, Z.; Hu, J.; Wang, Y. Low-rank tensor learning for classification of hyperspectral image with limited labeled sample. *IEEE Signal Process.* **2017**, *145*, 12–25.
- Richards, A.; Xiuping, J.J. Band selection in sentinel-2 satellite for agriculture applications. In *Remote Sensing Digital Image Analysis*, 4th ed.; Springer-Verlag: Berlin, Germany, 2006.
- Zhang, T.; Su, J.; Liu, C.; Chen, W.; Liu, H.; Liu, G. Band selection in sentinel-2 satellite for agriculture applications. In Proceedings of the 23rd International Conference on Automation & Computing, University of Huddersfield, Huddersfield, UK, 7–8 September 2017.
- Xie, Y.; Zhao, X.; Li, L.; Wang, H. Calculating NDVI for Landsat7-ETM data after atmospheric correction using 6S model: A case study in Zhangye city, China. In Proceedings of the 18th International Conference on Geoinformatics, Beijing, China, 18–20 June 2010.
- Gao, B. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sens. Environ.* **1996**, *58*, 1–6.
- Ham, J.; Chen, Y.; Crawford, M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501.
- Hearst, Marti A. Support Vector Machines. *IEEE Intell. Syst. J.* **1998**, *13*, 18–28.
- Huang, X.; Zhang, L. An SVM Ensemble Approach Combining Spectral, Structural, and Semantic Features for the Classification of High-Resolution Remotely Sensed Imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 257–272.
- Delalieux, S.; Somers, B.; Haest, B.; Spanhove, T.; Vanden Borre, J.; Mucher, S. Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers. *Remote Sens. Environ.* **2012**, *126*, 222–231.
- Kemker, R.; Salvaggio, C.; Kanan, C. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 60–77.

12. Pirotti, F.; Sunar, F.; Piragnolo, M. Benchmark of machine learning methods for classification of a sentinel-2 image. In Proceedings of the XXIII ISPRS Congress, Prague, Czech Republic, 12–19 July 2016.
13. Mateo-García, G.; Gómez-Chova, L.; Camps-Valls, G. Convolutional neural networks for multispectral image cloud masking. In Proceedings of the IGARSS, Fort Worth, TX, USA, 23–28 July 2017.
14. Guo, X.; Huang, X.; Zhang, L.; Zhang, L.; Plaza, A.; Benediktsson, J. A. Support Tensor Machines for Classification of Hyperspectral Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3248–3264.
15. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H. Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163.
16. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer Verlag: New York, NY, USA, 2002.
17. Kolda, T.; Bader, B. Tensor Decompositions and Applications. *SIAM Rev.* **2009**, *51*, 455–500.
18. Lopez, J.; Santos, S.; Torres, D.; Atzberger, C. Convolutional Neural Networks for Semantic Segmentation of Multispectral Remote Sensing Images. In Proceedings of the LATINCOM, Guadalajara, Mexico, 14–16 November 2018.
19. European Space Agency. Available online: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2> (accessed on 15 July 2019).
20. Kemker, R.; Kanan, C. Deep Neural Networks for Semantic Segmentation of Multispectral Remote Sensing Imagery. *arXiv* **2017**, arXiv:abs/1703.06452.
21. Hamida, A.; Benoît, A.; Lambert, P.; Klein, L.; Amar, C.; Audebert, N.; Lefèvre, S. Deep learning for semantic segmentation of remote sensing images with rich spectral content. In Proceedings of the IGARSS, Fort Worth, TX, USA, 23–28 July 2017.
22. Wang, Q.; Lin, J.; Yuan, Y. Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1279–1289.
23. Li, S.; Qiu, J.; Yang, X.; Liu, H.; Wan, D.; Zhu, Y. A novel approach to hyperspectral band selection based on spectral shape similarity analysis and fast branch and bound search. *Eng. Appl. Artif. Intell.* **2014**, *27*, 241–250.
24. Zhang, L.; Zhang, L.; Tao, D.; Huang, X.; Du, B. Compression of hyperspectral remote sensing images by tensor approach. *Neurocomputing* **2015**, *147*, 358–363.
25. Astrid, M.; Lee, Seung-Ik. CP-decomposition with Tensor Power Method for Convolutional Neural Networks compression. In Proceedings of the BigComp, Jeju, Korea, 13–16 February 2017.
26. Chien, J.; Bao, Y. Tensor-factorized neural networks. *IEEE Trans. Neural Networks Learn. Syst.* **2018**, *29*, 1998–2011.
27. An, J.; Lei, J.; Song, Y.; Zhang, X.; Guo, J. Tensor Based Multiscale Low Rank Decomposition for Hyperspectral Images Dimensionality Reductio. *Remote Sens.* **2019**, *11*, 1485.
28. Li, J.; Liu, Z. Multispectral Transforms Using Convolution Neural Networks for Remote Sensing Multispectral Image Compression. *Remote Sens.* **2019**, *11*, 759.
29. An, J.; Song, Y.; Guo, Y.; Ma, X.; Zhang, X. Tensor Discriminant Analysis via Compact Feature Representation for Hyperspectral Images Dimensionality Reduction. *Remote Sens.* **2019**, *11*, 1822.
30. Absil, P.-A.; Mahony, R.; Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*, 1st ed.; Princeton University Press: Princeton, NJ, USA, 2007.
31. De Lathauwer, L.; De Moor, B.; Vandewalle, J. On the best rank-1 and rank-(R_1, R_2, \dots, R_N) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1324–1342.
32. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press, 2016.
33. Sheehan, B. N.; Saad, Y. Higher Order Orthogonal Iteration of Tensors (HOOI) and its Relation to PCA and GLRAM. In Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, MN, USA, 26–28 April 2007.
34. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
35. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A Multilinear Singular Value Decomposition. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1253–1278.
36. Rodes, I.; Inglada, J.; Hagolle, O.; Dejoux, J.; Dedieu, G. Sampling strategies for unsupervised classification of multitemporal high resolution optical images over very large areas. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012.

Sample Availability: Samples of the compounds are available from the authors.

455 © 2020 by the authors. Submitted to *Remote Sens.* for possible open access publication
456 under the terms and conditions of the Creative Commons Attribution (CC BY) license
457 (<http://creativecommons.org/licenses/by/4.0/>).