MDPI

Article

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 Ingenering Department, Miguel Hidalgo 102, Comunidad de Valencia 38253 Juventino Rosas, Guanajuato, Mexico; jyanezptc@upjr.edu.mx
* Correspondence: josue.lopez@cinvestav.mx

Version November 27, 2020 submitted to Remote Sens.

Abstract: Tensor-based algorithms for data compression have evolved in recent years according to the needs of several research areas. Tucker Decomposition (TKD) is one of the most popular factorization methods based on tensor algebra, but it is clear that it is not the only algorithm which can produce a factorization for a given input data set. Besides, this decomposition does not have singular solutions, i.e., it converges to local minima. Hence, depending on the input data, tensor-based decompositions can achieve better solution to a specific input. The phenomenology of Remote Sensing (RS) Hyperspectral Images (HSI) belongs to the set of natural numbers, i.e., the set of positive integers. Then, a non-negative tensor factorization suggest a more suitable decomposition for positive data by nature. The main purpose of this work is to prove the benefits in processing time, as well as in accuracy, of using a well-posed factorization algorithm. Specifically, this paper performs a quantitative analysis of tensor-based factorization algorithms applied to semantic segmentation of HSI using Deep Convolutional Networks (DCN).

Keywords: deep convolutional networks; hyperspectral imagery; tensor decomposition

1. Introduction

Reducing the dimensionality of input data for machine learning algorithms has been one of the most active research areas in recent years []. The insertion of tensor-based algorithms for these types of tasks inspired a change in several areas, such as image processing. [].

Most of the researchs in the image processing area require data acquired by multiple sensors. Even in the simplest case, color images, data acquired by three sensors that perceive reflectance of an object are processed. Each sensor receives reflectance in different wavelength ranges, and by merging that data, a color image is produced. Thus, the images can be represented in a form of three-dimensional arrays or third-order tensors. Two dimensions represent the spatial properties and the third dimension denotes the depth, i.e., the spectral bands. Medical analysis [], minorology [], agriculture, [], radar images [] and, above all, remote sensing multi- and hyper-spectral images [] can be represented, by nature, as third-order tensors.

Spectral images make certain image processing tasks much easier []. Recently, the use of this type of data has grown exponentially in various areas such as agriculture [], medical analysis [], biomedical [], natural disaster prediction [], security affairs [], among others. The ability to obtain

information about a target not only by its reflectance in the spatial domain, but also by response at different wavelengths, has driven a growth in accuracy and precision in tasks such as classification and segmentation [].

Few years ago, several unsupervised classification and segmentation algorithms [] were developed, taking advantage of the properties that spectral data produce. Subsequently, with the introduction of supervised machine learning algorithms such as Support Vector Machines (SVM) [], k-Nearest Neighbors (k-NN) [] and Artificial Neural Networks (ANN) [], it was found that, under certain conditions, there is a direct relationship between the number of bands used and the performance of these algorithms []. However, with the aim of improving results, ANN models evolved into Deep Learning Neural Networks (DLNN) []. This caused the computational complexity to rise considerably and spectral image processing was not easily achievable. The foregoing requires having robust computer equipment to achieve competitive results in time.

Several works opted for matrix factorization algorithm to reduce the high-dimensionality of spectral images []. More recently, with the development of tensor factorization algorithms [], it has been found that some algorithms based on tensor algebra produce advantages over those based on matrices []. Nevertheless, the data produced by both of them are hard to understand for supervised classification algorithms that need spatial relation between pixels to produce a wise prediction [].

In this work, the proposal is to find an alternative solution to the problem described previously. We propose a model that reduces the computational load of hyperspectral imagery classification supervised algorithms through tensor decomposition models. This produces a lower dimensional tensor while preserving the structural and numerical nature of the original data.

1.1. Previous work

There are several works focused on the development of frameworks that reduce computational complexity of machine learning algorithms for semantic segmentation of hyperspectral datasets []. The crucial factor, which is addressed in this work, is to achieve compression of the input data to reduce the high number of computatios, but without sacrificing pixel accuracy, overall accuracy, precision and recall in the classification task.

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 algorithms 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, there was also the possibility of using multispectral images due to the small number of spectral bands, which still made efficient results in classification without dimensionality reduction achievable, as done in [11], [18], [21] and [**?**]. However, the need to increase classification performance forces researchers to use data with more features that favor and aid the classification of various classes, which are difficult to differentiate with little spectral data. Thus, more recent researchs have decided to use hyperspectral images with tensor decompositions, which has increased the results in classification accuracy [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 this work pursuits to preserve the nature of the image just compressing the main information in the positive 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. [] (2019)	MSI	TKD	Spectral	FCN
Sayeh, M. [] (2019)	HSI	NTD	Spatial-Spectral	3D-CNN
Our framework	**MSI/HSI**	**iNTD/NTD**	**Spectral**	**CNN**

1.2. Motivation

Nowadays, RS image processing is applied in several areas related to caring of the planet. Nevertheless, task such as classification becomes more complex due to low spatial resolutions, this is offset by the use of devices with other features such as spectral sensors.

On the other hand, CNNs have been widely used in recent years in the area of RGB image semantic classification and segmentation. Its performance is highly competitive and the development of various improvement strategies have considerably reduced its computational cost []. However, the computational complexity of the algorithm means that the increase in the dimensionality of the input data produces a significant increase in the computational load []. This is why the processing of high-dimensional images such as multi and hyperspectral images becomes much more computationally expensive.

Some data compression strategies have favorably reduced the dimensionality of the data. Decomposition methods based on the matrix and tensor approaches have been applied as pre-processing of input data of neural networks. In tensor decompositions, the processing of the data in its natural format, i.e., as N-th order tensors, improves the decomposition process because it considers the dependence of the data in its different modes. Although a decomposition can compress the data, it is also important to note that the inappropriate selection of some decomposition parameters could lead to information losses, which would penalize the performance of a CNN.

Under these considerations, this work is motivated to develop a low computational complexity and competitive in performance framework that helps various fields of application of remote sensing image processing to solve classification tasks.

1.3. Contribution

Unlike previous works, this work seeks to adapt the data in a more efficient way to the input of deep convolutional networks. Convolutional Neural Network 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, the proposed framework maintains the integer and non-negativity tensor nature of the spectral images and the spatial dimensionality to preserve spatial-spectral correlation of the data while reducing the spectral dimensionality, in order to decrease computational load in the pixel-wise classification process. In addition, we address the problem of setting the 3-rank selection hyperparameters by measuring the divergences, which helps to estimate a lower multi-rank approximation.

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

1. The framework INTD1-CNN proposed in this work, develops an approximation strategy to improve performance of semantic segmentation convolutional neural networks by finding suitable tensor data, preserving spatial correlation and values in the set of the natural numbers while compressing the spectral domain and in turn decreasing computational load.
2. Furthermore, this work proposes an empirical strategy, based on information theory, for defining the n-rank in the spectral domain (mode 3) of the compression models based on the TKD.
3. This work also presents an exhaustive performance analysis measuring and comparing its efficiency with metrics as pixel accuracy (PA) in function of the number of new tensor bands, F1, MCC, Kappa coefficient, orthogonality degree of the factor matrices and 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 decribed the framework proposed for compression and pixel-wise clasification 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-dimesional array, which elements $x_{i_1,i_2,\ldots,i_n}$ are indexed by $i_n \in 1, 2, \ldots, I_n$ for $1 \leq n \leq N$.

Throughtout 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 $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is the sum of the products of their entries, i.e., $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} a_{i_1 \ldots i_N} b_{i_1 \ldots i_N}$.

2.3. N-Mode Product

It means the multiplication of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 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 $\mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$ [17].

2.4. Rank-One Tensor

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

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)} \tag{1}$$

where $\circ$ denotes the outer product and $\mathbf{a}^{(\mathbf{n})}$ denotes a vector in a sequence of N vectors. Each element of the tensor is the product of the corresponding vector elements, i.e., $x_{i_1 i_2 \ldots i_N} = a^{(1)}_{i_1} \ldots a^{(N)}_{i_N}$.

2.4.1. N-Rank

The n-rank of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 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 $n = 1, \ldots, N$, then $\mathcal{X}$ has a rank $-(R_1, \ldots, R_N)$ tensor [17].

Table 2. Tensor algebra notation summary

$\mathcal{A}$, $\mathbf{A}$, $\mathbf{a}$, a	Tensor, matrix, vector and scalar respectively
$\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$	N-order tensor of size $I_1 \times \cdots \times I_N$.
$a_{i_1 \ldots i_N}$	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. $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$
$\mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)}$	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 $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\mathcal{W}^{(-n)} = \mathcal{X} \times_1 \mathbf{U}^{(1)\mathrm{T}} \cdots \times_{n-1} \mathbf{U}^{(n-1)\mathrm{T}} \times_{n+1} \mathbf{U}^{(n+1)\mathrm{T}} \cdots \times_N \mathbf{U}^{(N)\mathrm{T}} \tag{8}$$

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

3.1.1. Non-negative Tucker Decomposition (NTD)

The NTD is a decomposition based on the Tucker model. It is a new tensor factorization method with nonegativity constraints []. For the third-order case, the NTD, as defined by Cichocky [15], can be formulated as follows. Given a third-order tensor $\mathcal{X} \in \mathbb{R}_+^{I_1 \times I_2 \times I_3}$ find a core tensor $\mathcal{G} \in \mathbb{R}_+^{I_1 \times I_2 \times I_3}$ 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 given in Eq. (2). As well as for the TKD model, the best 3-rank approximation of a nonnegative tensor can be computed by an iterative algotihm as HOOI, maximizing the cost function given in equation 7. Algorithm 1 shows the HOOI algorithm for a NTD.

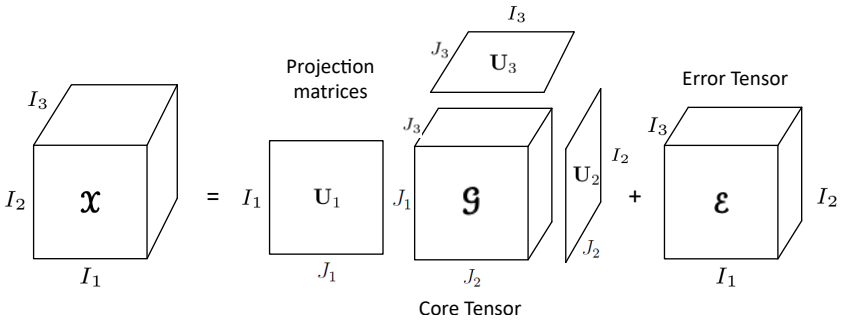


Figure 1. Tucker decomposition for a third-order tensor.

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

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

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, $I_!$, I_2 and I_3 represent the height, widht 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 an 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 best rank-(R_1, R_2, R_n) approximation and its core tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times J_3}$, through Non-negative Tensor Decompositions. The n-rank of the decomposition $\text{rank}_n(\mathcal{X})$ is set as rank-(I_1, I_2, J_3), where $J_3 < I_3$. This built the input space of a pixel-wise classification using CNNs and produce an output matrix $\hat{\mathbf{Y}}$ of predicted classes, achieving competitive performance metrics for pixel-wise classification while decreasing computational load in the classification process.

4.3. Mathematical Definition

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

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

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, the classifier and the decision making.

5.1. Tensor modeling

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

$$
\mathbf{x}_{i_1 i_2} = \sum_{c=1}^{C} (\alpha_{i_1 i_2 c} \mathbf{m}_c + \boldsymbol{\eta}) \tag{10}
$$

where α_c is the contribution of material c at pixel $i_1 i_2$, $\mathbf{m}_c$ denotes the endmember of a specific material c, and $\boldsymbol{\eta}$ represents an additive noise vector. The abundance vectors $\boldsymbol{\alpha}_{i_1 i_2}$ must always satisfy two constraints, i) the non-negativity, $\alpha_{i_1 i_2 c} \geq 0$ for all $c = 1, \ldots, C$, and ii) the sum-to-one restriction, $\sum_{c=1}^{C} \alpha_{i_1 i_2 c} = 1$. Figure 2a shows the spectral signatures for the Indian Pines dataset.

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 $\mathcal{X}$, 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 $\mathcal{X}$ while keeping classifier performance, we propose to use a restricted NTD denoted as $\mathcal{T}$, producing a core tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times J_3}$ and n factors matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, expressed as

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

where the decomposition is restricted to preserve the spatial domain and to be only in the 3rd-mode by the Tucker-1 model

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

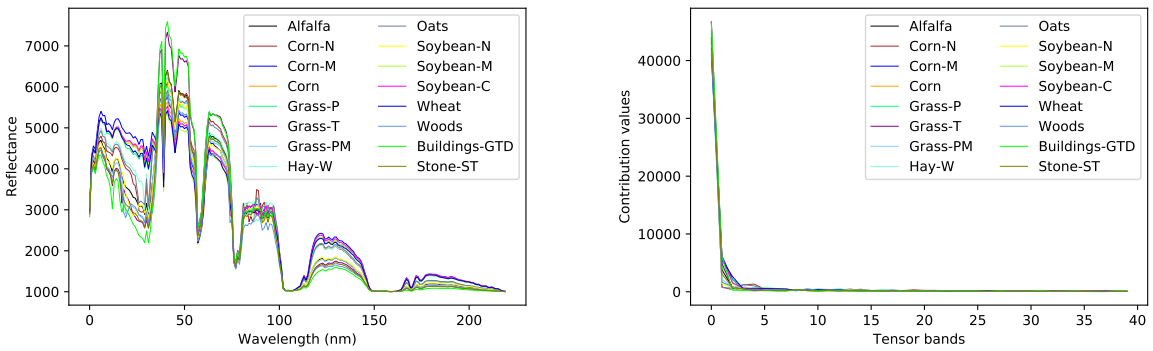


Figure 2. Behavior of the 16 classes of the Indian Pines dataset, a) in the spectral domain (spectral signatures) and b) in the the tensor bands domain after NTD.

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

$$\mathbf{g}_{i_1 i_2} = \sum_{c=1}^{C} (\beta_{i_1 i_2 c} \mathbf{s}_c + \boldsymbol{\eta}) \tag{13}$$

where β_c is the contribution of material c at pixel $i_1 i_2$ and $\mathbf{s}_c$ denotes the endmember of a specific material c. Figure 2b shows the new tensor bands values for each class after NTD.

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 decompostions 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.,

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

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

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

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

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

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

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

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).

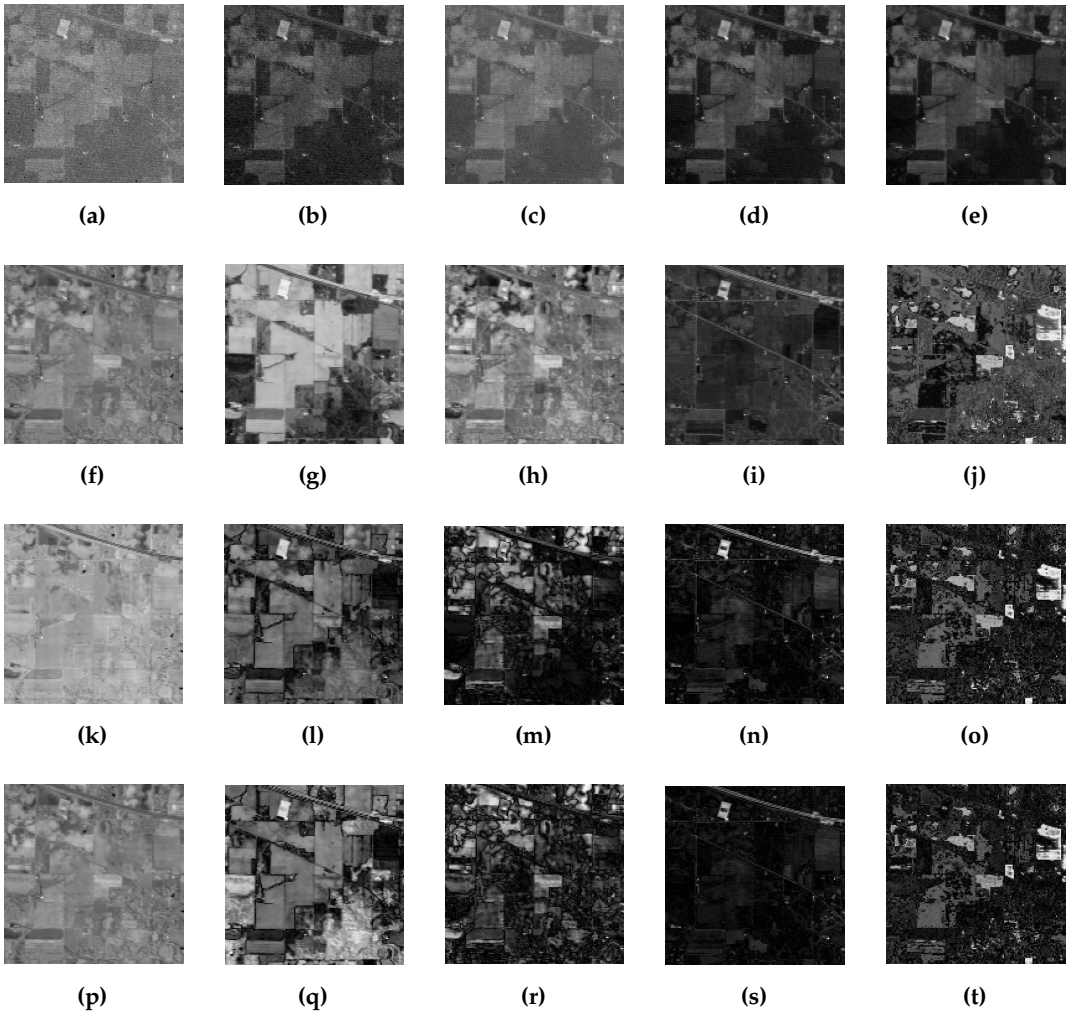


Figure 3. Visualisation of Indian Pines 1st to 5th 3a to 3e original spectral bands, 3f to 3j tensor bands with TKD, 3k to 3o tensor bands with NTD, and 3p to 3t tensor bands with INTD.

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 4 shows the big picture of the framework proposed.

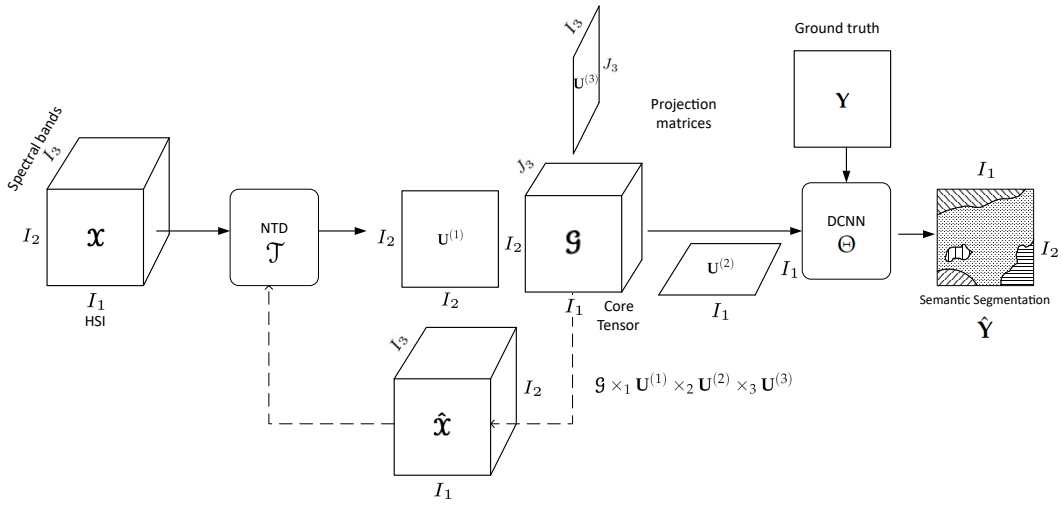


Figure 4. Big picture of the framework proposed.

5.4. Divergence analysis

The framework proposed in this work faces two main challenges. The first is the selection of the decomposition method. We are looking for a better representation of the input dataset for a 3D-CNN, so, we limit the set of methods to those that produce a decomposition tensor with the same structure as the input tensor. TKD, NTD and the INTD proposed generate a core tensor with the desired properties.

The second challenge is the search for the rank $-(J_1, J_2, J_3)$. As we want to preserve the spatial domain $J_1 = I_1$ and $J_2 = I_2$, but J_3 has to be selected so that the performance of the classifier does not decrease considerably. Both decisions are made under a criterion under the probabilistic point of view.

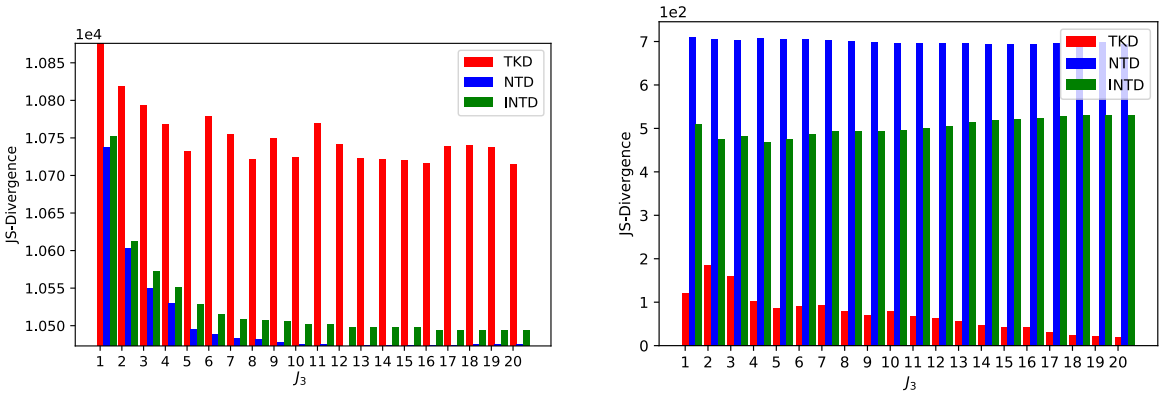
For this work the Jensen-Shanon divergence (JSD) is used as a metric to quantify how different the core tensor of a decomposition is from the input dataset from an information theory point of view. This method measures the difference between two probability distributions. The JSD can be computed as

$$D_{JS}(X\|G) = \frac{1}{2}D_{KL}(X\|M) + \frac{1}{2}D_{KL}(G\|M) \tag{18}$$

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

$$D_{KL}(X\|G) = \sum_{i=1}^{I} X_i(x) \log \frac{X(x_i)}{G(x_i)} \tag{19}$$

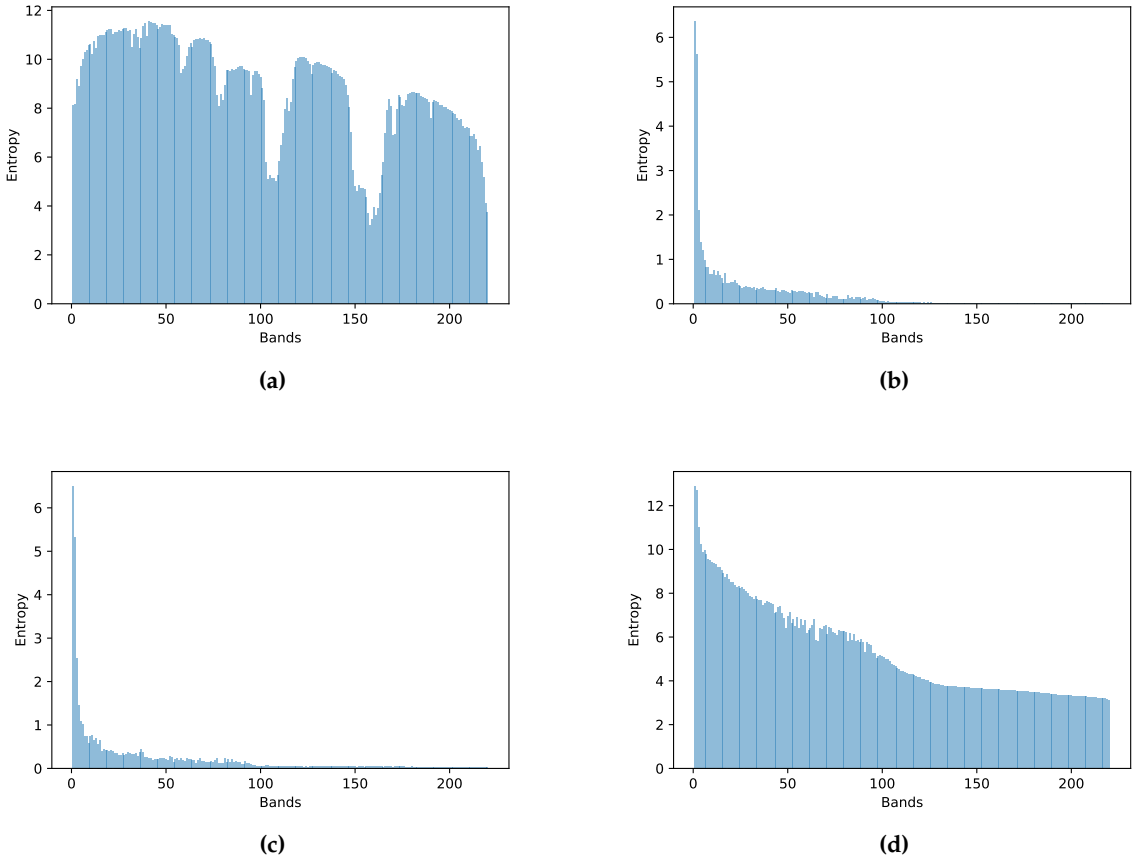
where $X(x_i)$ and $G(x_i)$ represent the probability of the i-th element at distributions X and G respectively. Figure 5 shows the JS divergence between the core tensor of the TKD, NTD and INTD versus the original MSI (Figure 5a), as well as the divergence between the MSI and the recontruction of each decomposition (Figure 5b). In Figure 5a it can be seen that, as expected, decompositions with non-negativity constraints generate core tensors with lower divergence with respect to the input dataset. In turn, this figure shows how the INTD stabilizes its divergence in a lower value of J_3 in comparisson with TKD and NTD. On the other hand, Figure 5b shows that, in reconstruction, the TKD is more precise than the others, which can be attributed to the freedom in the decomposition.



(a) **(b)**

Figure 5. JSD between a) TKD / NTD / INTD core tensor vs input dataset, and b) TKD / NTD / INTD reconstruction vs input dataset.

The divergence study is reinforced by an entropy analysis to quantify the information contribution of each band from the original spectral image and from the core tensor of each decomposition. Figure 6 presents the entropy of each band at the Indian Pines dataset.



(a) **(b)**

(c) **(d)**

Figure 6. Indian Pines entropy a) original dataset, b) TKD, c) NTD, d) INTD

6. Experimental Results

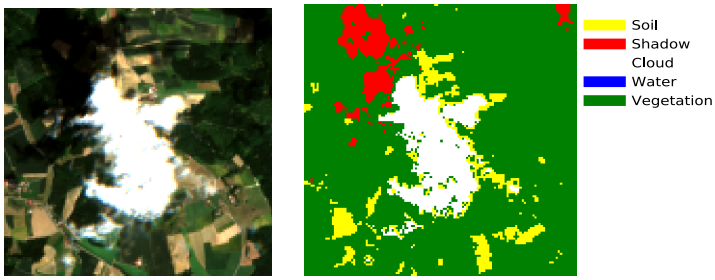
6.1. Input Data

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.

6.1.1. Sentinel-2

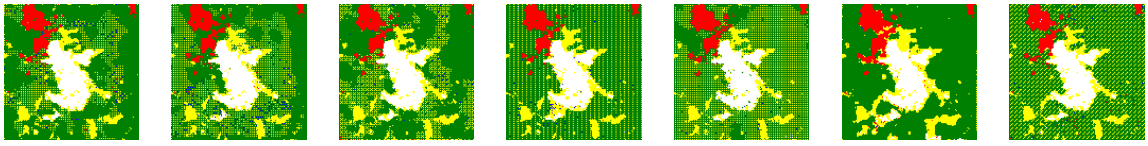
This dataset proposed by Lopez et al. [**?**] is composed of 110 RS Sentinel-2 scenarios from central Europe. It has 100 scenarios as the training space and 10 scenarios for testing, all of them with 128×128 pixels with spatial resolution of $20m^2$ and 9 spectral bands in the range $490 - 2190nm$. The 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.

Class	Dataset
Soil	425,147
Shadow	45,418
Cloud	250,020
Water	218,620
Vegetation	863,035



(a) **(b)**

Figure 8 & Table 2. Sentinel dataset, Table) Samples per class, a) True color image and b) Ground truth.



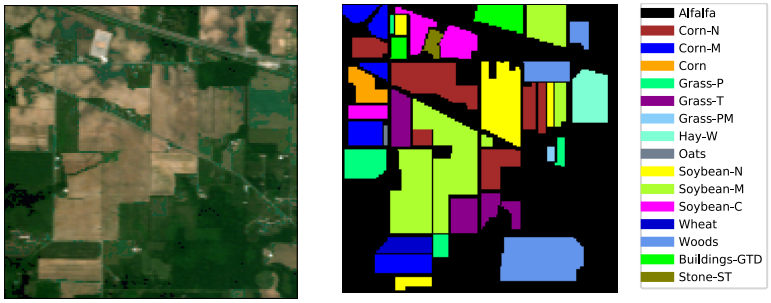
(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)**

Figure 9. 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, $J_3 = 9$.

6.1.2. Indian Pines

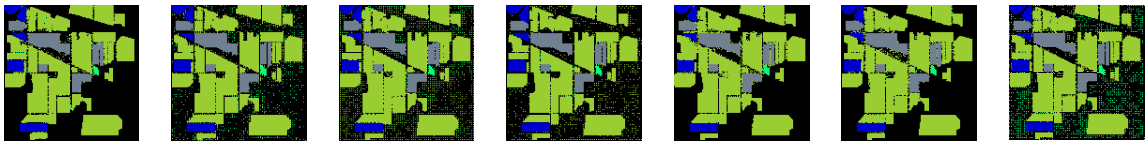
This dataset is a scene produced by AVIRIS in North-western Indiana and consists of 145×145 pixels and 224 spectral bands in the wavelength range $0.4 - 2.5\mu m$. The Indian Pines scene contains two-thirds agriculture, and one-third forest or other natural perennial vegetation. There are two major dual 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 **??** shows the true color image of Salinas dataset, as well as the ground truth with each of its corresponding classes and Table **??** the number of samples for each class.

Class	Samples
Alfalfa	46
Corn-N	1428
Corn-M	830
Corn	237
Grass-P	483
Grass-T	730
Grass-PM	28
Hay-W	478
Oats	20
Soybean-N	972
Soybean–M	2455
Soybean-C	593
Wheat	205
Woods	1265
Building-GTD	386
Stone-ST	93



(a) (b)

Figure 10 & Table 3. Indian Pines dataset, Table) Samples per class, a) True color image and b) Ground truth.



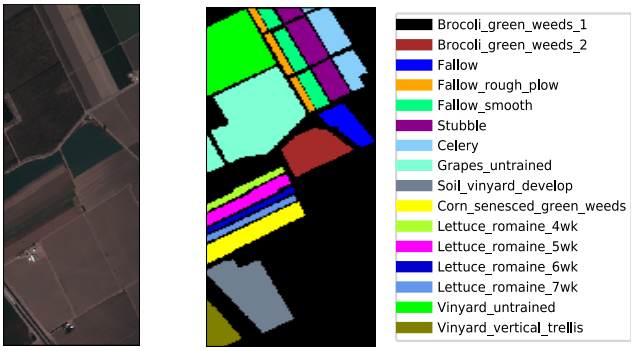
(a) (b) (c) (d) (e) (f) (g)

Figure 11. 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 compresing to $J_3 = 5$ and g) with INTD and no compression, $J_3 = 9$.

6.1.3. Salinas

This scene was collected by the AVIRIS sensor over Salinas Valley, California. It has 512×217 pixels with spatial resolution $3.7m$, and 224 spectral bands. It includes vegetables, bare soils, and vineyard fields. Salinas groundtruth contains 16 classes shows in table **??**. Salinas data are available at Salinas dataset. Figure **??** shows the true color image of the Salinas dataset, as well as the ground truth labeled with each of its corresponding classes and Table **??** the number of samples for each class.

Class	Samples
Brocoli-GW-1	2009
Brocoli-GW-2	3726
Fallow	1976
Fallow-RP	1394
Fallow-S	2678
Stubble	3959
Celery	3579
Grapes-U	11271
Soil-VD	6203
Corn-SGW	3278
Lettuce-R-4wk	1068
Lettuce-R-5wk	1927
Lettuce-R-6wk	916
Lettuce-R-7wk	1070
Vinyard-U	7268
Vinyard-VT	1807



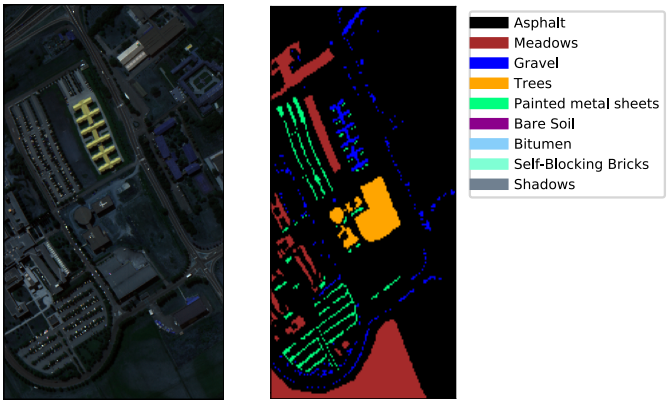
(a) (b)

Figure 12 & Table 4. Salinas dataset, Table) Samples per class, a) True color image and b) Ground truth.

6.1.4. Pavia University

This scene was collected by the ROSIS sensor over Pavia University, nothern Italy. It has 610×340 pixels with spatial resolution $1.3m$. and 103 spectral bands. Pavia groundtruth contains 9 classes, some of them described in table **??**. Pavia University data are available at Pavia University dataset. Figure **??** shows the true color image of the Pavia dataset, as well as the ground truth labeled with each of its corresponding classes.

Class	Samples
Asphalt	6631
Meadows	18649
Gravel	2099
Trees	3064
Painted metal sheets	1345
Bare Soil	5029
Bitumen	1330
Self-Blocking Bricks	3682
Shadows	947



(a) **(b)**

Figure 13 & Table 5. Pavia dataset, Table) Samples per class, a) True color image and b) Ground truth.

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 6. 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

6.2. CNN Specifications

The model used to evaluate the framework proposed in this work is Segnet []. Table 7 shows the hyperparameters of the CNN set by cross-validation and the hardware specifications.

Table 7. Experiments' software and hardware specifications.

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

J_3	TKD	NTD	INTD
1	140.267	367.535	385.986
2	116.638	347.336	382.857
3	105.574	343.436	381.385
4	98.8506	343.336	381.627
5	92.5201	339.647	382.37
6	87.4121	337.896	382.528
7	83.4363	335.908	382.578
8	79.5316	335.914	382.685
9	76.1573	335.742	382.514

Table 8. RMSE for TKD, NTD and INTD from rank 1 to 9.

6.3. Algorithms metrics

6.3.1. Relative Mean Square Error (RMSE)

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

$$\xi\left(\hat{\mathcal{X}}\right) = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\left\|\hat{\mathcal{X}}_q - \mathcal{X}_q\right\|_F^2}{\left\|\mathcal{X}_q\right\|_F^2}, \tag{20}$$

where $\mathcal{X}_q$ represents the q-th MSI from a dataset with Q MSIs and $\hat{\mathcal{X}}_q$ its corresponding reconstruction computed by (4).

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 8. Furthermore, the fall of the error is very slow as the 3-rank of the decomposition increases.

6.3.2. Loss function

To quantify the convergence of the iterative process in a supervised classification algorithm, it can be used a metric of difference between the lables $\mathbf{Y}$ and its corresponding prediction $\hat{\mathbf{Y}}$ as

$$MSE(\hat{\mathbf{Y}}) = \sum \left\|\hat{\mathbf{Y}} - \mathbf{Y}\right\|_F^2 \tag{21}$$

Figure 14 shows the behavior of the loss function for each decomposition as well as for the original Sentinel-2 dataset. Figure 14d 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.

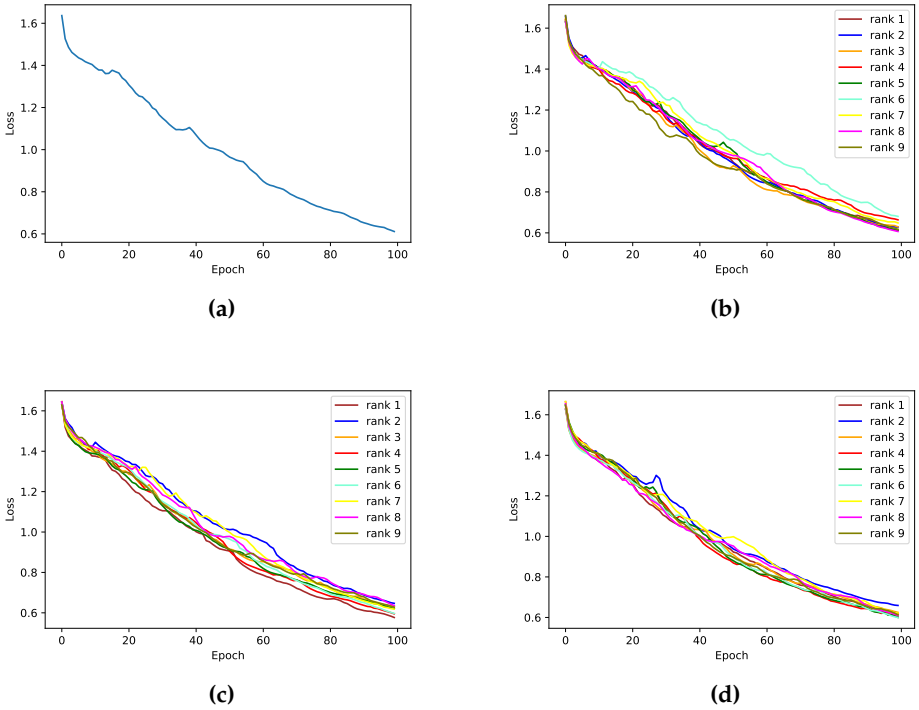


Figure 14. CNN Loss function with a) Sentiel-2 original dataset, b) TKD, c) NTD, and d) INTD.

6.4. Evaluation metrics

6.4.1. Cohen's Kappa Coefficient

Cohen's kappa coefficient is a very robust metric used to measure reliability of multi-class and imbalanced class classification algorithms []. It is computed by

$$\kappa = \frac{\rho_o - \rho_e}{1 - \rho_e} \tag{22}$$

where ρ_o is the observed agreement, and ρ_e is the expected agreement. This metric will always produce values less than or equal to 1, where 1 means perfect agreement. Negative values indicate no agreement. i.e., futile classification.

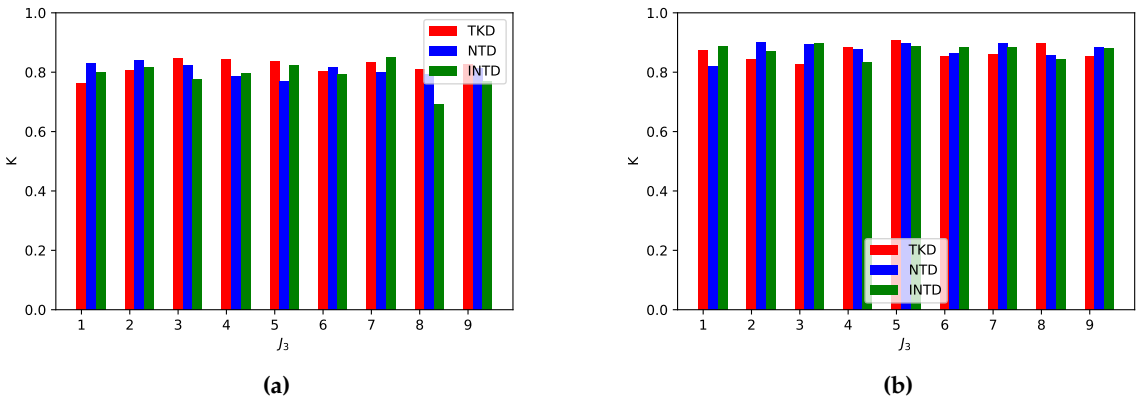


Figure 15. Kappa coefficient for TKD, NTD and INTD, a) Sentinel-2 dataset and b) Indian Pines

Actual \ Predicted	Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	41446	1521	541	112	2522	0.8982
Shadow	19	2710	4	13	53	0.9682
Cloud	1721	420	13592	73	340	0.8418
Water	3	56	0	7180	2	0.9915
Vegetation	4965	1117	831	100	84499	0.9233
Recall	0.8606	0.4653	0.9080	0.9601	0.9666	0.9120

Table 9. Confusion matrix with the original Sentinel-2 MSI dataset as input of the CNN.

6.4.2. Pixel Accuracy (PA)

The PA metric is used to compute a ratio between the amount of correctly classified pixels and the total number of pixels as follows. Given a confusion matrix relating the True Positive (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN), the PA is computed by

$$PA = \frac{\sum_{c=1}^{C} \tau_{cc}}{\sum_{c=1}^{C} \sum_{d=1}^{C} \tau_{cd}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

where c is the number of class for $c = 1, \ldots, C$ and τ_{cc} is the amount of pixels of class c correctly assigned to class c i.e., TP and TN, and τ_{cd} is the amount of pixels of class c inferred to belong to class d. Despite this metric is wide used, it is not a totally fair metric for imbalanced classes. In this work this metric is used for comparison. However, we also used metrics more in line with multi class classification tasks.

6.4.3. Precision

Another metric used in this work as a performance evaluation metric in multiclass classification is precision. This metric measures the average of pixels from class c correctly classified. It is computed with the following equation

$$\Psi_c = \frac{\tau_{cc}}{\sum_{d=1}^{C} \tau_{cd}} = \frac{TP}{TP + FP} \tag{24}$$

where Ψ_c denotes the precision of class c.

6.4.4. Recall or Sensitivity

Recall, or also known as sensitivity is a metric that indicates the proportion of pixels classified as class c that actually belong to class c. It is computed with the following equation

$$v_c = \frac{\tau_{cc}}{\sum_{d=1}^{C} \tau_{dc}} = \frac{TP}{TP + FN} \tag{25}$$

where v_c denotes the recall of class c for $c = 1, \ldots, C$.

6.4.5. F1 Score

In order to summarize precision and recall in one only metric, the F1 score is used, which is computed by

$$F1 = \frac{2\Psi v}{\Psi + v} \tag{26}$$

Actual \ Predicted	Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	42214	925	752	137	2114	0.9148
Shadow	20	2718	1	25	35	0.9710
Cloud	2210	299	13176	155	306	0.8160
Water	5	49	0	7187	0	0.9925
Vegetation	10182	939	798	315	79278	0.8663
Recall	0.7727	0.5513	0.0.8946	0.9191	0.9699	0.8824

Table 10. Confusion matrix with the original Sentinel-2 MSI dataset as input of the CNN.

Actual \ Predicted	Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	41734	1340	437	127	2504	0.9044
Shadow	46	2663	9	9	72	0.9514
Cloud	2209	161	13310	40	426	0.8243
Water	5	95	0	7141	0	0.9861
Vegetation	5617	651	1104	325	83815	0.9158
Recall	0.8412	0.5423	0.8956	0.9344	0.9654	0.9073

Table 11. Confusion matrix with the NTD core tensor as input of the CNN.

381 0.8

Actual \ Predicted	Soil	Shadow	Cloud	Water	Vegetation	Precision
Soil	40864	705	601	85	3887	0.8856
Shadow	24	2692	8	15	60	0.9617
Cloud	548	274	14651	39	634	0.9074
Water	7	74	1	7154	5	0.9879
Vegetation	5162	1185	1076	179	83910	0.9169
Recall	0.8768	0.5460	0.8967	0.9574	0.9481	0.9110

Table 12. Confusion matrix with the INTD core tensor as input of the CNN.

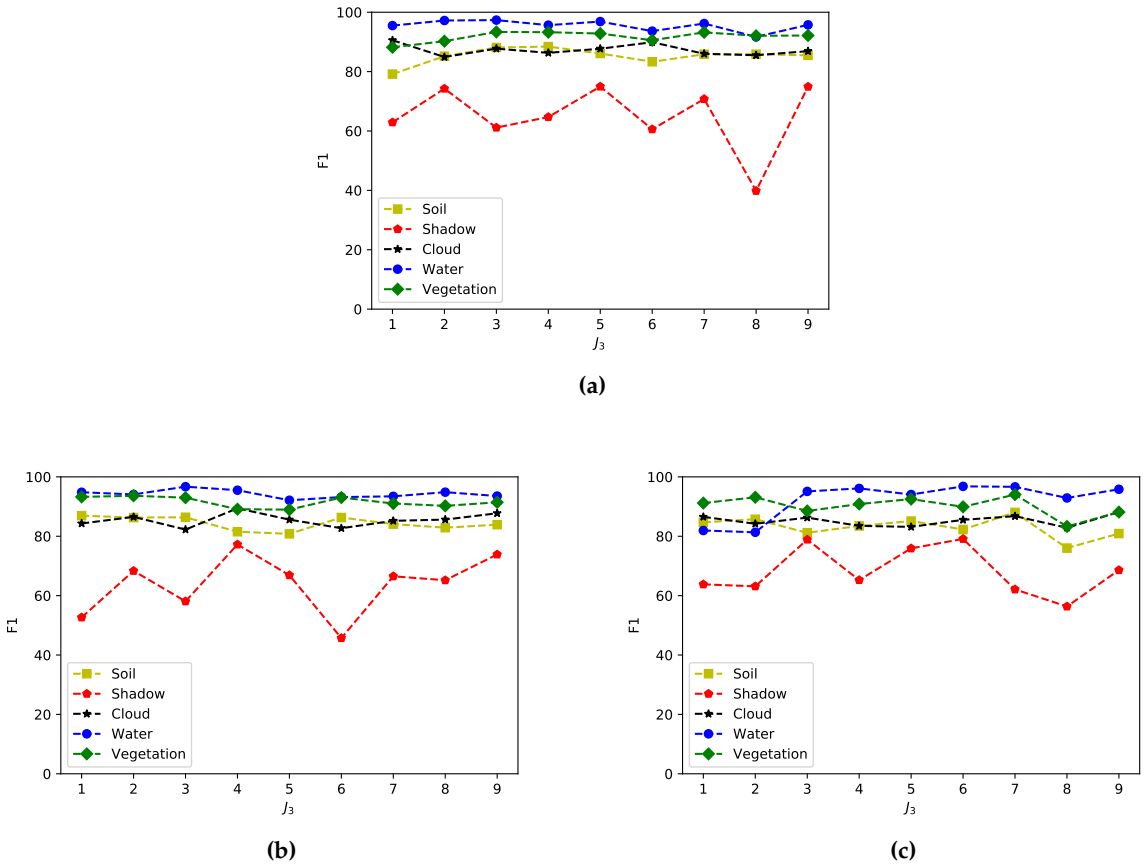


Figure 16. F1 score for Sentinel-2 dataset with a) TKD, b) NTD, and c) INTD.

Table 13. Quantitative results[1] for the Sentinel-2 test 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 recontruction error, average processing time per scenario, PA and Kappa's coefficient results for $J_3 = 1, \ldots, 9$.

J_3	TKD					NTD					INTD				
	ξ	**Time (s)**	**PA**	κ	**MCC**	ξ	**Time (s)**	**PA**	κ	**MCC**	ξ	**Time (s)**	**PA**	κ	**MCC**
1	398.95	9.05	90.15	**0.8720**	0.7832	398.95	10.02	**92.21**	0.8350	0.7677	398.95	9.52	89.82	0.8372	0.8105
2	169.84	17.86	86.84	0.8032	0.8444	329.92	19.85	87.93	0.7462	0.8699	409.54	18.32	84.19	0.7864	0.8228
3	106.04	32.54	87.12	0.7942	0.8201	310.10	35.79	87.39	0.7870	0.7856	413.29	32.96	86.83	0.7932	0.7776
4	73.80	59.10	86.07	0.8548	0.8458	309.29	60.02	91.20	0.8310	0.8357	414.55	60.05	89.75	0.7737	0.7957
5	55.63	105.72	89.54	0.7942	0.7886	304.49	105.83	88.24	**0.8585**	0.7683	414.92	106.82	**91.47**	0.8292	0.7276
6	36.85	174.19	85.39	0.8558	0.8164	303.61	180.04	91.21	0.8256	0.7962	415.82	179.89	89.30	0.7645	0.8228
7	26.65	282.97	86.02	0.8418	0.8486	293.72	283.35	90.50	0.8405	0.8424	416.41	284.12	90.39	0.7743	0.8457
8	12.24	351.40	**91.73**	0.8144	0.8312	280.50	350.02	88.66	0.8335	0.7556	414.42	350.74	89.89	**0.8627**	0.7825
9	9.06	415.69	89.35	0.7837	0.7943	280.38	408.21	86.87	0.8106	0.8548	414.35	407.05	88.33	0.8251	0.7337

[1] For the original MSI: Time = 397.21 s, PA = 0.9120, κ = 0.8552 and MCC = 0.8050.

Table 14. Quantitative results[1] for the Indian Pines 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 recontruction error, average processing time per scenario, PA and Kappa's coefficient results for $J_3 = 1, \ldots, 10$.

J_3	TKD				NTD				INTD			
	ξ	Time (s)	PA (%)	κ	ξ	Time (s)	PA (%)	κ	ξ	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 2 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 6).

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 9–11) 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 15, 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 **??** and **??** allow us to make a direct 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 analyisis 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 5 we can conclude that the proposed integer non-negative approximation

8.1. Open issues

- Compress not only the input data, but also the CNN network to reduce overall complexity and or create new ANN architectures for specific RS-MSI or HSI image applications.
- Instead of the HOOI algorithm, use greedy HOOI and other algorithms that determine the core tensor for a broad comparison.
- Instead of the HOOI algorithm, use greedy HOOI and other algorithms that determine the core tensor for a broad comparison.

For classification purposes, the use of other machine learning algorithms, such as SVM and 498 Random Forest. 499 • Increase the input data with more scenarios and their corresponding ground truth to a deeper 500 study of the behaviors of several classifiers included those based on ANN and the scope of the 501 TD methods. 502 • Denoising the original input data for an improvement of the new subspace of reduced 503 dimensionality.

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 XXXXXXX.

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

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. Gao, B. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sens. Environ.* **1996**, *58*, 1–6.
7. 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.
8. Hearst, Marti A. Support Vector Machines. *IEEE Intell. Syst. J.* **1998**, *13*, 18–28.

9. 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.
10. 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.
11. 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 ,⋯,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.