

## Article

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

Josué López <sup>1,\*</sup>, Deni Torres <sup>1</sup> and Clement Atzberger <sup>2</sup>

<sup>1</sup> Center for Research and Advanced Studies of the National Polytechnic Institute, Telecommunications Group, Av del Bosque 1145, Zapopan 45017, Mexico; dtorres@gdl.cinvestav.mx

<sup>3</sup> University of Natural Resources and Life Science, Institute of Geomatics, Peter Jordan 82, Vienna 1180, Austria; clement.atzberger@boku.ac.at

\* Correspondence: josue.lopez@cinvestav.mx

Version November 9, 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 [1]. The insertion of tensor-based algorithms for these types of tasks inspired a change in several areas, such as image processing [2].

Most of the researches 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 [3], mineralogy [4], agriculture [5], radar images [6] and, above all, remote sensing multi- and hyper-spectral images [7] can be represented, by nature, as third-order tensors.

Spectral images make certain image processing tasks much easier [8]. Recently, the use of this type of data has grown exponentially in various areas such as agriculture [9], medical analysis [10], biomedical [11], natural disaster prediction [12], security affairs [13], 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 [14].

32 Few years ago, several unsupervised classification and segmentation algorithms [] were  
33 developed, taking advantage of the properties that spectral data produce. Subsequently, with the  
34 introduction of supervised machine learning algorithms such as SVM [], kNN [] and ANN [], it was  
35 found that, under certain conditions, there is a direct relationship between the number of bands used  
36 and the performance of these algorithms []. However, with the aim of improving results, neural  
37 network models evolved into deep neural networks []. This caused the computational complexity to  
38 rise considerably and spectral image processing was not easily achievable. The foregoing requires  
39 having robust computer equipment to achieve competitive results in time.

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

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

#### 49 1.1. Previous work

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

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

60 On the other hand, there was also the possibility of using multispectral images due to the small  
61 number of spectral bands, which still made efficient results in classification without dimensionality  
62 reduction achievable, as done in [11], [18], [21] and [?]. However, the need to increase classification  
63 performance forces researchers to use data with more features that favor and aid the classification of  
64 various classes, which are difficult to differentiate with little spectral data. Thus, more recent researches  
65 have decided to use hyperspectral images with tensor decompositions, which has increased the results  
66 in classification accuracy [26], [27], [29], [?] and [?].

67 Recently, Sayeh et al. [?] published a work close to our research. They proposed a non-negative  
68 tensor decomposition of hyperspectral images but, different to our research, they try to preserve certain  
69 spatial-spectral features into the so called abundance maps, i.e. the projection matrices, while this work  
70 pursues to preserve the nature of the image just compressing the main information in the positive core  
71 tensor.

72 Table 1 summarizes some of the most cited related papers, which deal with the  
73 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. et al. [] (2019)	HSI	NTD	Spatial-Spectral	3D-CNN
<b>Our framework</b>	<b>MSI/HSI</b>	<b>iNTD/NTD</b>	<b>Spectral</b>	<b>CNN</b>

### 74 1.2. Motivation

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

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

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

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

### 94 1.3. Contribution

95 Unlike previous works, this work seeks to adapt the data in a more efficient way to the input of  
 96 deep convolutional networks. Convolutional network models are designed to extract and interpret all  
 97 the spatial properties of an image by moving the kernels over the input data []. Therefore, producing  
 98 uncorrelated data in space and spectrum, would make harder the interpretation of the data in the  
 99 convolutional network [? ]. Thus, the proposed framework maintains the integer and nonnegativity  
 100 tensor nature of the spectral images and the spatial dimensionality to preserver spatial-spectral  
 101 correlation of the data while reducing the spectral dimensionality, in order to decrease computational  
 102 load in the pixel-wise classification process. Besides, we approach the problem of setting the  
 103 compression range selection hyperparameters by the measurement of divergences, which aids to  
 104 estimate a best rank approximation.

105 We can summarize the contribution of this work with the following two points:

- 106 1. The framework INTD1-CNN proposed in this work, develops a new strategy to improve  
107 performance of semantic segmentation convolutional neural networks by finding suitable  
108 tensor data, preserving spatial correlation and values in the set of the natural numbers while  
109 compressing the spectral domain and in turn decreasing computational load.
- 110 2. Furthermore, this work proposes a strategy for defining the range in mode 3 of the compression  
111 models based on the Tucker decomposition from the information theory point of view.
- 112 3. This work also presents an exhaustive performance analysis measuring and comparing its  
113 efficiency with the most popular metrics, i.e., as pixel accuracy (PA), also PA in function of the  
114 number of new tensor bands, precision, recall, F1, orthogonality degree of the factor matrices  
115 and the core tensor, reconstruction error of the original tensor, and execution time.

116 The remainder of this work is organized as follows. Section ?? introduces tensor algebra notation  
117 and basic concepts to familiarize the reader with the symbology used in this paper. Section ?? presents  
118 the problem statement of this work and the mathematical definition. In Section ??, CNN theory is  
119 described for classification and semantic segmentation. Section ?? presents the framework proposed  
120 for compression and semantic segmentation of spectral images. Experimental results are presented in  
121 Section ?? . Finally, Sections ?? and 8 present a discussion and conclusions based on the results obtained  
122 in the experiments.

## 123 2. Notation and definitions

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

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

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

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

### 136 2.1. Matricization

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

### 139 2.2. Inner Product

140 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;  
141 i.e.,  $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} a_{i_1 \dots i_N} b_{i_1 \dots i_N}$ .

### 142 2.3. N-Mode Product

143 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  
144 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].

### 145 2.4. Rank-One Tensor

146 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  $\mathbf{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  denoted  $\text{rank}_n(\mathbf{X})$ , is the column rank of  $\mathbf{X}_{(n)}$ ; i.e., the dimension of the vector space spanned by the mode- $n$  fibers. Hence, if  $R_n \equiv \text{rank}_n(\mathbf{X})$  for  $n = 1, \dots, N$ , we can say that  $\mathbf{X}$  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 $\mathbf{a}_{i_1 i_2 \dots}$	Column, row and tube fibers of the third order tensor $\mathcal{A}$
$\mathbf{A}_{i_1 \dots}, \mathbf{A}_{:i_2 \dots}, \mathbf{A}_{\dots :i_3}$	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 matrix-based singular value decomposition, 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  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$  and three positive indices  $J_1, J_2$  and  $J_3$ , find a core tensor  $\mathbf{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$  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:

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

where  $\mathcal{E}$  denotes the approximation error tensor. The core tensor  $\mathbf{G}$  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 –  $(R_1, R_2, R_3)$ .

We can also denote the TKD using the matricization approach and express it by

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

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

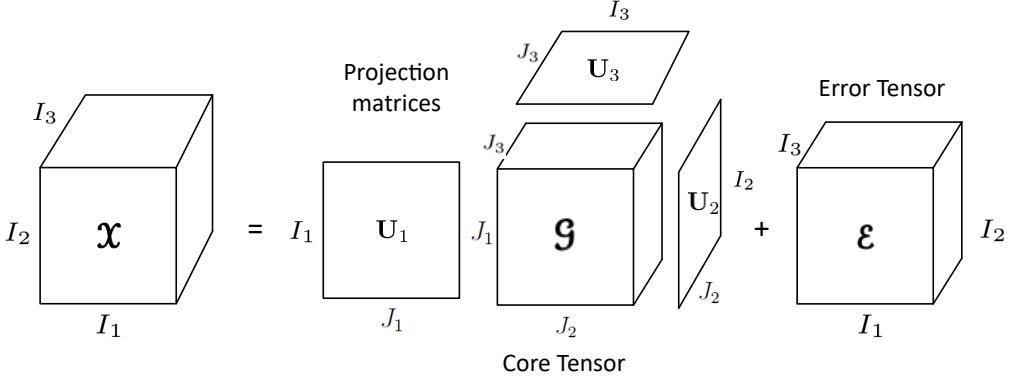
$$\mathbf{X}_{(3)} = \mathbf{U}^{(3)} \mathbf{G}_{(3)} (\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})^T \quad (3c)$$

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

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

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

where  $\hat{\mathbf{X}}$  is the reconstructed tensor.



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

Then, we can acquire the core tensor  $\mathbf{G}$  by the multilinear projection

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

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

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

and  $\|\cdot\|_F$  represents the Frobenius norm. To compute the best 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 orthogonal matrices are known, so that the core tensor is obtained with (5). Then, it maximizes the cost function

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

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

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

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  $\mathbf{W}_{(n)}^{(-n)}$  for  $n = 1, 2, 3$  [?].

### 3.1.1. Non-negative Tucker Decomposition (NTD)

The NTD is a decomposition based on the Tucker model. It develops a new tensor factorization method with nonnegativity constraints []. For the third-order case, the NTD, as defined by Cichocky [15], can 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  $\mathbf{G} \in \mathbb{R}_+^{I_1 \times I_2 \times I_3}$  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 rank approximation of a nonnegative tensor can be computed by an iterative algorithm as HOOI, maximizing the cost function

195 given in equation 7. Algorithm 1 shows the HOOI algorithm for a NTD.

---

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

---

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

---

198 **4. Problem phenomenology**

199 **4.1. Spectral Imagery**

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

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

211 **4.2. Problem Statement**

212 Let  $\mathbf{X} \in \mathbb{N}^{I_1 \times I_2 \times I_3}$  be a spectral image represented as a third-order tensor, and  $\mathbf{Y} \in \mathbb{N}^{I_1 \times I_2}$  its  
213 corresponding ground truth matrix for a specific number of classes  $C$ . Find the best rank- $(R_1, R_2, R_n)$   
214 approximation and its core tensor  $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2 \times J_3}$ , through Non-negative Tensor Decompositions. The  
215 rank of the decomposition  $\text{rank}_n(\mathbf{X})$  is set as  $\text{rank-}(I_1, I_2, J_3)$ , where  $J_3 < I_3$ . This built the input space of  
216 a pixel-wise classification using CNNs and produce an output matrix  $\hat{\mathbf{Y}}$  of predicted classes, achieving  
217 competitive performance metrics for pixel-wise classification while decreasing computational load in  
218 the classification process.

219 **4.3. Mathematical Definition**

220 We can mathematically define the problem statement described above as an optimization problem.

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

## 221 5. Methodology

222 The following subsections described the methodology followed for the framework propose in this  
 223 work. We can summarize the big picture in three steps: the HSI modeling, the tensor decomposition,  
 224 the classifier and the decision making.

225 *5.1. Tensor modeling*

226 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 space of the natural  
 227 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  
 228 by the Linear Mixing Model (LMM) as follows

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

229 where  $\alpha_c$  is the contribution of material  $c$  at pixel  $i_1 i_2$ ,  $\mathbf{m}_c$  denotes the endmember of a specific material  
 230  $c$ , and  $\boldsymbol{\eta}$  represents an additive noise vector. The abundance vectors  $\alpha_{i_1 i_2}$  must always satisfy two  
 231 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,  
 232  $\sum_{c=1}^C \alpha_{i_1 i_2 c} = 1$ . Figure 2a shows the spectral signatures for the Indian Pines dataset.

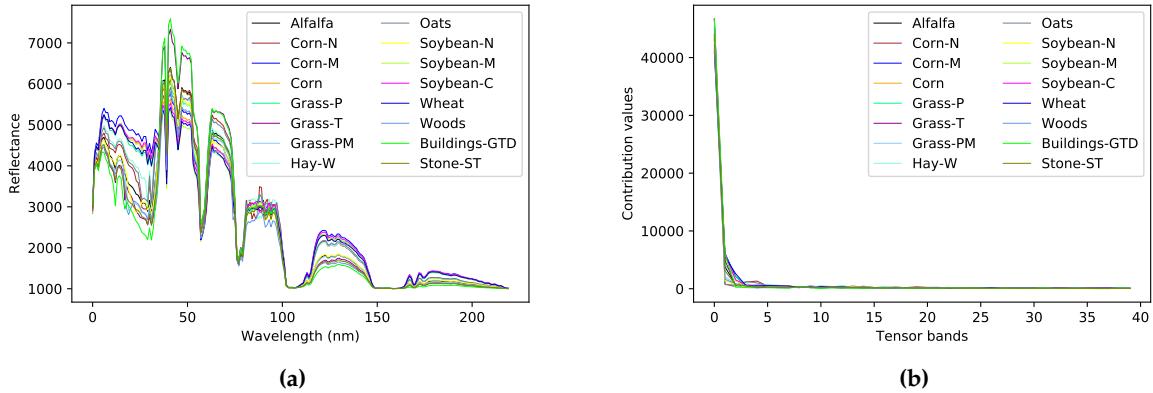
233 *5.2. Tensor factorization*

234 Consider  $\mathbf{Y} \in \mathbb{C}^{I_1 \times I_2}$  as the matrix of actual classes corresponding to our dataset  $\mathbf{X}$ , and  $\hat{\mathbf{Y}} \in$   
 235  $\mathbb{C}^{I_1 \times I_2}$  as the prediction matrix, where  $\mathbb{C}$  defines the set of  $C$  different classes. In order to reduce  
 236 data dimensionality of the input dataset  $\mathbf{X}$  while keeping classifier performance, we propose to  
 237 use the restricted NTD denoted as  $\mathcal{T}$ , producing a core tensor  $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2 \times J_3}$  and  $n$  factors matrices  
 238  $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ , expressed as

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

239 where the decomposition is restricted to preserve the spatial domain and to be only in the 3rd-mode  
 240 by the Tucker1 model

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



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

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

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

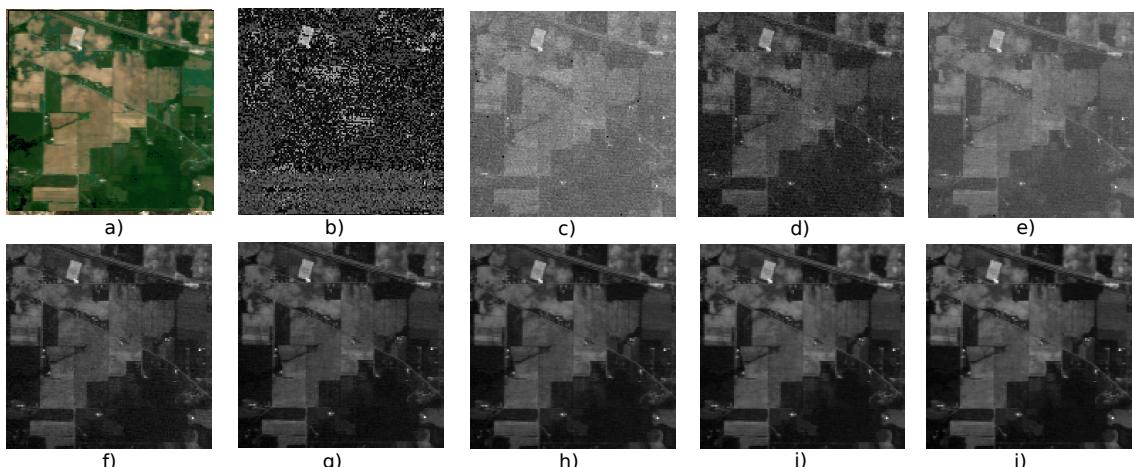
where  $\beta_c$  is the contribution of material  $c$  at pixel  $i_1 i_2$  and  $\mathbf{s}_c$  denotes the endmember of a specific material  $c$ . We can see in Figure 2b the new tensor bands values for each class.

We also propose a variant of the NTD that does an integer decomposition, i.e., 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.

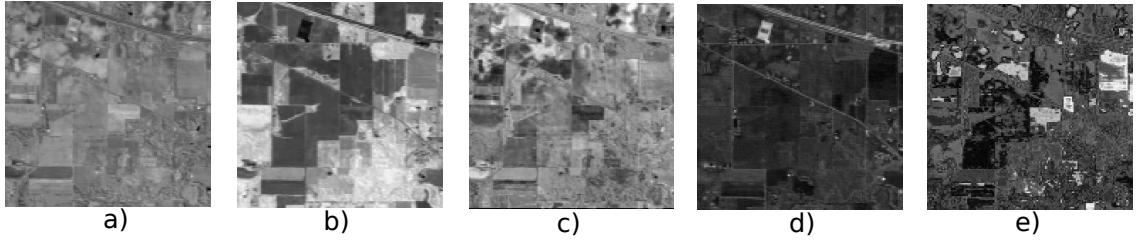
### 5.3. Classifier

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  $\mathbf{g} \in \mathbb{N}^{I_1 \times I_2 \times J_3}$ , with  $J_3 < I_3$ , and its corresponding ground truth  $\mathbf{Y}$  form the input tuple of the classifier  $\Theta$ , which produce a predicted label for each element of the input, i.e.,

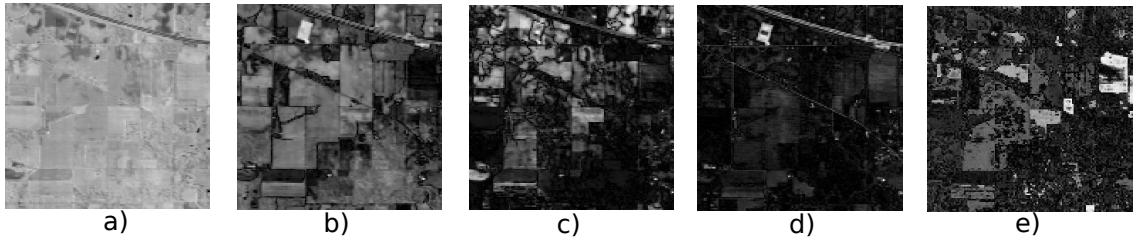
$$(\mathbf{g}, \mathbf{Y}) \xrightarrow{\Theta} \hat{\mathbf{Y}} \quad (14)$$



**Figure 3.** Indian Pines dataset a) True color image b) - j) 1st to 9th spectral band.



**Figure 4.** TKD tensor bands of the Indian Pines dataset a) - e) 1st to 5th tensor band.



**Figure 5.** NTD tensor bands of the Indian Pines dataset a) - e) 1st to 5th tensor band.

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

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

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

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

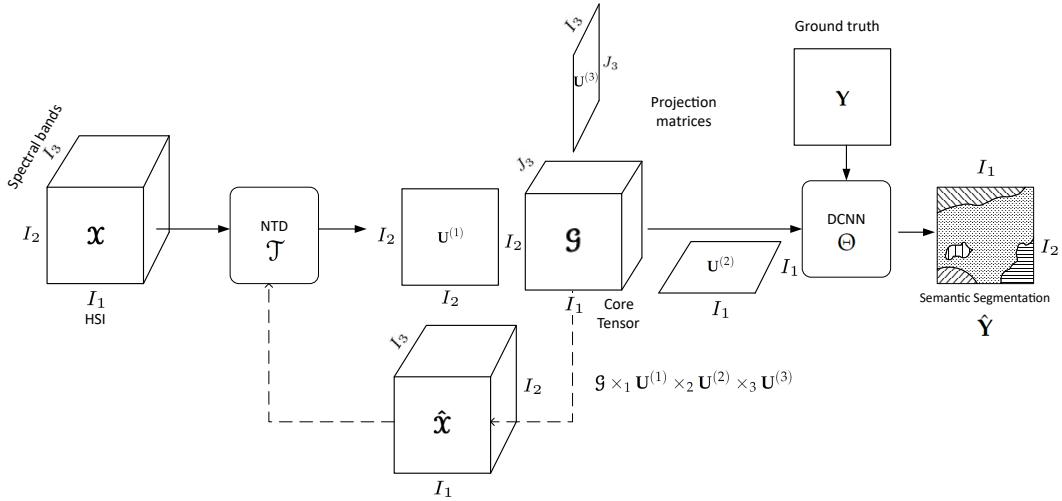
257     where  $H(y, p)$  denotes the cross entropy of targets  $y$  with a probability  $p$ .

258     We use the softmax function as the output of our classifier, to represent the probability distribution  
 259     over  $C$  different classes. Formally, the softmax function is given by

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

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

263     In this paper, we aim to feed supervised classifiers, based on 3D-CNN, with a lower dimensionality  
 264     tensor than the original dataset. This has three particular motivations: 1) to avoid overfitting the  
 265     DCNN, 2) to reduce the computational complexity, and 3) keep the classifier performance while  
 266     reducing the execution time. Figure 6 shows the big picture of the framework proposed.



**Figure 6.** Big picture of the framework proposed.

267 *5.4. Decomposition analysis*

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

272 The second challenge is the search for the rank – ( $J_1, J_2, J_3$ ). As we want to preserve the spatial  
 273 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  
 274 decrease considerably. Both decisions are made under a criterion under the probabilistic point of view.

275 The Kullback-Leibler divergence is a metric for quantifying the difference between the probability  
 276 distributions of the original MSI  $X$  and the core tensor  $G$  as

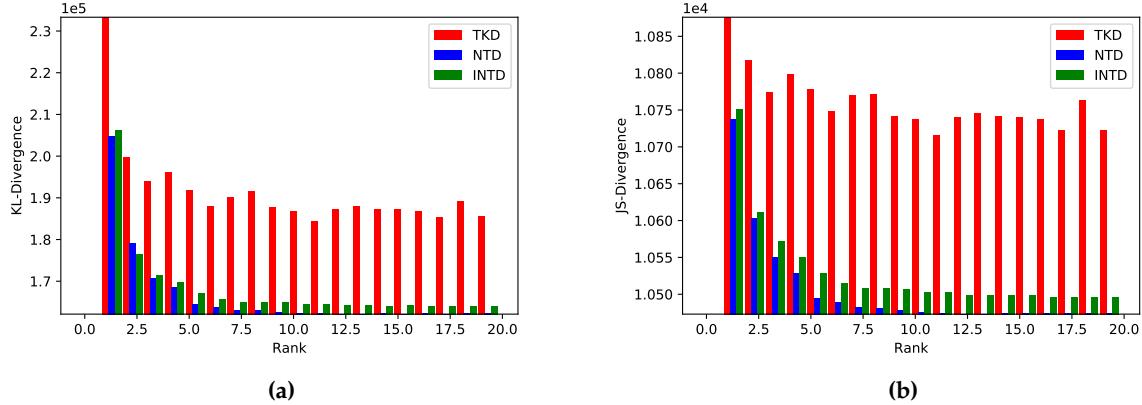
$$D_{KL}(X\|G) = \sum_{i=1}^I p_i(x) \log \frac{p_i(x)}{g_i(x)} \quad (18)$$

277 where  $D_{KL}(X\|G)$  represents the KL divergence of the two probability distributions. Figure 7a shows  
 278 the results of this metric for the three decomposition used in this work.

279 Besides, a symmetric version of the KL divergence is used, the Jensen Shanon divergence. This is  
 280 another method of measuring the similarity between two probability distributions. Defining  $M = \frac{X+G}{2}$ ,  
 281 we can write the JS divergence as

$$D_{JS}(X\|G) = \frac{1}{2}D_{KL}(X\|M) + \frac{1}{2}D_{KL}(G\|M) \quad (19)$$

282 where  $D_{JS}(X\|G)$  represents the JS divergence of the probability distributions  $X$  and  $G$ . Figure 7b  
 283 shows the JS divergence for the TKD, NTD and INTD.



**Figure 7.** Divergences between the input dataset and the core tensor generated by TKD, NTD and INTD, a) KL-divergence and b) JS-divergence.

## 284 6. Experimental Results

### 285 6.1. Input Data

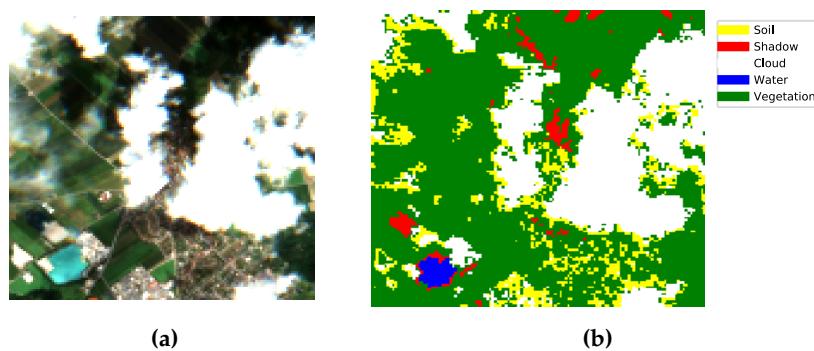
286 For this work, we chose three of the most popular multi- and hyperspectral dataset for  
287 classification.

#### 288 6.1.1. Sentinel-2

289 This dataset propose by Lopez et al. [?] is composed of RS Sentinel-2 scenarios from central  
290 Europe. It has 100 scenarios for the training space and 10 scenarios for testing, all of them with  
291  $128 \times 128$  pixels with spatial resolution of  $20m^2$  and 9 spectral bands in the range  $490 - 2190nm$ .  
292 The labels are semi-manually assigned for five classes of interest: vegetation, soil, water, clouds and  
293 shadows. Data are available in the link [Sentinel-2 Dataset](#).

**Table 3.** Average of contribution per class in Sentinel-2 dataset.

Class	Dataset	Train	Test
Soil	425,147	374,792	50,355
Shadow	45,418	42,281	3,137
Cloud	250,020	235,546	14,474
Water	218,620	195,512	23,108
Vegetation	863,035	790,269	72,766



**Figure 8.** Sentinel dataset, a) True color and b) Ground truth.

### 294 6.1.2. Indian Pines

295 This dataset is a scene produced by AVIRIS in North-western Indiana and consists of  $145 \times 145$   
 296 pixels and 224 spectral bands in the wavelength range  $0.4\text{--}2.5\mu\text{m}$ . The Indian Pines scene contains  
 297 two-thirds agriculture, and one-third forest or other natural perennial vegetation. There are two major  
 298 dual lane highways, a rail line, as well as some low density housing, other built structures, and smaller  
 299 roads. Since the scene is taken in June some of the crops present, corn, soybeans, are in early stages of  
 300 growth with less than 5% coverage. The ground truth available is designated into sixteen classes and is  
 301 not all mutually exclusive. Indian Pines data are available at [Indian Pines dataset](#). Figure 9 shows the  
 302 true color image of Salinas dataset, as well as the ground truth with each of its corresponding classes  
 303 and Table 4 the number of samples for each class.

**Table 4.** Average of contribution per class in Indian Pines dataset.

Class	Samples
Alfalfa	46
Corn-notill	1428
Corn-mintill	830
Corn	237
Grass-pasture	483
Grass-tress	730
Grass-pasture-mowed	28
Hay-windrowed	478
Oats	20
Soybean-notill	972
Soybean-mintill	2455
Soybean-clean	593
Wheat	205
Woods	1265
Building-Grass-Trees-Drives	386
Stone-Steel-Towers	93



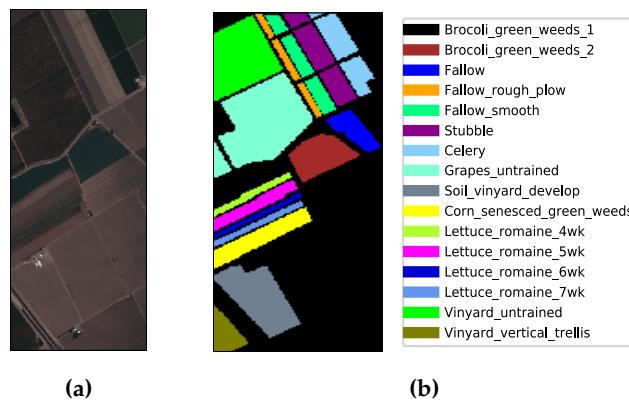
**Figure 9.** Indian Pines dataset, a) True color and b) Ground truth.

### 304 6.1.3. Salinas

305 This scene was collected by the AVIRIS sensor over Salinas Valley, California. It has  $512 \times 217$   
 306 pixels with spatial resolution  $3.7\text{m}$ , and 224 spectral bands. It includes vegetables, bare soils, and  
 307 vineyard fields. Salinas groundtruth contains 16 classes shows in table 5. Salinas data are available at  
 308 [Salinas dataset](#). Figure 10 shows the true color image of the Salinas dataset, as well as the ground truth  
 309 labeled with each of its corresponding classes and Table 5 the number of samples for each class.

**Table 5.** Average of contribution per class in Salinas dataset.

Class	Samples
Brocoli-green-weeds-1	2009
Brocoli-green-weeds-2	3726
Fallow	1976
Fallow-rough-plow	1394
Fallow-smooth	2678
Stubble	3959
Celery	3579
Grapes-untrained	11271
Soil-vinyard-develop	6203
Corn-senesced-green-weeds	3278
Lettuce-romaine-4wk	1068
Lettuce-romaine-5wk	1927
Lettuce-romaine-6wk	916
Lettuce-romaine-7wk	1070
Vinyard-untrained	7268
Vinyard-vertical-trellis	1807

**Figure 10.** Salinas dataset, a) True color and b) Ground truth.

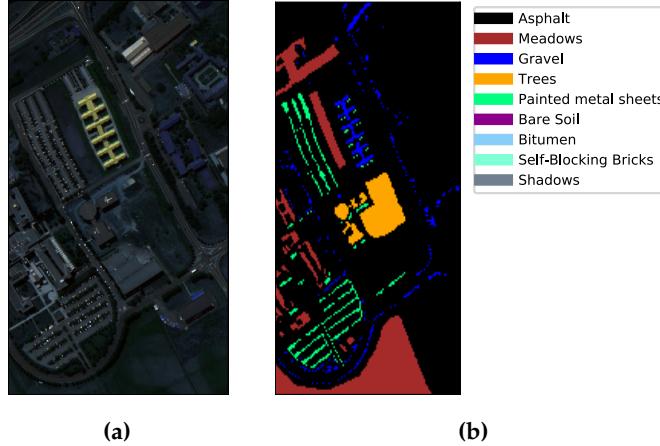
#### 310 6.1.4. Pavia University

311 This scene was collected by the ROSIS sensor over Pavia University, nothern Italy. It has  $610 \times 340$   
 312 pixels with spatial resolution  $1.3m$ . and 103 spectral bands. Pavia groundtruth contains 9 classes, some  
 313 of them described in table 6. Pavia University data are available at [Pavia University dataset](#). Figure 11  
 314 shows the true color image of the Pavia dataset, as well as the ground truth labeled with each of its  
 315 corresponding classes.

316

**Table 6.** Average of contribution per class in Pavia dataset.

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



**Figure 11.** Pavia dataset, a) True color and b) Ground truth.

In the following table 7 it is summarized the datasets used in this work as well as their spatial and spectral characteristics, the number of classes and their samples.

**Table 7.** Summary of the different dataset used for experiments in this work.

Dataset	Spatial dimensions	Bands	Classes	Samples
Sentinel-2 CNNMSI	$128 \times 128$	9	5	16,384
Indian Pines	$145 \times 145$	220	16	21,025
Salinas	$512 \times 217$	224	16	111,104
Pavia University	$610 \times 340$	103	9	207,400

### 319 6.2. CNN Specifications

320 The model used to evaluate the framework proposed in this work is Segnet [ ]. The strategie of  
 321 cross-validation was used to set the hyperparameters of the CNN getting the following

- 322 • learning rate:  $1 \times 10^{-3}$
- 323 • epochs: 100
- 324 • optimizer: Adam [ ]
- 325 • initialization: Xavier [ ]
- 326 • kernel dimensions:  $3 \times 3$
- 327 • Activation Function: ReLU / Softmax

#### 328 6.2.1. Computational Specifications

329 The software and hardware tools used to run the framework proposed are summarized as  
 330 follows

- 331 • Platform: Python 3.7
- 332 • AI Framework: Tensorflow 1.13
- 333 • GPU: NVIDIA GeForce GTX 1050 Ti
- 334 • Processor: Intel core i7
- 335 • RAM: 8GB
- 336 • SSD: 128GB / HDD: 1TB

### 337 6.3. Algorithms metrics

#### 338 6.3.1. Relative Mean Square Error (rMSE)

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

$$rMSE(\hat{\mathbf{x}}) = \frac{1}{Q} \sum_{q=1}^Q \frac{\|\hat{\mathbf{x}}_q - \mathbf{x}_q\|_F^2}{\|\mathbf{x}_q\|_F^2}, \quad (20)$$

<sup>341</sup> where  $\mathbf{X}_q$  represents the  $q$ -th MSI from a dataset with  $Q$  MSIs and  $\hat{\mathbf{X}}_q$  its corresponding reconstruction  
<sup>342</sup> computed by (4).

### <sup>343</sup> 6.3.2. Loss

<sup>344</sup> To quantify the convergence of the iterative process in a supervised classification algorithm, it can  
<sup>345</sup> be used a metric of difference between the lables  $\mathbf{Y}$  and its corresponding prediction  $\hat{\mathbf{Y}}$  as

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

<sup>346</sup> where  $\mathbf{X}_q$  represents the  $q$ -th CNNMSI from our dataset with  $Q$  MSIs and  $\hat{\mathbf{X}}_q$  its corresponding  
<sup>347</sup> reconstruction computed by (4).

## <sup>348</sup> 6.4. Evaluation metrics

### <sup>349</sup> 6.4.1. Cohen's Kappa Coefficient

<sup>350</sup> Cohen's kappa coefficient is a very robust metric used to measure reliability of multi-class and  
<sup>351</sup> imbalanced class classification algorithms []. It is computed by

$$\kappa = \frac{\rho_o - \rho_e}{1 - \rho_e} \quad (22)$$

<sup>352</sup> where  $\rho_o$  is the observed agrrement, and  $\rho_e$  is the expected agreement. This metric will always produce  
<sup>353</sup> values less than or equal to 1, where 1 means perfect agreement. Negative values indicate no agreement.  
<sup>354</sup> i.e., futile classification.

### <sup>355</sup> 6.4.2. Pixel Accuracy (PA)

<sup>356</sup> We used the PA metric to compute a ratio between the amount of correctly classified pixels and  
<sup>357</sup> the total number of pixels as follows. Given a confusion matrix relating the True Positive (TP), True  
<sup>358</sup> Negatives (TN), False Positives (FP) and False Negatives (FN), the PA is computed by

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

<sup>359</sup> where  $c$  is the number of class for  $c = 1, \dots, C$  and  $\tau_{cc}$  is the amount of pixels of class  $c$  correctly  
<sup>360</sup> assigned to class  $c$  i.e., TP and TN, and  $\tau_{cd}$  is the amount of pixels of class  $c$  inferred to belong to class  
<sup>361</sup>  $d$ . Despite this metric is wide used, it is not a totally fair metric for imbalanced classes. In this work we  
<sup>362</sup> used this metric with comparison purposes. However, we also used metrics more in line with multi  
<sup>363</sup> class classification tasks.

### <sup>364</sup> 6.4.3. Precision

<sup>365</sup> Another metric used in this work as a performance evaluation metric in multiclass classification  
<sup>366</sup> is precision. This is a metric that measures the percentage of pixels from class  $c$  correctly classified. It  
<sup>367</sup> is computed with the following equation

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

<sup>368</sup> where  $\Psi_c$  denotes the precision of class  $c$ , which is the number of TP divided by the TP plus FP.

### <sup>369</sup> 6.4.4. Recall or Sensitivity

<sup>370</sup> Recall, or also known as sensitivity is a metric that indicates the proportion of pixels classified as  
<sup>371</sup> class  $c$  that actually belong to class  $c$ . It is computed with the following equation

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

<sup>372</sup> where  $v_c$  denotes the recall of class  $c$  for  $c = 1, \dots, C$ .

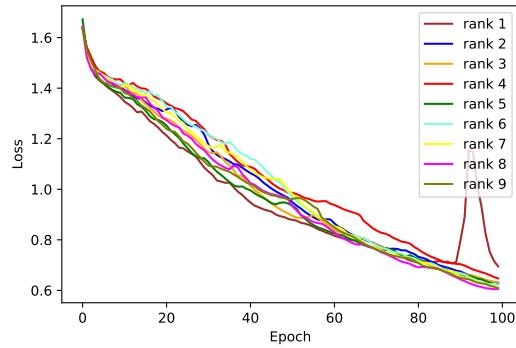
#### <sup>373</sup> 6.4.5. F1 Score

<sup>374</sup> In order to summarize precision and recall in one only metric, we use the F1 score, which is  
<sup>375</sup> computed by

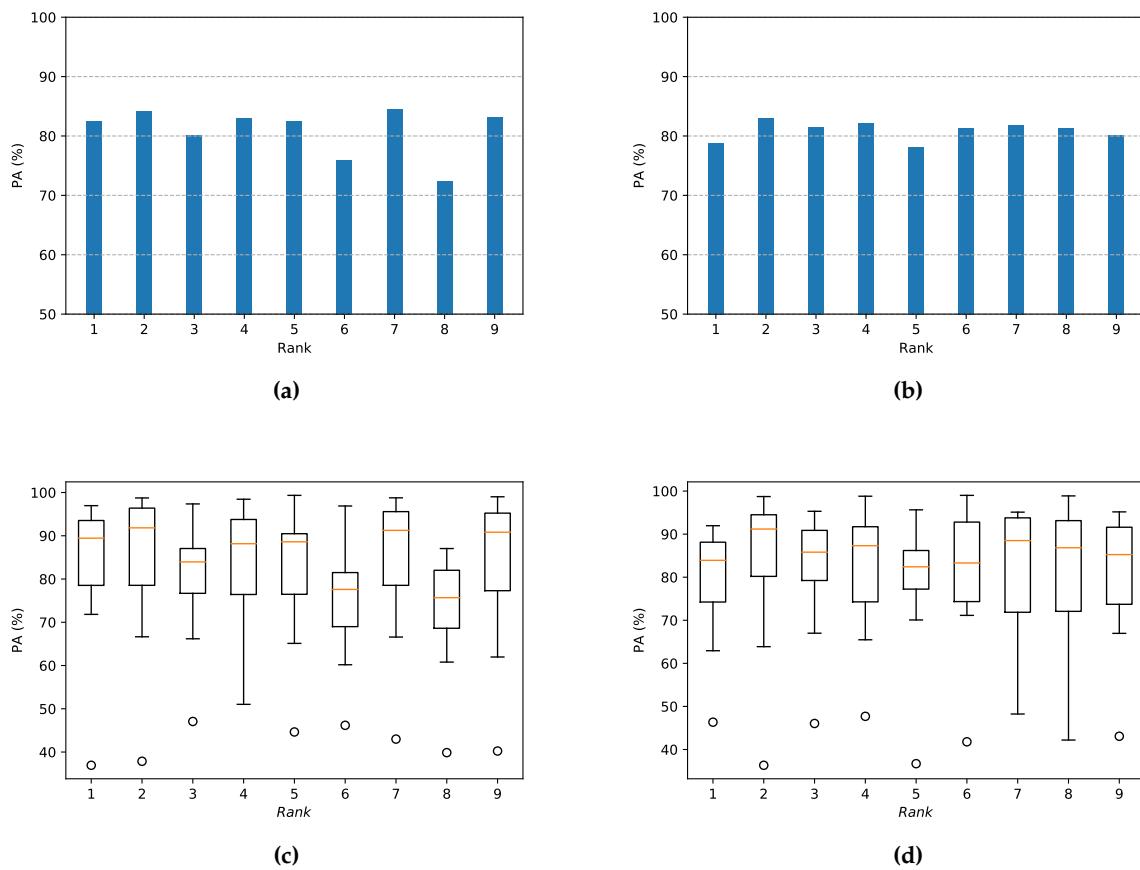
$$F1 = \frac{2\Psi v}{\Psi + v} \quad (26)$$

<sup>376</sup> This metric provides a very appropriate measure of multiclass classification for imbalanced dataset.

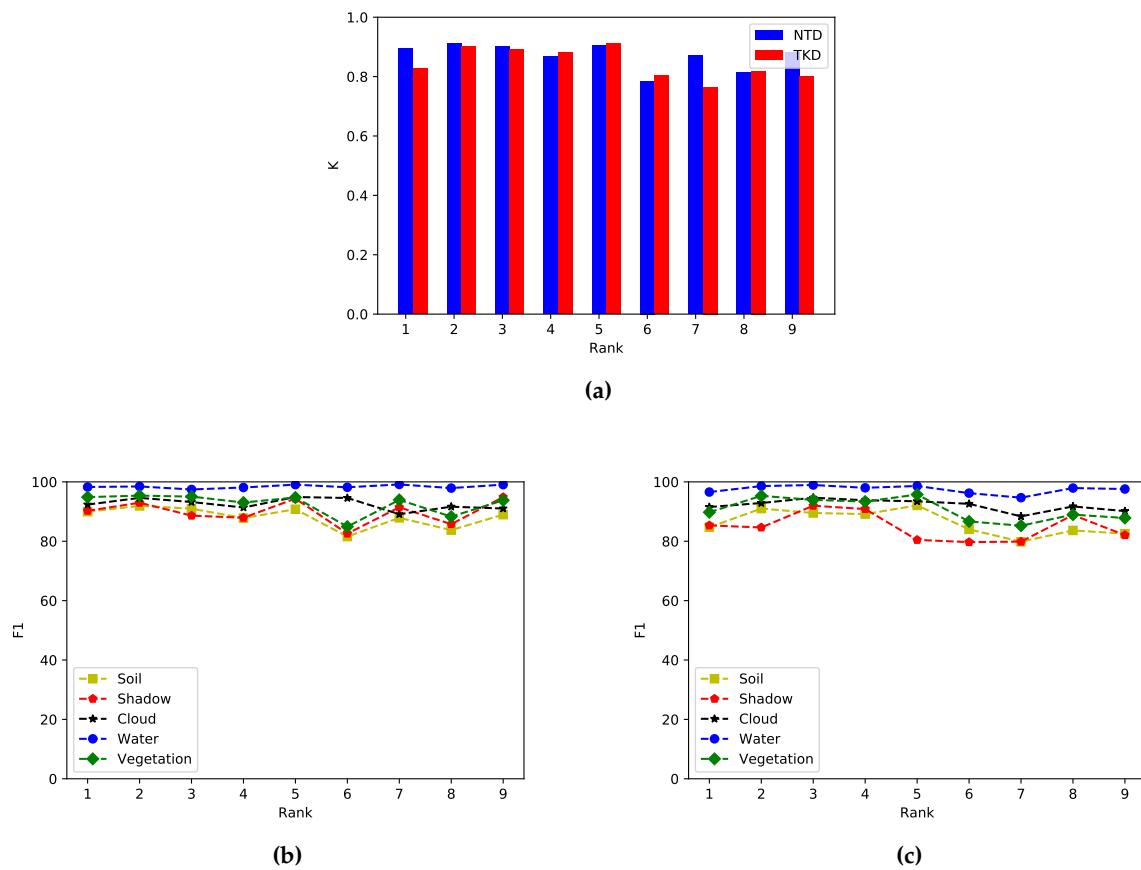
## <sup>377</sup> 7. Discussion and Comparison



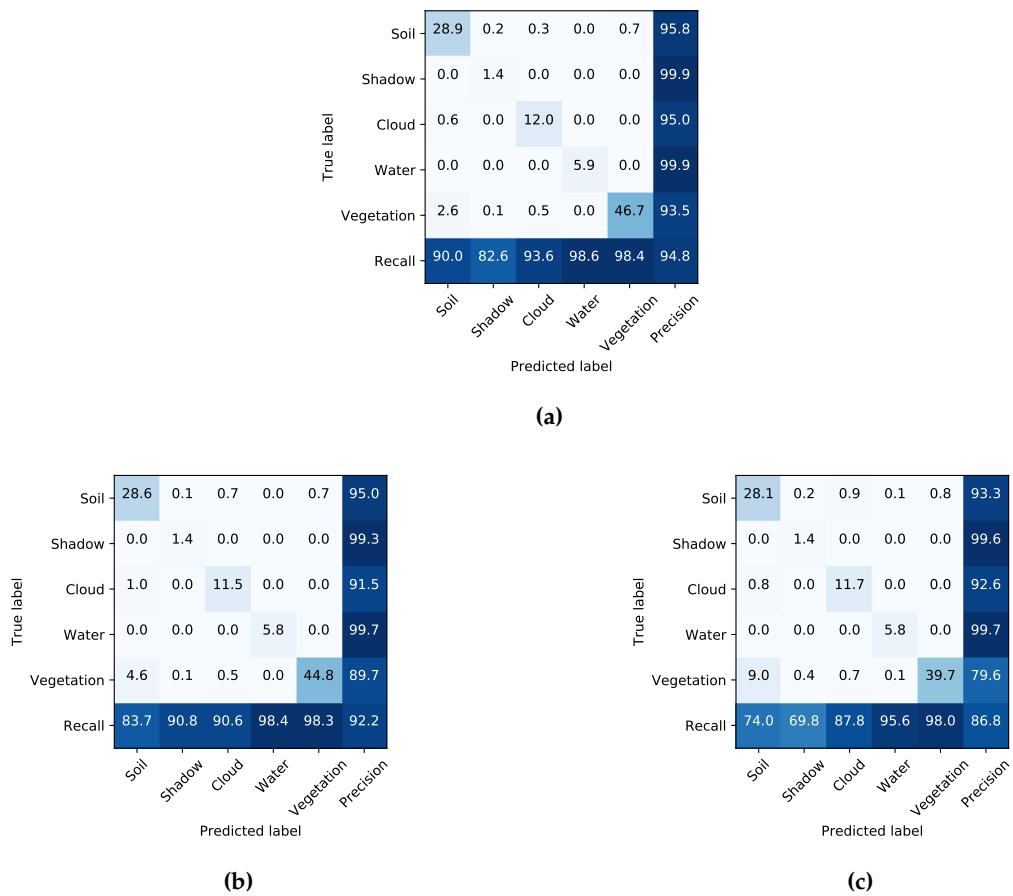
**Figure 12.** Loss function of the CNN after NTD for rank  $J_3 = 1$  to 9.



**Figure 13.** Pixel accuracy vs Rank results a) Comparative bar plot NTKD and TKD, b) Comparative bar plot NTKD and TKD c) Box and whiskers plot for NTKD, and d) Box and whiskers plot for TKD



**Figure 14.** Performance evaluation metrics. a) Kappa score comparison NTD vs TKD, b) F1 for NTD and c) F1 for TKD



**Figure 15.** Confusion matrix for a) Sentinel-2 original dataset, b) NTD, c) TKD.

## 378 8. Conclusions

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

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

384 **Acknowledgments:**

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

## 386 Abbreviations

387 The following abbreviations are used in this manuscript:

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

**390 References**

- 391 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.
- 394 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.
- 396 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.
- 398 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.
- 401 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.
- 404 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.
- 406 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.
- 408 8. Hearst, Marti A. Support Vector Machines. *IEEE Intell. Syst. J.* **1998**, *13*, 18–28.
- 409 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.
- 412 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.
- 415 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.
- 417 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.
- 419 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.
- 421 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.
- 423 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.
- 426 16. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer Verlag: New York, NY, USA, 2002.
- 427 17. Kolda, T.; Bader, B. Tensor Decompositions and Applications. *SIAM Rev.* **2009**, *51*, 455–500.
- 428 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.
- 431 19. European Space Agency. Available online: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2> (accessed on 15 July 2019).
- 433 20. Kemker, R.; Kanan, C. Deep Neural Networks for Semantic Segmentation of Multispectral Remote Sensing Imagery. *arXiv* **2017**, arXiv:abs/1703.06452.
- 435 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.
- 438 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.
- 440 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.

- 442 24. Zhang, L.; Zhang, L.; Tao, D.; Huang, X.; Du, B. Compression of hyperspectral remote sensing images by  
443 tensor approach. *Neurocomputing* **2015**, *147*, 358–363.
- 444 25. Astrid, M.; Lee, Seung-Ik. CP-decomposition with Tensor Power Method for Convolutional Neural Networks  
445 compression. In Proceedings of the BigComp, Jeju, Korea, 13–16 February 2017.
- 446 26. Chien, J.; Bao, Y. Tensor-factorized neural networks. *IEEE Trans. Neural Networks Learn. Syst.* **2018**, *29*, 1998–2011.
- 447 27. An, J.; Lei, J.; Song, Y.; Zhang, X.; Guo J. Tensor Based Multiscale Low Rank Decomposition for Hyperspectral  
448 Images Dimensionality Reductio. *Remote Sens.* **2019**, *11*, 1485.
- 449 28. Li, J.; Liu, Z. Multispectral Transforms Using Convolution Neural Networks for Remote Sensing Multispectral  
450 Image Compression. *Remote Sens.* **2019**, *11*, 759.
- 451 29. An, J.; Song, Y.; Guo, Y.; Ma, X.; Zhang, X. Tensor Discriminant Analysis via Compact Feature Representation  
452 for Hyperspectral Images Dimensionality Reduction. *Remote Sens.* **2019**, *11*, 1822.
- 453 30. Absil, P.-A.; Mahony, R.; Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*, 1st ed.; Princeton  
454 University Press: Princeton, NJ, USA, 2007.
- 455 31. De Lathauwer, L.; De Moor, B.; Vandewalle, J. On the best rank-1 and rank-(R 1 ,R 2 ,…,R N ) approximation  
456 of higher-order tensors. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1324–1342.
- 457 32. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press, 2016.
- 458 33. Sheehan, B. N.; Saad, Y. Higher Order Orthogonal Iteration of Tensors (HOOI) and its Relation to PCA and  
459 GLRAM. In Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, MN, USA,  
460 26–28 April 2007.
- 461 34. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture  
462 for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
- 463 35. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A Multilinear Singular Value Decomposition. *SIAM J. Matrix  
464 Anal. Appl.* **2000**, *21*, 1253–1278.
- 465 36. Rodes, I.; Inglada, J.; Hagolle, O.; Dejoux, J.; Dedieu, G. Sampling strategies for unsupervised classification  
466 of multitemporal high resolution optical images over very large areas. In Proceedings of the 2012 IEEE  
467 International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012.

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

469 © 2020 by the authors. Submitted to *Remote Sens.* for possible open access publication  
470 under the terms and conditions of the Creative Commons Attribution (CC BY) license  
471 (<http://creativecommons.org/licenses/by/4.0/>).