# A TENSOR DECOMPOSITION METHOD FOR UNSUPERVISED FEATURE LEARNING ON SATELLITE IMAGERY


## ABSTRACT

We introduce a tensor factorization approach to unsupervised feature learning of hyper-spectral imagery, and demonstrate its effectiveness on land type classification of publicly available datasets. We use sampled patches from hyper-spectral images to construct a tensor and then factorize it. A dictionary capturing spectral and spatial characteristics of the image is then constructed using the factorization. The final feature space is then obtained by finding coefficients of all image patches on the dictionary. The results show that this approach can produce state of the art accuracy, compared to other methods for feature learning in the classification task.



***Index Terms*—** Tensor decomposition, feature learning, hyperspectral imagery


## 1. INTRODUCTION

Hyper-spectral images, i.e. images with a high density of samples in the electromagnetic spectram, are powerful devices in identifying ground objects according to their unique spectral signature. These images are used in a veriety of applications including environmental sciences [1] and land-cover mapping. But as their spectral dimension is in the range of hundreds, a feature learning step is needed to process them. To this end, many algorithms have been proposed which on a high level can be classified as supervised and unsupervised feature learning methods. In the supervised approach, known samples are required for devising features that can then distinguish the classes. However given that hand-labling gyper-spectral data is a laborous task; we will focus on Unsupervised Feature Learning (UFL) methods, as they don't require prior knowledge of the structures present in images or training data. In the unsupervised approach only the raw data is given to the algorithm and from that alone the algorithm extracts features representative of the data. The UFL approaches include PCA based approaches such as [2],[3], various discriminant analysis based approaches such as [4], manifold learning based approaches such as [5], [6] and auto encoder based approaches such as [7],[8]. Existing approaches use local optimization methods which can get stuck in local minima. In this paper we introduce a tensor decmposition framework for formulating feature learning of hyper-spectral imagery.

Tensor decomposition methods have been employed outside the scope of satellite image processing. Frequently, the use of tensor representations focuses on de-noising and compression, rather than feature learning for unsupervised classification (i.e., clustering). [9] used non-negative tensor decomposition for dimensionality reduction of images, and demonstrated its use for unsupervised face recognition. [10] the tensor decomposition method for obtaining image descriptors. [11] introduces the TenSR Framework for sparse representation of multidimensional signals, based on Tucker factorization, which has the added degree of freedom of a "core" tensor transforming the components.

*1.0.0.1. Our contribution*

In this paper we propose a novel tensor decomposition based formulation for the task of UFL. To the best of our knowledge, this is the first tensor-decomposition formulation of unsupervised feature learning for satellite imagery classification (explain the tensor PCA). Inspired by the dictionary learning approach in this setting [12], we use an overcomplete factorization in which we encourage sparsity with respect to the patch representations. This has been used previously in the context of Tucker decompositions [13] but not, to our knowledge, in the context of decompositions into linear combinations of rank-1 tensors ("CP decomposition"). We compare the performance of our method with a state-of-the-art method using a 3d auto-encoder [8], and show that our new method achieves competetive performance.

## 2. PRELIMINARIES AND NOTATION

Here we introduce our notation for tensors and their decomposition. Tensors or multi-way arrays are functions of three or more indices $(i, j, k, \ldots)$. They are the generalization of the concept of matrices. One difference between tensors and matrices is in their decomposition. Matrix decomposition doesn't produce unique results unless it has strict constraints such as orthogonality, but tensor decomposition is unique under mild constraints. [14] provides a thorough review of tensors and their decomposition.

We use the following notation. The size of a matrix with $n$ rows and $m$ columns is denoted by $(n, m)$, and $A(:, m)$ is the $m$th column of $A$. Vectorization of a matrix along the columns is the process of concatenating all its columns into a long vector, and similarly for row vectorization.

Here we work with three-way tensors, and $(I, J, K)$ is the size of the tensor $T$ along each dimension. The Canonical Polyadic Decomposition (CPD) of $T$ with $c$ components extracts from it three matrices with sizes $(I, c)$, $(J, c)$, $(K, c)$, each of which are the concatenation of $c$ rank-1 vectors of appropriate size, as shown in Figure 1.

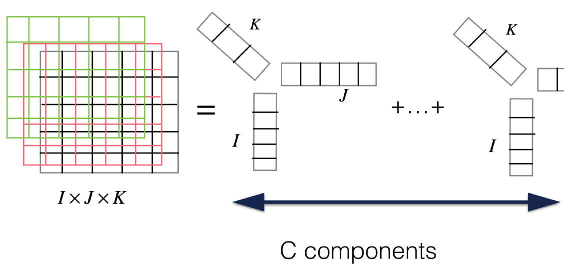


**Fig. 1**. Canonical Polyadic Decomposition with $c$ components.

## 3. TENSOR DECOMPOSITION FEATURE LEARNING

We now describe our new method for learning features using tensor decomposition. An overview is shown in Figure 2.

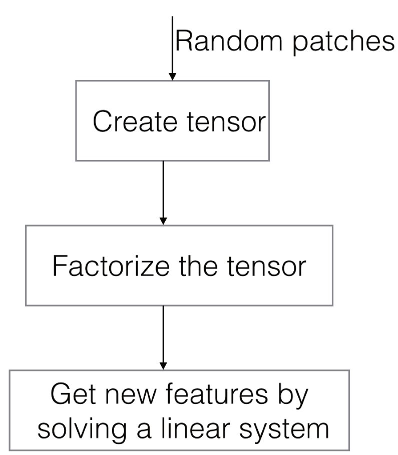


**Fig. 2**. Flowchart summary of the proposed method.

*1) Create tensor:* The input to the algorithm is a set of randomly sampled hyper-spectral patches (with uniform distribution), where the size of these patches is a user-defined parameter. We define the variables $p$ as size of the dimension of each patch, $b$ as the number of spectral bands, and $s$ as total

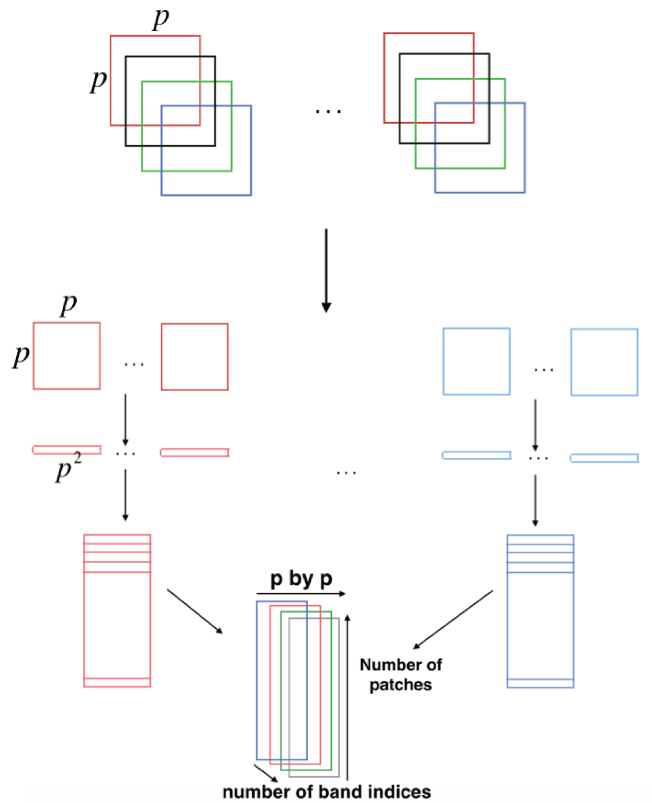


**Fig. 3**. The process of building a tensor out of randomly sampled multi-spectral patches. Colors represent the different spectral bands. Patches of each band are flattened and concatenated into a tensor.

number of the patches. Each band of the patch is vectorized to get vectors of size $p^2$. (We flatten the patch as we do not wish to factorize the spatial dimensions—we would like to be able to capture arbitrary intensity patterns in a patch with our filters.) Then a matrix of size $(p^2, s)$ is created by concatenating all the vectors as rows of the matrix. The tensor is then constructed by adding each band as the third index. These processes are shown in Figure 3.

*2) Factorize tensor:* We now have a $(p^2, s, b)$ tensor. To interpret the results of a tensor factorization with $c$ components, consider the element $(i, j, :)$, which is the vectorization of all spectral bands of the $i$th pixel in the $j$th sampled patch. The following equation holds between the original tensor and its factorization, where $T$ is the original tensor and $T_p$, $T_b$ and $T_s$ are the factors along each original dimension:

$$T(i, j, :) = \sum_c T_p(i, c) T_s(j, c) T_b(:, c)$$

From the above equation, we can see that the elements of $T_b(:, c)$ can be interpreted as a decomposition of the original spectrum of bands (in T) into $c$ indices for each spectrum (can be thought of as learning spectral indices present in the image). With this insight, we perform non-negative CPD factorization on this tensor with $c$ factors and minimizing the $l_1$-norm along all dimensions. This style of decomposition with sparse factors was considered previously for Tucker decompositions by [13]. We note that although the CPD is unique for the mimimum number of components $c$, in general we use a non-minimal value of $c$. We use the sparsity-encouraging $l_1$-minimization constraint since we want to encourage the different classes to be represented by distinct spectral indices, and the sparsity constraint discourages the construction of common patterns that can be used across classes. In particular, when we extract features using these patterns, the decompositions into patterns will be unique if they are sufficiently sparse, in spite of the possible overcompleteness with respect to this dimension.

*3) Compute new features:* After the CPD factorization, we take the $c$ components along the band dimension ($b$) as spectral indices. Now we define a system of equations that takes a new image patch and decomposes it. We define a $(p^2 b, c)$ dictionary using the tensor decomposition. Take $C_i$ to be the ith factor of $T_b$. We construct $p^2$ patterns from this one factor, according to the following formula:

$$C_{i,j} = C_i T_p(j,i) \quad j = 1, \ldots, p^2$$

Now we have $p^2$ texture patterns, corresponding the neighboring pixel's spectral indices. With this procedure we obtain $p^2$ vectors each of size $b$, whic we then concatenate into a $p^2 b$ dimension column vector. Concatenating all these column vectors in a matrice, gives us a $(p^2 b, c)$-size dictionary. Denote this dictionary by $D$.

Given a new hyper-spectral patch, we vectorize the patch and concatenate these vectors to form a new $p^2 b$ vector $V_b$, analogous to our transformation of our filters earlier. Now we have the system of constraints:

$$Dx = V_b \quad \text{s.t.} \quad x \geq 0$$

where $x$ is a $c$ by 1 column vector. This is a linear program, which can be solved by standard methods. In particular, if the solution is sufficiently sparse, it is known that it is unique [15]. Solving this system gives the decomposition of the new hyper-spectral patch over the texture patterns. We drop all but the top $n$ entries of $x$ to obtain the values of our $n$ new features for the patch. (We preserve the ordering of the remaining $n$ values.) This $n$ is a user-defined parameter.

The extended feature space as described above is then used in a standard classification method. In particular, in our experiments, we considered using the SVM (with 10-fold cross-validation) for supervised classification.

## 4. EXPERIMENTS

To Verify the method, three publicly available hyperspectral datasets were used and compared against a state of the art method ([8]), which uses 3d convolutional auto encoders for feature learning. The Indian pines data set acquired by the AVIRIS sensor is a 145 by 145 image of 224 bands. The corrected version with 220 bands was used. The Salinas valley dataset. The Pavia University data set. The patch size was set to 10, 65 components were extracted from the tensor and 50 used in dictionary. 4000 randomly sampled patches were used to construct the tensor and the classification was done with SVM and 10-fold cross-validation. 10 percent of data was used as training and the rest for testing. (MATLAB's implementation of SVM was used). In addition to normal experiment settings, were features learned from the image are used for testing we also use the other exeriment setting were a dictionary from one image is used on the other and compare it with reference. (in order to match the spectral signature of different datasets, only the common bands were used and the rest were discarded).

### 4.1. Results

Tables show the result of TD versus the 3D convolutional auto-encoder method on the datasets. Tables 1 to 4 are driven from regular experiment settings on Indian pines, Pavia university, Pavia center and Salinas datasets. Table 5 to 8 show the results of cross dictionary usage in testing for datasets.

| Class | 3DCAE | TD |
|---|---|---|
| 1 | 90.48 | 69.04 |
| 2 | 92.49 | 94.66 |
| 3 | 90.37 | 92.42 |
| 4 | 86.90 | 83.72 |
| 5 | 94.25 | 95.45 |
| 6 | 97.07 | 99.09 |
| 7 | 91.26 | 69.23 |
| 8 | 97.79 | 98.38 |
| 9 | 75.91 | 94.44 |
| 10 | 87.34 | 86.80 |
| 11 | 90.24 | 96.30 |
| 12 | 95.76 | 88.68 |
| 13 | 97.49 | 100 |
| 14 | 96.03 | 98.68 |
| 15 | 90.48 | 100 |
| 16 | 98.82 | 80.95 |
| AA(%) | 92.04 | 92.09+-1.58 |
| OA(%) | 92.35 | 96.39+-0.41 |

**Table 1**. Overall and average accuracy and class accuracies of one run and mean var for Indian pines.

The results show that the tensor decomposition methods provides more accurate results, in separating classes of forest and agriculture and overall accuracy. A note on the method's performance should be made. when comparing the class-wise accuracies of the indian pines we can see that although the overall and average accuracies of the proposed

| Class | 3DCAE | TD |
|---|---|---|
| 1 | 100.00 | 99.83 |
| 2 | 99.29 | 99.91 |
| 3 | 97.13 | 100 |
| 4 | 97.91 | 99.04 |
| 5 | 98.26 | 99.17 |
| 6 | 99.98 | 99.61 |
| 7 | 99.64 | 99.38 |
| 8 | 91.58 | 94.05 |
| 9 | 99.28 | 99.96 |
| 10 | 96.65 | 98.75 |
| 11 | 97.74 | 99.69 |
| 12 | 98.84 | 99.94 |
| 13 | 99.26 | 99.51 |
| 14 | 97.49 | 98.35 |
| 15 | 87.85 | 84.44 |
| 16 | 98.34 | 99.38 |
| AA(%) | 97.45 | 99.14+-0.24 |
| OA(%) | 95.81 | 98.47+-0.49 |

**Table 2**. Overall and average accuracy and class accuracies of one run for Salinas.

| Class | 3DCAE | TD |
|---|---|---|
| 1 | 95.21 | 98.33 |
| 2 | 96.06 | 99.98 |
| 3 | 91.32 | 85.18 |
| 4 | 98.28 | 95.63 |
| 5 | 95.55 | 99.42 |
| 6 | 95.30 | 99.60 |
| 7 | 95.14 | 92.26 |
| 8 | 91.38 | 93.27 |
| 9 | 99.96 | 93.38 |
| AA(%) | 95.36 | 95.95+-0.50 |
| OA(%) | 95.39 | 97.93+-0.23 |

**Table 3**. Overall and average accuracy and class accuracies of one run for PaviaU.

| Class | Ind on Sal | Sal on Ind |
|---|---|---|
| AA(%) | 63.26+-3.19 | 97.30+-0.11 |
| OA(%) | 71.77+-0.80 | 95.01+-0.10 |

**Table 4**. Overall and average accuracy and class accuracies of one run for Indian pines trained on Salinas and vice versa.

| Class | SCAE-Hyperion | TD |
|---|---|---|
| AA(%) | 96.56+-0.51 | 93.47+-0.21 |
| OA(%) | 95.84+-0.94 | 96.51+-0.22 |

**Table 5**. Overall and average accuracy and class accuracies of one run for Pavia U trained on Pavia compared with Refs

methods is higher than the 3DAE, in two classes its accuracy is significantly lower. This can be explained by the number of pixels belonging to these two classes campared to other ones. As in this method, one large tensor is formed for extracting the dictionary from all the classes simoltanously, smaller classes whose comprising portions can't be described well from elements of larger classes will suffer performance wise. This need not be the case for any small class as we see that class number 9 (Oats) also has fewer pixels but the proposed method has a high accuracy for that class indicating that it's spectral-spatial signature can be reasonably reconstructed from the larger classes. A future direction to remedy this problem could be constructing seperate tensors for different classes, which would then take this method from an unsupervised feature learning to a supervised one.

## 5. CONCLUSION AND DISCUSSION

We proposed a novel tensor decomposition method for unsupervised feature learning. We compared our method against a state of the art method for UFL on publicly available datasets, where it was shown that it achieved competetive accuracy. The drawback of the method is that if the dataset contains classes that have few datapoints but vary widely in terms of their spectral and spatial signatures, their class accuracy could be low while the overall and average accuracies remain competetive or even higher than the other method. One way to remedy this would be to construct seperate tensors for various classes, which would be a supervised feature learning approach.

## 6. REFERENCES


[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag*, vol. 1, no. 2, pp. $6_36, 2013$.

[2] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geosci. Remote Sens*, vol. 43, no. 3, pp. 466479, 2005.

[3] Y. Ren, L. Liao, S. J. Maybank, Y. Zhang, and X. Liu, "Hyperspectral image spectral-spatial feature extraction via tensor principal component analysis," *IEEE Geosci. Remote Sens. Lett*, vol. 14, no. 9, pp. 14311435, 2017.

[4] Q. Du, "Modified fishers linear discriminant analysis for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett*, vol. 4, no. 4, pp. 503507, 2007.

[5] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image

classification," *IEEE Trans. Geosci. Remote Sens*, vol. 48, no. 11, pp. 40994109, 2010.

[6] G. Chen and S.-E. Qian, "Dimensionality reduction of hyperspectral imagery using improved locally linear embedding," *Journal of Applied Remote Sensing*, vol. 1, no. 1, pp. 013509, 2007.

[7] Y. Li C. Tao, H. Pan and Z. Zou, "Unsupervised spectralspatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 24382442, 2015.

[8] Y. Geng Z. Zhang X. Li S. Mei, J. Ji and Q. Du, "Unsupervised spatialspectral feature learning by 3d convolutional autoencoder for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6808 – 6820, 2019.

[9] Amnon Shashua and Anat Levin, "Linear image coding for regression and classification using the tensor-rank principle," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, 2001, vol. 1, pp. I–I.

[10] Muhammad Ali and Hassan Foroosh, "Character recognition in natural scene images using rank-1 tensor decomposition.," *Proceedings of the IEEE International Conference on Image Processing*, pp. 2891–2895, 2016.

[11] Na Qi, Yunhui Shi, Xiaoyan Sun, and Baocai Yin, "Tensr: Multi-dimensional tensor sparse representation," *Computer Vision and Pattern recognition*, pp. 5916–5925, 2016.

[12] G. Sheng, W. Yang, T. Xu, and H. Sun, "High-resolution satellite scene classification using a sparse coding based multiple feature combination," *Journal of Remote Sensing*, vol. 33, no. 8, pp. 2395–2412, 2012.

[13] Morten Mørup, Lars Kai Hansen, and Sidse M Arnfred, "Algorithms for sparse nonnegative tucker decompositions," *Neural computation*, vol. 20, no. 8, pp. 2112–2131, 2008.

[14] N. Sidiropoulos, L. De Lathauwer, F. Xia, K. Huang, E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 65, no. 13, pp. 2108–2123, 2017.

[15] Alfred M Bruckstein, Michael Elad, and Michael Zibulevsky, "On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4813–4820, 2008.