16th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 26-28, 2022

Tensor-Based and Projection-Based Methods for Dimensionality Reduction of Hyperspectral Images

Laura-Bianca BILIUS[1,2], Stefan Gheorghe PENTIUC[1,2]
[1]Faculty of Electrical Engineering and Computer Science, Stefan cel Mare University of Suceava
[2]MintViz Lab | MANSiD Research Center, Stefan cel Mare University of Suceava
laura.bilius@usm.ro, pentiuc@eed.usv.ro

Abstract— **Inasmuch as the hyperspectral images are represented by large amounts of data it is necessary to adopt an appropriate method to reduce their size without affecting the quality of their processing results. This paper addresses the use of Tucker1 decomposition for tensor compression and dimensionality reduction, followed by a projection-based method, Principal Component Analysis (PCA). After dimensional reduction, some classifications were performed using the features extracted. Various supervised learning algorithms were used for which we calculated k-fold cross-validation loss. We made a comparison of these methods in terms of the classification results obtained. According to the results, at the same size of the transformed data, PCA features have led to lower accuracy than Tucker1 ones, and the original data.**

Keywords—Tucker1 decomposition; dimensionality reduction; supervised learning; hyperspectral images; PCA

I. INTRODUCTION

Hyperspectral images have proven great potential in various applications (*eg*., mineral detection, mapping, agriculture). However, the hyperspectral images may be represented by a large volume of data. For example, a hyperspectral image of 400x450 pixels and 102 spectral bands is represented by over 18 million real values. Their large size creates difficulties in handling and analyzing them. Reducing their size aims to remove redundant information that may affect the performance of the classification and extract the most valuable features. In this way, machine learning algorithms will learn efficiently and provide qualitative predictions. The concern of reducing the dimensionality of hyperspectral images is high, for this purpose being proposed various algorithms, such as, PARAFAC [1], Tucker [1], Block-Term [2], tensor-rank decomposition [3].

This paper describes an analysis of two general methods for reducing the dimensionality of data that will be applied to spectral image processing: a tensor decomposition method, Tucker1, and a projection-based method, Principal Component Analysis (PCA). The links between these methods will be highlighted, and by performing a comparison study, the benefits and disadvantages of reducing the dimensionality of hyperspectral images in the operation of various supervised algorithms will be analyzed.

An important advantage of Tucker decomposition is the presence of a resulted core tensor, which in the case of hyperspectral images can keep the size of spatial dimension and reduce only the spectral dimension offering an efficient representation of the data.

The contributions of this paper consist of a hyperspectral images dimensionality reduction analysis using Tucker1 decomposition and identification of the most appropriate supervised learning algorithms for these data. We also highlight the benefits of extracting valuable features using Tucker1, justifying them by experimental results.

II. RELATED WORK

A great diversity of techniques for hyperspectral data dimensionality reduction were proposed in the scientific literature, *e.g.*, Parafac [4] or BTD [2], and various supervised [5] or unsupervised learning algorithms [6] are introduced to increase the accuracy of the classification and reduce the computational cost. Tucker decomposition is found in various applications such as, chemical analysis [7], signal processing [8], computer vision [9], data mining [10], and so on. Tucker decomposition is also used for hyperspectral images analysis. For example, Wang *et al.* [11] introduced a method based on lapped transform and Tucker decomposition that proved to be an efficient method in signal processing, Ye *et al.* [12] proposed Three Dimensional Tensor Convolutional Neural Network (CNN) for spectral features and a semi-supervised model based on CNN for classification that proved to be an efficient method on real data.

Tucker1 Decomposition is often associated in the literature to PCA, so Wang et al. [11] affirm that Tucker Decomposition is a high-order PCA. According to Hadi-Fanaee *et al.* [13], Tucker1 decomposition is also named Multi-way PCA or MPCA (Multi-linear PCA) that is mostly used when the variance is important for one dimension. Benthem and Keenan in [14] studied Tucker1 decomposition for hyperspectral fluorescence data by proposing a method based on PARAFAC decomposition and a core tensor. They used Tucker decomposition to obtain a compressed tensor for a faster response on PARAFAC decomposition. According to Bro *et al.* [15], Tucker1 decomposition unfolds the tensor into a two-way matrix and then applies the Principal Component Analysis algorithm.

The aim of this paper is to study the efficiency of the hyperspectral image classification using Tucker1 decomposition features.

978-1-6654-8162-5/22/$31.00 ©2022 IEEE

III. METHODOLOGY

Tucker Decomposition is a method for analysis and compression. It decomposes a tensor into a core tensor and a set of factor matrices [16]. Thus, for a three-way tensor $X \in \mathbb{R}^{IxJxK}$, the decomposition is defined as:

$$X \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}\, a_p \circ b_q \circ c_r = [G;\, A,\, B,\, C] \quad (1)$$

where $A \in \mathbb{R}^{IxP}$, $B \in \mathbb{R}^{JxP}$ and, $C \in \mathbb{R}^{KxP}$ are the factor matrices, $G \in \mathbb{R}^{PxQxR}$ is the core tensor, and P, Q and R are the number of components along each dimension [1]. For a graphical representation see Fig. 1. The core tensor keeps the most relevant information from the original tensor [11].

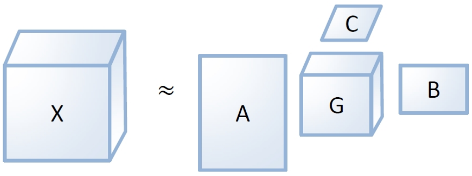

Fig. 1. Graphical representation of Tucker Decomposition for a three-way tensor [2].

An important observation concerns a related model, namely Tucker1 decomposition that sets any two of the factor matrices to be identity matrices, so $X \approx [G;\, A,\, \boldsymbol{I},\, \boldsymbol{I}] = A \times G$ (see Fig. 2).

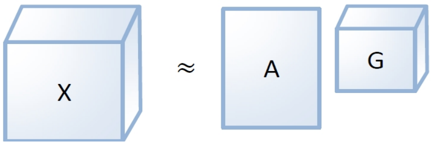

Fig. 2. Graphical representation of Tucker1 Decomposition [2].

Tucker1 or MPCA is defined as generalized PCA, the difference between them consists in the fact that PCA reshapes multidimensional data; meanwhile, MPCA operates directly on multidimensional data [17]. In other words, Tucker1 is defined as "*a more advanced way of unfolding which yields separate component matrices for each mode*" [18]. Also, Tucker1 uses multiple orthogonal transformations to construct multidimensional data where for each dimension there is an orthogonal transformation that is converted into multidimensional data with a lower dimension that captures more variance from the tensor than PCA [17]. More exactly, Tucker1 decomposition searches the principal components for each mode of the tensor data [14].

IV. RESULT

In this paper, we approached Tucker1 decomposition on hyperspectral images to keep the spatial information untouched. To this end, we only reduced the dimensionality along the spectral dimension. Whereas the spatial dimension doesn't change, we tested various Tucker1 components numbers for spectral dimension; see Table I and Table II for more details on the datasets dimensions, the explained variation, and execution time of Tucker1 Decompositions.

Regarding data processing, let be a hyperspectral image $H \in \mathbb{R}^{IxJxK}$. After Tucker1 decomposition is applied on H, a tensor $H' \in \mathbb{R}^{IxJxn}$ is obtained, where n represents the number of spectral components. The classification features are represented by the matricization of H' tensor, *ie.*, $M_T \in \mathbb{R}^{(I*J)xn}$.

Also, we computed the PCA features for which we used the same number of components; see Table III. For this approach, we unfolded the tensor and applied the algorithm using the same number of components as we used for Tucker1. The reason why we used the same number of components was to have input data of the same size for learning; see Table I.

Regarding PCA data processing, let be the tensor $H \in \mathbb{R}^{IxJxK}$ that is unfolded into a matrix of dimensions $M' \in \mathbb{R}^{(I*J)xK}$. After applying PCA, we obtained a matrix $M_P \in \mathbb{R}^{(I*J)xn}$; see Fig. 3. Concerning the original features that we used for classification, we unfolded the tensor $H \in \mathbb{R}^{IxJxK}$ into a matrix $M \in \mathbb{R}^{(I*J)xK}$.

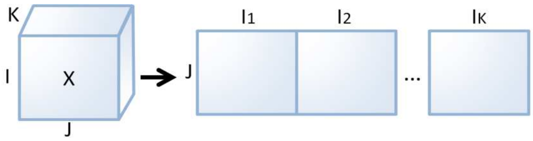

Fig. 3. A three-way tensor unfolding.

TABLE I. INFORMATION ABOUT THE DATASETS DIMENSIONS.

Data set	*Dimension*	Tucker1/PCA[a]				
		10	*20*	*30*	*40*	*50*
PaviaU	610x340x103	9.7%	19.4%	29.1%	38.8%	48.5%
Salinas	512x217x204	4.9%	9.8%	14.7%	19.6%	24.5%
PaviaC	400x450x102	9.8%	19.6%	29.4%	39.2%	49.0%

a. The percentage value of the data obtained after Tucker1 decomposition and PCA from the entire data set for a different number of components.

TABLE II. THE EXECUTION TIME AND EXPLAINED VARIATION (**E.V.**) OF THE TUCKER1 (**T^c**) AND PCA DECOMPOSITIONS (**P^c**).

Data set	Method		Number of components				
			10	*20*	*30*	*40*	*50*
PaviaU	**T^c**	E.V.	99.96	99.98	99.99	99.99	99.99
		Time [s]	7.54	8.77	8.96	9.02	10.88
	P^c	E.V.	99.81	99.93	99.96	99.98	99.99
		Time [s]	2.63	3.58	4.04	4.55	4.45
Salinas	**T^c**	E.V.	99.99	99.99	99.99	99.99	99.99
		Time [s]	5.95	6.14	7.22	7.42	9.95
	P^c	E.V.	99.96	99.98	99.99	99.99	99.99
		Time [s]	3.75	4.17	4.76	5.04	5.20
PaviaC	**T^c**	E.V.	99.94	99.98	99.99	99.99	99.99
		Time [s]	7.42	9.60	8.98	9.13	9.94
	P^c	E.V.	99.83	99.93	99.96	99.98	99.99
		Time [s]	2.25	2.37	3.70	4.10	5.23

To test the efficiency of the three methods, we applied various machine learning algorithms from the Scikit-learn library [19], and we extracted the ones with the highest cross-validation accuracy, specifically, Decision Tree (DT), Extra Tree (ET), Gaussian Naïve Bayes (GNB), k-Nearest Neighbors (kNN), Linear Discriminant Analysis (LDA), Passive Aggressive (PA), Ridge (R), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM) classifiers. In this direction, we computed the 10-fold cross-validation score for each classifier [19]; see Table III.

It is noticeable that the difference between the accuracy of the original data and the accuracy of Tucker1 decomposition features is very small compared to PCA (see Table III), *e.g.*, the accuracy value for Tucker1 using 20 components is equal to the accuracy of the original data. These accuracy values confirm that features extracted by Tucker1 decomposition are valuable for hyperspectral images classification.

TABLE III. 10-FOLD CROSS-VALIDATION OF THE ORIGINAL DATA AND THE FEATURES OBTAINED AFTER APPLYING TUCKER1 DECOMPOSITION AND PCA (PRINCIPAL COMPONENT ANALYSIS).

D-N[d]		*KNN*	*SVM*	*DT*	*GNB*	*LDA*	*PA*	*ET*	*R*	*SGD*
T-10	Acc/	90.4	86.5	88.6	86.1	86.3	73.6	85.5	75.3	87.1
	SD	3.6	3.0	3.5	4.8	4.1	8.8	3.3	1.9	3.1
P-10	Acc/	73.4	77.1	69.6	73.4	72.5	39.1	66.3	57.8	67.0
	SD	2.6	3.1	2.2	3.5	2.2	10.8	2.2	0.8	2.0
T-20	Acc/	91.1	87.0	89.3	87.6	88.6	77.5	82.0	80.9	89.1
	SD	3.6	3.1	3.5	4.5	4.9	7.4	3.5	2.7	3.4
P-20	Acc/	79.4	82.0	75.1	79.9	79.8	50.9	66.1	66.5	75.9
	SD	2.6	3.1	2.4	3.4	2.3	9.3	2.8	1.2	2.0
T-30	Acc/	91.1	87.0	89.1	87.6	89.1	76.8	81.3	82.8	89.1
	SD	3.6	3.1	3.4	4.3	4.8	7.2	3.2	2.8	3.3
P-30	Acc/	84.7	85.9	80.6	83.2	83.2	52.9	67.8	71.3	79.9
	SD	2.7	2.8	2.7	3.6	3.1	8.1	2.4	1.7	2.9
T-40	Acc/	91.1	87.0	89.0	87.6	89.2	77.7	79.0	83.3	89.2
	SD	3.6	3.1	3.3	4.3	4.9	6.9	3.4	2.8	3.4
P-40	Acc/	86.9	87.6	83.1	85.2	84.1	55.2	69.6	74.5	83.1
	SD	2.5	2.7	2.5	3.7	4.1	8.6	2.8	2.6	2.2
T-50	Acc/	91.1	87.0	88.9	87.6	89.2	77.9	76.1	83.4	89.1
	SD	3.6	3.1	3.3	4.3	4.9	6.8	3.2	2.9	3.4
P-50	Acc/	87.6	87.8	83.9	85.0	84.9	60.0	68.5	76.2	84.4
	SD	2.4	2.8	2.4	4.0	3.7	29.7	3.4	2.0	2.1
Org.[e]	Acc/	91.1	90.3	89.2	77.1	89.5	77.5	87.7	84.8	85.3
	SD	3.6	3.5	3.4	4.3	4.7	7.1	3.4	2.9	4.7

b. We noted as "D-N" the decomposition method (Tucker1 or PCA) and the corresponding components number, *e.g.*, T-10 is Tucker1 decomposition using 10 components.

c. We noted as "Org." the original hyperspectral dataset.

As we can see from Fig. 3, kNN and SVM obtained on average the greater values for 10-fold cross-validation. An important observation is that even if PCA decomposes quicker a big data set, Tucker1 offered greater accuracy values.

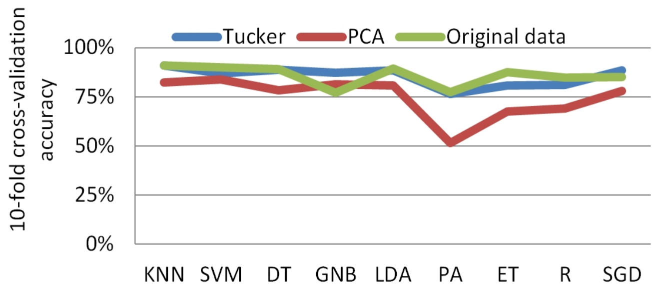

Fig. 4. Accuracy mean values for each classifier mentioned in Table III.

The obtained results can be justified by the fact that the PCA reshapes the data into vectors, suffering because this technique can fail on high-dimensional data [20], meanwhile, Tucker1 or MPCA operates directly on tensor data taking into account the correlations [21].

Fig. 4, 5 and, 6 illustrates the classification results for each method approached in this paper. The hyperspectral images used in this study can be found on the website [22]. These figures were generated for the highest 10-fold cross-validation accuracy values identified for each data set used in this study.

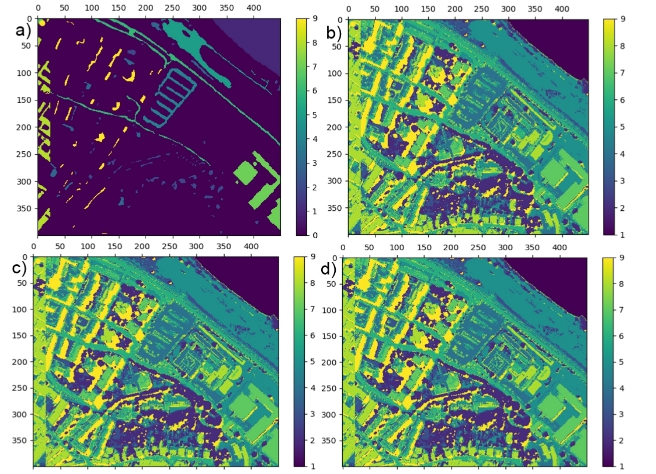

Fig. 5. (a) The Ground-Truth of the Pavia City dataset; graphical representation of the classification result using (b) 10 Tucker1 features, (c) 20 PCA features, and (d) the original data.

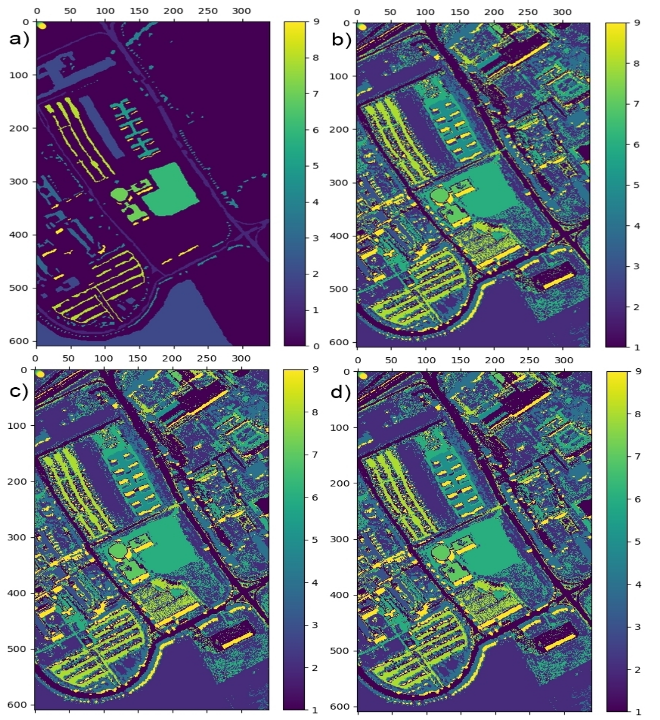

Fig. 6. (a) The Ground-Truth of the Pavia University dataset, graphical representation of the classification result using (b) 30 Tucker1 features, (c) 40 PCA features, and (d) the original data.

Also, the Fig. 4, 5 and, 6 illustrate the results obtained using various number of Tucker1 components, respectively PCA components, as well as the results obtained from the original dataset used as input data for the machine learning algorithms. The execution time of machine learning algorithms for classification of the datasets discussed in this study using 10, 20, 30, 40, and 50 Tucker1 and PCA components was on average 14.7, 46.1, 54.5, 64.6, respectively 83.3 seconds. With respect to the entire datasets, the execution time was on average 456.4 seconds.

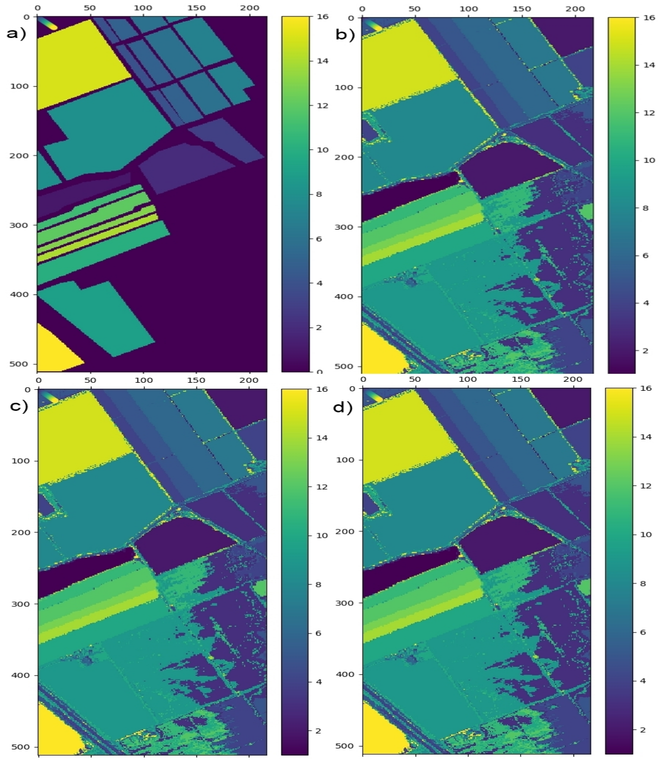

Fig. 7. (a) The Ground-Truth of the Salinas dataset; graphical representation of the classification result using (b) 30 Tucker1 features, (c) 40 PCA features, and (d) the original data.

V. Conclusion

In this paper, we applied Tucker1 decomposition to decompose only one mode of a tensor, specifically, only the spectral mode of a hyperspectral image is compressed, which means the feature extraction is applied on the spectral space. Even if in the scientific literature the PCA algorithm is often associated with Tucker1, according to the obtained results, PCA features gave lower accuracy values in a shorter time than Tucker1. Also, Tucker1 features delivered accuracy values very close to the original data. In conclusion, although the time required for Tucker1 decomposition is not very long compared to PCA, a compromise must be made between time and classification accuracy.

References

[1] T. G. Kolda, B. W. Bader. “Tensor Decompositions and Applications”. SIAM Review, Vol. 51, No. 3, 2009. https://doi.org/10.1137/07070111X

[2] Z. Zeng, T. Huang, Y. Chen and X. Zhao, "Nonlocal Block-Term Decomposition for Hyperspectral Image Mixed Noise Removal," in IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, (2021), https://doi.org/10.1109/JSTARS.2021.3079210

[3] J. Xue and Y. Zhao, "Rank-1 Tensor Decomposition for Hyperspectral Image Denoising with Nonlocal Low-Rank Regularization," 2017 International Conference on Machine Vision and Information Technology, 2017, https://doi.org/10.1109/CMVIT.2017.22

[4] L. -B. Bilius and S. G. Pentiuc, "Improving the Analysis of Hyperspectral Images Using Tensor Decomposition," 2020 International Conference on Development and Application Systems (DAS), 2020, pp. 173-176, doi: https://doi.org/10.1109/DAS49615.2020.9108935

[5] D. Anderson, A. Bapst, J. Coon, A. Pung, and M. Kudenov (2017) "Supervised non-negative tensor factorization for automatic hyperspectral feature extraction and target discrimination", Proc. SPIE 10198, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIII,; https://doi.org/10.1117/12.2267730

[6] Bilius LB, Pentiuc ŞG. “Efficient Unsupervised Classification of Hyperspectral Images Using Voronoi Diagrams and Strong Patterns”. Sensors (Basel), Vol. 20 (19), 2020. https://doi.org/10.3390/s20195684

[7] René Henrion, N-way principal component analysis theory, algorithms and applications, Chem.& Intellligent Lab Systems, Vol 25, 1, 1994, pp 1-23, ISSN 0169-7439, https://doi.org/10.1016/0169-7439(93)E0086-J

[8] Lieven De Lathauwer, Joos Vandewalle, Dimensionality reduction in higher-order signal processing and rank-(R1,R2,…,RN) reduction in multilinear algebra, Linear Algebra and its Applications, Volume 391, 2004, ISSN 0024-3795, https://doi.org/10.1016/j.laa.2004.01.016

[9] Vasilescu M.A.O., Terzopoulos D. (2002) Multilinear Analysis of Image Ensembles: TensorFaces. In: Heyden A., Sparr G., Nielsen M., Johansen P. (eds) Computer Vision — ECCV 2002. Lecture Notes in Computer Science, vol 2350. Springer. https://doi.org/10.1007/3-540-47969-4_30

[10] Berkant Savas, Lars Eldén, Handwritten digit classification using higher order singular value decomposition, Pattern Recognition, Volume 40(3), 2007, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2006.08.004

[11] Lei Wang, Jing Bai, Jiaji Wu, Gwanggil Jeon, Hyperspectral image compression based on lapped transform and Tucker decomposition, Signal Processing: Image Communication, Volume 36, 2015, Pages 63-69, ISSN 0923-5965, https://doi.org/10.1016/j.image.2015.06.002

[12] Fuyin Ye and Zhao Chen 2021, “Semisupervised Classification Based on Tensor Convolutional Neural Network for Hyperspectral Images”, Journal of Physics: Conference Series, 1880 012023, https://iopscience.iop.org/article/10.1088/1742-6596/1880/1/012023

[13] Hadi Fanaee-T, João Gama, "Tensor-based anomaly detection: An interdisciplinary survey", Knowledge-Based Systems, Volume 98, 2016, ISSN 0950-7051. https://doi.org/10.1016/j.knosys.2016.01.027

[14] M.H. Van Benthem, & M.R. Keenan (2008). Tucker1 model algorithms for fast solutions to large PARAFAC problems. Journal of Chemometrics, 22. https://doi.org/10.1002/CEM.1130

[15] Rasmus Bro. (2001). Multiway analysis in the food industry. Models, algorithms and applications. Ph.D. dissertation, University of Amsterdam, http://www.models.kvl.dk/sites/default/files/brothesis_0.pdf

[16] Stephan Rabanser, Shchur Oleksandr and Stephan Günnemann. “Introduction to Tensor Decompositions and their Applications in Machine Learning” (2017). https://arxiv.org/abs/1711.10781

[17] Modi Mansi, “Comparing Multilinear PCA and Kernel PCA with Linear PCA technique”, https://www.sjsu.edu/faculty/guangliang.chen/Math285S16/proposal-Mansi.pdf

[18] F. Feng, L. Wei, D. Qian D, Z. Bing, "Dimensionality Reduction of Hyperspectral Image with Graph-Based Discriminant Analysis Considering Spectral Similarity", Remote Sensing, 9, no. 4: 323, 2017. https://doi.org/10.3390/rs9040323

[19] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. https://scikit-learn.org/stable/

[20] H. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "MPCA: Multilinear Principal Component Analysis of Tensor Objects," in IEEE Transactions on Neural Networks, vol. 19, no. 1, pp. 18-39, Jan. 2008, doi: https://doi.org/10.1109/TNN.2007.901277

[21] Huang, J., Su, K., El-Den, J., Hu, T., & Li, J. (2014). “An MPCA/LDA Based Dimensionality Reduction Algorithm for Face Recognition”. Mathematical Problems in Engineering, 2014, 1-12. https://doi.org/10.1155/2014/393265

[22] Le Sun, “Datasets For Classification”, http://lesun.weebly.com/hyperspectral-data-set.html