Noise analysis using Tucker decomposition and PCA on spectral images

Análisis de ruido utilizando descomposición de Tucker y PCA en imágenes espectrales

PADILLA-ZEPEDA, Efraín†*, TORRES-ROMAN, Deni and MENDEZ-VAZQUEZ, Andrés

Instituto Politécnico Nacional, Center for Research and Advanced Studies, Telecommunications Group.

ID 1st Author: *Efraín, Padilla-Zepeda* / **ORC ID:** 0000-0002-9880-7157, **CVU CONACYT ID**: 924485

ID 1st Coauthor: *Deni, Torres-Román* / **ORC ID:** 0000-0002-9813-7712, **CVU CONACYT ID**: 20075

ID 2nd Coauthor: *Andrés, Méndez-Vázquez* / **CVU CONACYT ID**: 52042

DOI: 10.35429/EJB.2020.12.7.10.16 Received: January 15, 2020; Accepted: June 30, 2020

Abstract

Given the improvement of Remote Sensing (RS) sensors, it has been possible to increase spatial and spectral resolution on many of them. Nevertheless, the amount of data to represent and post-process has become highly prohibitive. Therefore, the need to be able to process such huge data sets, and one of the possible ways to deal with problems is the use of compression methods, however, data loss happen if the need of data size reduction is a must. RS spectral imagery contain high quantities of redundant information along the spectral domain, thus, making possible to use compression methods effectively as for example, tensor decomposition algorithms. In Tucker decomposition (TKD), an interesting and strange phenomenon happens when spatial domain is maintained and spectral domain is reduced, as a preprocessing step of a semantic segmentation task. Under these conditions, it is possible to observe an improvement on Pixel Accuracy (PA) metric when it is compared with the same uncompressed spectral image. Therefore, this work presents a study on how noise affects the Tucker Decomposition compared with Principal Component Analysis (PCA) and its impact in semantic segmentation.

Spectral Image, Tucker Decomposition, Noise

Resumen

Dada la mejora de los sensores de percepción remota, ha sido posible aumentar la resolución espacial y espectral en muchos de ellos. Sin embargo, la cantidad de datos para representar y postprocesar se ha vuelto altamente prohibitiva. Por lo tanto, existe la necesidad de poder procesar grandes conjuntos de datos, y una de las formas posibles de lidiar con este problema es el uso de métodos de compresión, sin embargo, la pérdida de datos es inevitable. Las imágenes espectrales contienen gran cantidad de información redundante a lo largo del dominio espectral, por lo tanto, es posible utilizar métodos de compresión de manera efectiva como, por ejemplo, algoritmos de descomposición tensorial. En la descomposición de Tucker, ocurre un fenómeno interesante y extraño cuando se mantiene el dominio espacial y se reduce el dominio espectral, como una etapa de preprocesamiento de segmentación semántica. Bajo estas condiciones, es posible observar una mejora en la precisión de píxeles cuando se compara con la misma imagen espectral sin comprimir. Por lo tanto, este trabajo presenta un estudio sobre cómo el ruido afecta la descomposición de Tucker en comparación con el análisis de componentes principales y su impacto en segmentación semántica.

Imagen spectral, Descomposición de Tucker, Ruido

Citation: PADILLA-ZEPEDA, Efraín, TORRES-ROMAN, Deni and MENDEZ-VAZQUEZ, Andrés. Noise analysis using Tucker decomposition and PCA on spectral images. ECORFAN Journal-Bolivia. 2020. 7-13:10-16.

* Correspondence to Author (Email: asanchezv@ipn.mx)

† Researcher contributing as first author

Introduction

The Greek word “spectral”, which relates to “colors”, combined with “image” figuratively mean “image of colors”. This concept is based on taking a portion of the electromagnetic spectrum and breaking it into pieces for the purpose of analytical computations (Borengasser *et al.*, 2007). Therefore, in this paper, we took the decision to represent the Spectral Images (SI) as tensors given their flexibility as data representation. Why to take such choice? On possible answer is that spectral images images contain abundant spatial and spectral information and is always corrupted by various noises, especially Gaussian Noise (Kong *et al.*, 2019), making necessary to look for good representations of such spatial data and noise. Not only that, there is a generalized problem in spectral imagery applications with noise, given that, the performace is highly dependent on the Signal to Noise Ratio (SNR) (Rasti *et al.*, 2018).

A strange phenomenon happen on spectral imagery when spatial domain is maintained and spectral domain is reduced through Tucker decomposition as a preprocessing step of a semantic segmentation task, seen in (López *et al.*, 2020), since we can observe an improvement on Pixel Accuracy (PA) metric compared with uncompressed spectral image.

In this work, we have done a noise analysis after two compression methods which are Principal Component Analysis (PCA) and Tucker Decomposition (TKD). In such methods, it has been possible to observe some bands of the resulting images after compression and compare it graphically. Also, compare the PA in Semantic Segmentation task through a Neural Network (Multi-Layer Perceptron) in both cases.

This work is organized as follows: Section 2 introduces basic concepts of tensors and notation. Section 3 presents some concepts of the noise present in spectral imagery. In section 4 the theory behind compression methods that are implemented in this work are defined. In section 5 the experiments and results are presented. Finally, in section 6 the results are discussed.

Tensor Algebra Notation and Spectral Image Representation

A tensor is a multidimensional array. The order of a tensor is the number of its dimensions, also called the ways or modes. Therefore and Nth-order is an array with N dimensions (Kolda & Bader, 2009).

The following notation is used in this paper:

x : a scalar.

$\mathbf{x}$: a 1st-order tensor or a vector.

$\mathbf{X}$: a 2nd-order tensor or a matrix.

$\boldsymbol{\mathcal{X}}$: a 3rd-order tensor.

$x_{i,j,\dots,n}$: the element $(i,j,\dots,n)$ of a tensor $\boldsymbol{\mathcal{X}}$.

$\mathbf{x}_i$: the ith column of a matrix **X.**

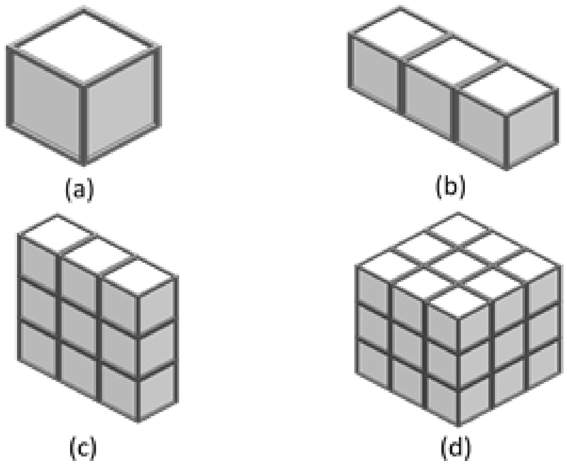

Figure 1 Graphic tensor examples. (a) $x \in \mathbb{R}$: a scalar, (b) $\mathbf{x} \in \mathbb{R}^{3}$: a 1st-order tensor, (c) $\mathbf{X} \in \mathbb{R}^{3\times3}$: a 2nd-order tensor, (d) $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{3\times3\times3}$: a 3rd-order tensor
Source: own work [SolidWorks].

Thus, the representation of a SI is a 3rd-order tensor:

$\boldsymbol{\mathcal{S}}$: Spectral image (1)

Where $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{h\times w\times b}$, h, w and b are the number of elements (pixels) in Mode-1 (column), Mode-2 (row) and Mode-3 (tube) fibers respectively

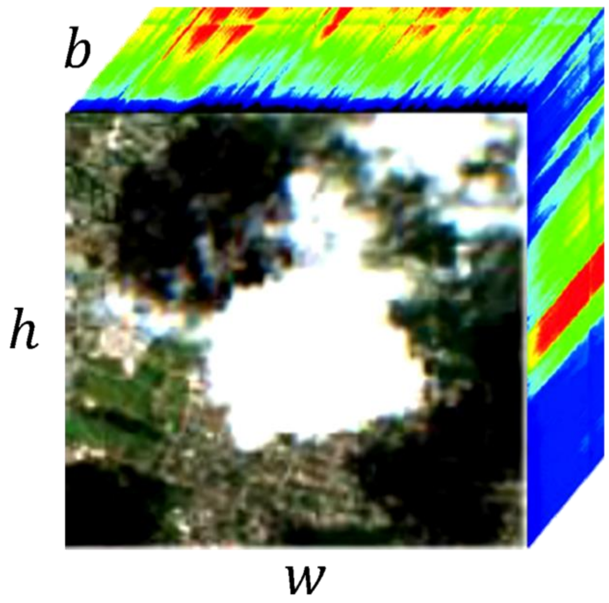

Figure 2 Spectral image example, where h, w and b are the height (rows), width (columns) and bands (tubes) respectively
Source: Sentinel-2 mission. ESA

Noise Assumptions

The presence of different noise sources in a SI makes its modelling and the denoising task very challenging, therefore, SI denoising approaches often consider either of the following types or a mixture of them as seen in (Rasti *et al*., 2018):

Noise Types

Signal Independent Noise

Thermal noise and quantization noise in HSI are modeled by signal independent Gaussian additive noise. Usually, noise is assumed to be uncorrelated spectrally having a diagonal noise covariance matrix, the model was introduced by (Landgrebe & Makaret, 1986).

The Gaussian assumption has been broadly used in hyperspectral analysis since it considerably simplifies the analysis and the noise variance estimation.

Sparse Noise

Impulse noises such as salt and pepper noise, missing pixels, missing lines and other outliers often exist in the acquired SI and are usually due to a malfunctioning of the sensor.

Pattern Noise

Spectral imaging systems may also induce artifacts in spectral images, usually referred to as pattern noise.

Additive Noise and Simulated Noise Generation

Generally, in the state of the art can be found many ways to get (Rasti *et al*., 2018):

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \tag{2}$$

Where $\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{N}} \in \mathbb{R}^{h \times w \times b}$ are the noisy SI, clean SI and the additive noise of the SI respectively.

There are many other noises (i.e. multiplicative noise), nevertheless, for this work we are going to consider only the additive noise for its simplicity and good results in noise modeling.

The noise can be generated by adding zero-mean Gaussian noise as seen in (Rasti *et al*., 2018)(Bioucas-Dias & Nascimento, 2008).

$$N = [n_{ij}] \tag{3}$$

Where $n_{ij} \sim N(0, \sigma_i^2)$ is normally distributed.
The variance of the noise σ_i^2 variates along the spectral axis according to:

$$\sigma_i^2 = \sigma^2 \frac{e^{-\frac{(i-p/2)^2}{2\eta^2}}}{\sum_{j=1}^{p} e^{-\frac{(j-p/2)^2}{2\eta^2}}} \tag{4}$$

Where the power of the noise is controlled by σ, and η behaves like the standard deviation of a Gaussian bell curve, p is the number of bands. In this work, we consider only the additive noise as expressed by equation (2), the noise is generated with variances given by equation (4) (Landgrebe & Makaret, 1986).

Compression Methods

Principal Component Analysis (PCA)

Large datasets are increasingly common and are often difficult to interpret. PCA is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does by creating new uncorrelated variables that successively maximize variance. Finding such new variable, the principal components, reduces to solving an eigenvalue/eigenvector problem. For further information see (Jollife & Cadima, 2016).

For a given standardized dataset $\mathbf{A} \in \mathbb{R}^{m\times n}$ we get the covariance matrix from

$$\mathbf{C} = \mathbf{A}^{T}\mathbf{A} \tag{5}$$

Where $\mathbf{C} \in \mathbb{R}^{n\times n}$.

The n eigenvectors obtained from $\mathbf{C}$ are the principal components of the dataset and its corresponding eigenvalues are related with its importance. Discarding some of the less significant components (the number of components discarded depends of the application) we can get a transformation matrix $\mathbf{X}$ with the k remaining components in its columns, the dimensions of $\mathbf{X}$ are n by k.

From the following equation we get the new compressed dataset $\mathbf{Y}$ from the linear transformation of the original standardized dataset $\mathbf{A}$ multiplied by the transformation matrix $\mathbf{X}$.

$$\mathbf{Y} = \mathbf{AX} \tag{6}$$

Where $\mathbf{Y} \in \mathbb{R}^{m\times k}$.

To apply PCA and compress along the spectral domain of a SI, we get the vectorized frontal slices (h by w) of $\boldsymbol{\mathcal{S}}$. These vectors will be the columns of $\mathbf{A}$, thus $\mathbf{A} \in \mathbb{R}^{hw\times b}$ is a matricized form of $\boldsymbol{\mathcal{S}}$.

Tucker Decomposition (TDK)

Is a form of higher-order PCA. It decomposes a tensor into a core tensor multiplied (or transformed) by a matrix along each mode. Thus, in the three-way case where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ we have

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \tag{7}$$

The tensor $\boldsymbol{\mathcal{G}}$ is called the core tensor, and $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are the factor matrices and can be thought of as the principal components in each mode. In order to compress only along the last mode (in depth of the cube), we exchange the factor matrices with identity matrices.

$$\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{G}} \times_1 \mathbf{I} \times_2 \mathbf{I} \times_3 \mathbf{C} \tag{8}$$

The core tensor keeps the first two dimensions (spatial) but depth dimension (spectral) is reduced. There are many algorithms to compute the TKD like High Order Singular Value Decomposition (HOSVD) or Alternating Least Squares (ALS), for further information see (Kolda & Bader, 2009).

Dataset and Experiments

Dataset Description

The dataset consist of 115 scenes, each scene contain a multi-spectral image with dimensions $128 \times 128 \times 9$, an RGB reference image with dimensions $128 \times 128 \times 3$ and a corresponding matrix of labels with dimensions 128×128 from the Sentinel-2 mission of the of the European Space Agency (ESA) on Central Europe. Each pixel is labeled with one of the following classes of interest: Soil, shadow, cloud, vegetation or water.

Experiments

Experiment 1 – Band-1 Visualization after PCA and TKD Compression

In this experiment we took the 50th scene (out of 115) of the study case dataset with dimensions ($128 \times 128 \times 9$). We added simulated noise to the spectral image and after this, perform PCA and the TKD compressing the spectral domain. In the scene we observe a small vegetation zone at the top, a city in the middle and a mass of water in the RGB reference image, also we visualize only the first band (out of 9) of the study scene represented in grayscale in Fig. 3.

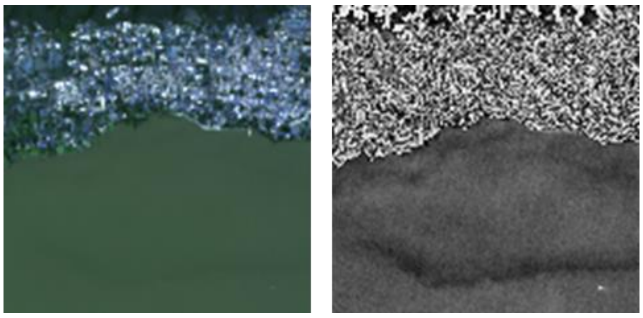

Figure 3 RGB reference image (left) and band-1 (right)
Source: ESA, own work [Python 3.7.6]

We have added simulated noise as seen in section 3.2 with parameters $\sigma = 13$ and $\eta = 72$ selected based on graphic empirical results.

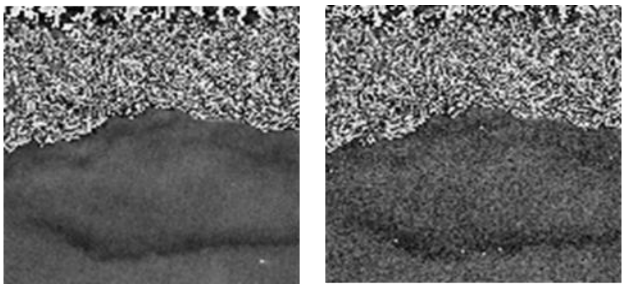

Figure 4 Band-1 of the study scene. Comparison between original (left) and added simulated noise (right).
Source: own work [Python 3.7.6]

After performed a spectral compression of the study scene with added simulated noise through PCA to 3 components and TDK to 3 tensorial bands preserving the spatial domain. An important remark is that we have done a reconstruction from the compressed study scene in both cases (from 3 components/tensorial bands back to the original 9 bands) only for visualization purposes in Fig. 5.

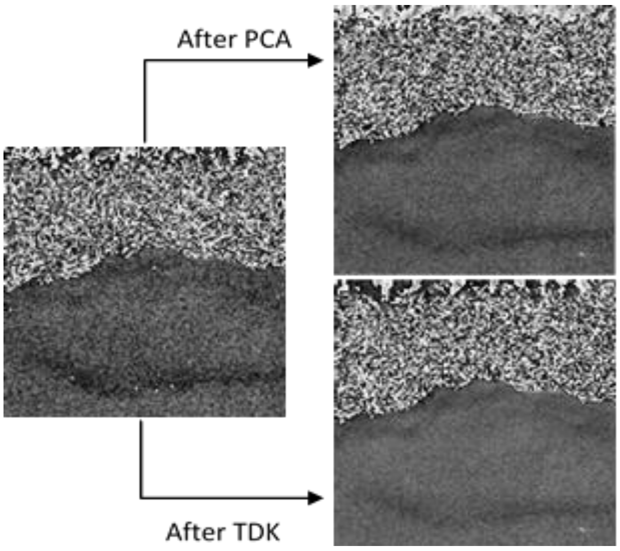

Figure 5 Band-1 of the study scene with added simulated noise (left), reconstruction of band-1 after PCA (top-right), reconstruction of band-1 after TKD (bottom-right)
Source: own work [Python 3.7.6]

Is easy to see a noise reduction over the mass of water zone in both cases, also we can observe a better performance of TKD over the vegetation zone.

Experiment 2 – Semantic Segmentation

In this section we reproduce the phenomenon of pixel accuracy improvement at some amount of the compression of multi-spectral images seen in (López et al., 2020) with the compression as a preprocessing step of a supervised pixel-wise semantic segmentation task.

Spectral image compression is performed in all possible cases from 9 original bands to (8, 7, …, 1) Principal Components (PC) in the case of PCA and compressed to (8, 7, …, 1) Tensorial Bands (TB) in the case of TKD. As a reminder, the images are only compressed in the spectral domain, spatial domain is kept for pixel-wise classification purposes.

The 115 compressed SI in tensor form was matricized into **X** such that each pixel is in each row and its corresponding values along the spectral axis are in the columns. The dimensions of the matrix **X** to be used in the classifications task are 115×128×128 by “b” bands, b depends of the amount of compression done by PCA or TKD, thus $\mathbf{X} \in \mathbb{R}^{\mathbf{1,884,160} \times b}$. The labels are vectorized into **y** in the same way to preserve the spatial relation. **X** and **y** was splitted picking random samples into .7-train/.3-test partition. RGB reference images are not used in this experiment.

For pixel-wise supervised classification of the 115 scenes we used a generic Multi-Layer Perceptron (MLP) neural network with the following parameters:

Parameter:	Value:
Number of hidden layers	2
Number of neurons in the hidden layers	100
Activation function	ReLu
Solver for weight optimization	Adam
Regularization term	L2 penalty
Learning rate	Adaptative
Iterations	10

Table 1 MLP parameters
Source: own work [Microsoft Excel]

We used this genereric MLP for PA analysis along the different amount of compression of the SI for simplicity and time reasons, probably with a more sophisticated classifier we will get higher pixel accuracies.

Experiment 2 Results

PC/TB:	PA-After PCA	PA-After TKD
8	0.64463386	0.720789813
7	0.642744424	0.718850841
6	0.646523296	0.737203847
5	0.645111526	0.72697117
4	0.642447209	0.727742513
3	0.629613904	0.720478445
2	0.604534647	0.723838032
1	0.52536055	0.687503538

Table 2 PA results along different amount of compression by PCA and TKD as a preprocessing step of semantic segmentation.
Source: own work [Python 3.7.6, Microsoft Excel]

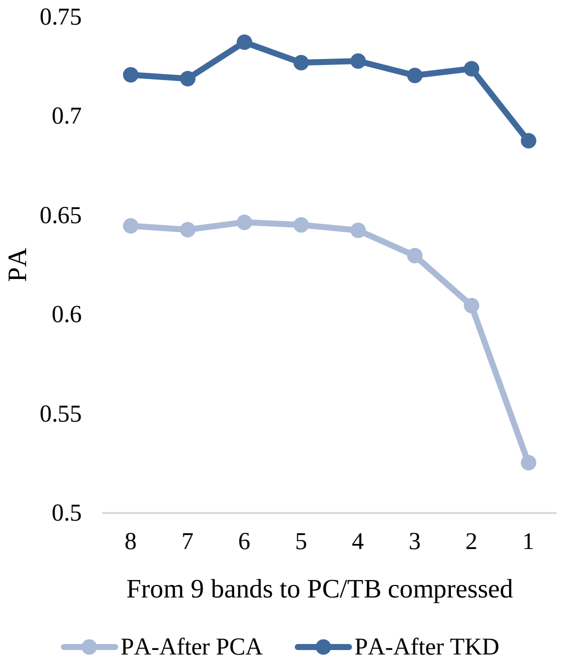

Figure 6 PA results in scatter plot along different amount of compression by PCA and TKD as a preprocessing step of semantic segmentation. A PA improve can be seen at 6 PC/TB; this increment is more notorious in TKD
Source: own work [Microsoft Excel]

Acknowledgment: to J. López for his help on the Tucker decomposition implementation and the provided dataset.

Conclusions

Generally, Tucker Decomposition can perform a better compression than Principal Component Analysis as can be seen graphically in the first experiment and numerically in the second one.

This can be seen at optimization for the Tucker Decomposition for spectral imagery compressing only on the spectral domain without matricization as in Principal Component Analysis. Thus, at the first experiment, we can see how both methods are able to reduce noise over masses of water. However, TKD preserves more information about vegetation zones. Furthermore, at the second experiment, the pixel accuracy obtained using TKD is superior than PCA in all the possible cases. For example, the PA after a TKD of six tensorial bands is higher than at seven and eighth which is related with the phenomenon seen at (López et al., 2020). Also, PCA shows this phenomenon, but with less significative results. Finally, both compression methods show poor performance with only one PC/TB respectively as expected. Therefore, future work is necessary to explain mathematically this phenomenon and its relationship with noise. This will allow to improve the classifier performance with modified state-of-the-art models for spectral image semantic segmentation with hardware accelerated implementations on TKD variants.

References

Bioucas-Dias, J. M., & Nascimento, J. M. P. (2008). Hyperspectral Subspace Identification. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, *46*(8). https://doi.org/10.1109/TGRS.2008.918089

Borengasser, M., Hungate, W., & Watkins, R. (2007). *Hyperspectral remote sensing: principles and applications*.

Jollife, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* (Vol. 374, Issue 2065). Royal Society of London. https://doi.org/10.1098/rsta.2015.0202

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. In *SIAM Review* (Vol. 51, Issue 3, pp. 455–500). https://doi.org/10.1137/07070111X

Kong, X., Zhao, Y., Xue, J., & Chan, J. C.-W. (2019). Hyperspectral Image Denoising Using Global Weighted Tensor Norm Minimum and Nonlocal Low-Rank Approximation. *Remote Sensing*, *11*(19), 2281. https://doi.org/10.3390/rs11192281

Landgrebe, D. A., & Makaret, E. (1986). Noise in Remote-Sensing Systems: The Effect on Classification Error. *IEEE Transactions on Geoscience and Remote Sensing*, *GE-24*(2), 294–300. https://doi.org/10.1109/TGRS.1986.289648

López, J., Torres, D., Santos, S., & Atzberger, C. (2020). Spectral Imagery Tensor Decomposition for Semantic Segmentation of Remote Sensing Data through Fully Convolutional Networks. *Remote Sensing*, *12*(3), 517. https://doi.org/10.3390/rs12030517

Rasti, B., Scheunders, P., Ghamisi, P., Licciardi, G., & Chanussot, J. (2018). Noise Reduction in Hyperspectral Imagery: Overview and Application. *Remote Sensing*, *10*(3), 482. https://doi.org/10.3390/rs10030482
.