



Spectral Classification of a Set of Hyperspectral Images using the Convolutional Neural Network, in a Single Training

Abdelali Zbakh, Zoubida Alaoui Mdaghri, Abdelillah Benyoussef, Abdellah El Kenz, Mourad El Yadari

► To cite this version:

Abdelali Zbakh, Zoubida Alaoui Mdaghri, Abdelillah Benyoussef, Abdellah El Kenz, Mourad El Yadari. Spectral Classification of a Set of Hyperspectral Images using the Convolutional Neural Network, in a Single Training. International journal of advanced computer science and applications (IJACSA), 2019, 10. hal-02172017

HAL Id: hal-02172017

<https://hal.science/hal-02172017>

Submitted on 3 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Spectral Classification of a Set of Hyperspectral Images using the Convolutional Neural Network, in a Single Training

Abdelali Zbakh¹, Zoubida Alaoui Mdaghri², Abdelillah Benyoussef³, Abdellah El Kenz⁴, Mourad El Yadari⁵

Lab: LaMCSsI, University Mohammed V, Faculty of Sciences Rabat, Morocco^{1, 2, 3, 4}

Moulay Ismail University of Meknes, Morocco⁵

Abstract—Hyperspectral imagery has seen a great evolution in recent years. Consequently, several fields (medical, agriculture, geosciences) need to make the automatic classification of these hyperspectral images with a high rate and in an acceptable time. The state-of-the-art presents several classification algorithms based on the Convolutional Neural Network (CNN) and each algorithm is training on a part of an image and then performs the prediction on the rest. This article proposes a new Fast Spectral classification algorithm based on CNN, and which allows to build a composite image from multiple hyperspectral images, then trains the model only once on the composite image. After training, the model can predict each image separately. To test the validity of the proposed algorithm, two free hyperspectral images are taken, and the training time obtained by the proposed model on the composite image is better than the time obtained from the model of the state-of-the-art.

Keywords—Classification; spectral; Convolutional Neural Network (CNN); deep learning; hyperspectral data; neural network

I. INTRODUCTION

Remote sensing makes it possible to identify objects remotely, without physical or chemical contact. Among the most used applications are hyperspectral images (HSI) captured by satellites or aircrafts. The captured HSI are then used in different domains and for different purposes: Geology (Detection of oil, water, ...), Agricultural (presence of diseases in plants and knowledge of plant species in a particular area).

A classic RGB image is an image represented by three layers or bands: Red, Green, and Blue while the hyperspectral image is represented by hundreds of bands. A hyperspectral image is represented by a data cube of two spatial dimensions (X and Y) and a spectral dimension Z. Each pixel corresponds to a spectrum of wavelength, generally corresponding to the visible and near-infrared domains (400 to 2500nm).

Among the methods that allow users to make the data of the HSI usable, and to extract the maximum of useful information, there is the classification.

Classification is an operation that divides a set of individuals into several classes, and each class groups together individuals who share the same similarity. There are two families of classification algorithms: Unsupervised classification (USVC) and Supervised classification (SVC).

In the USVC, we have unclassified elements and unknown classes, and we try to group the elements that have a certain similarity between them to construct a set of classes.

In the SVC, the classes are known in advance and we have examples on each class and we try to assign new elements to these classes. Examples of SVC algorithms: Naïve Bayes [1], Support Vector Machine (SVM), Deep learning [2] (Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM),)

In recent years, the domain of SVC and especially deep learning has undergone a great evolution. In addition, the classification of hyperspectral images by supervised algorithms [3], gave accuracy better than USVC algorithms.

This paper study a deep learning classification algorithm called convolutional neural network (CNN) to classify the content of a set of hyperspectral images using a single training. The classification object is to group in each class the pixels that have a certain similarity (common properties): water, vegetation, sand.

The rest of the paper is organized as follows: in Section 2, we will explain the classification by the convolutional neural network (CNN). In Section 3, we will describe our classification proposal. Section 4 contains the experimentation of the proposed model on several data sets. And we end in Section 5 with a conclusion and perspectives.

II. CONVOLUTIONAL NEURAL NETWORK

This section describes the CNN [4], the most popular supervised deep learning network, and who has shown its power of extracting features in computer vision applications.

A. CNN Standard Opérations

The convolutional neural network (CNN or ConvNet) is a particular and important type of neural network feed-forward (information spreads from layer to layer, without turning back possible). It is inspired by the biological processes that occur in the visual cortex in the brain of living beings. The CNN models are built on the same model as the multilayer perceptrons of which we find: an input layer, several intermediate hidden layers (depending on the depth of the model) and an output layer. CNN is used to solve several computer vision problems in artificial intelligence, for

example: Self driving cars, video processing, and image classification. The basic operations in a standard CNN network are:

- **Convolution operation:** The convolution operation is the basic operation in building a CNN network. It allows to slide, step by step, a window named kernel on the whole image, and for each step, we multiply the pixels of the kernel by the pixels of the region on which it slides. Then we take the sum of the result (see Fig. 1).
- **Max-Pooling operation:** The Max-pooling operation allows to slide, step by step, a window, generously of size 2x2, on the whole image, and takes at each step, the maximum value of the window. It is an optional operation in the design of the network, and in general, in classical CNN's architectures, it is put after each convolution operation, and which aims to reduce the number of samples or neurons. If the size of the window is large, we risk losing the information of the image (see Fig. 2).
- **Activation Functions:** These are correction functions, which play an important role in deep learning algorithms. The activation function takes in input an x value and returns the output $f(x)$. Activation functions are usually used after each convolution operation. The famous activations functions are: Identity ($f(x)=x$), Binary Step ($f(x)=0$ if $x<0$; else $f(x)=1$), Logistic or sigmoid ($f(x)=\frac{1}{1+e^{-x}}$), Tanh ($f(x)=\tanh(x)$) and Rectified Linear Unit (ReLU) ($f(x)=0$ if $x<0$; else $f(x)=x$).
- **Dropout:** The dropout operation, allows to randomly disable outputs of some neurons with a definite probability (0.5 for example). And this to simulate the real functionality of neurons, which can in an iteration of the learning phase, be inactive. Stalling speeds up learning
- **Fully connected (FC):** After several Convolution and Max-Pooling operations, come these operations to connect all the neurons of the previous layer (whatever the type), with the neurons of the next layer. It is not necessary to have a number of FCs, but there are often two consecutive layers as final layers in the network.

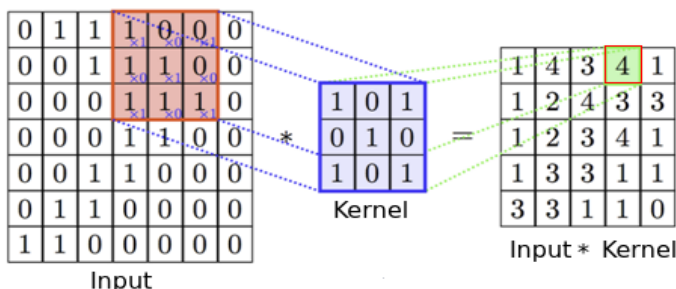


Fig. 1. Description of the Convolution Operation.

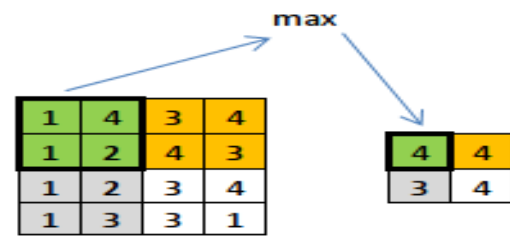


Fig. 2. Description of the Max-Pooling Operation.

B. Related Work

The first CNN classification model named LeNet-5 [5] was proposed by LeCun et al in 1998 to classify hand-written numbers. The model is composed of 7 layers (without counting the input layer). Other models of classification are appeared and which contains variety of number of layers: AlexNet [6] in 2012 with 9 layers, ZFNet [6] in 2013 with 8 layers, GoogleNet [7] in 2014 with 22 layers, VGGNet [6] in 2014 with 19 layers and ResNet [6] in 2015 with up to 269 layers.

In the classification of HSI using CNN, we find: spectral classification models [11, 9], 2D spatial classification models [9], 3D spatial classification models [10, 9] and Hybrid models [9].

Most of these classification models deals with Overall Accuracy (OA) performance and these OA are almost equal. Some work has focused on the speed of learning, which is a crucial criterion for choosing a model when working on two models of close accuracy. Among these work, we can mention:

- In [11], the authors proposed a CNN classification algorithm based on the spectral characteristics of HSI, and which contains 5 layers. The results (Accuracy, Training Time and Testing Time) are compared with LeNet-5, DNN and RBF-SVM.
- In [12], the authors propose a model of classification CNN, which is based on two channels: the first channel 1D to extract the spectral characteristics and the second channel 2D to extract the spatial characteristics. The results of the two canals are combined by the Softmax classifier. The training time of the model is compared with other model of the state-of-the-art (SSDCNN [13], SSDL [14]).
- In [3], the authors propose a 3D classification model of 5 layers, which uses at the same time the spectral and spatial characteristics of the image. The model is implemented using graphics processing units (GPUs) [9]. The results (Accuracy and training time) are compared with the classic MLP model and a CNN model of the state-of-the-art.

III. ARCHITECTURE OF THE PROPOSED MODEL

The classification algorithms of the hyperspectral images of the state-of-the-art [12-15-16], function according to the following principle:

Objective: classify the pixels of a hyperspectral image X, according to a certain number of class denoted C.

- Step 1: Divide the image X into two groups of data: X_train, to train the model and X_test to validate the model. Then create a classification model based on the parameters of image X (number of rows, number of columns and the number of spectral bands).
- Step 2: Train the created model on the X_train data, and record the time taken in this step denoted t_train.
- Step 3: Test the validity of the created model on X_test data.
- Step 4: Make the classification of the whole image X with the created model, and record the time spent in this step denoted t_pred. We note that training time is much greater than the prediction time.
- Step 5: Visualize the result.

Although this classification principle is used in almost all CNN based HSI classification algorithms, it has several flaws: among which, if we want to classify two new images Y and Z, we must repeat the same steps from 1 to 5 for the image Y and also for the image Z.

This classical method of classification takes a lot of time [10, 15], caused by the repetition of the training step for each image, and especially when working with a large number of images.

In this paper, we will propose an algorithm for spectral classifying an image hyperspectral composed of several HSI, based on the CNN, and using a single training.

The proposed classification algorithm proceeds as follows:

Objective: Classify the pixels of a hyperspectral image X1 according to a certain number of classes denoted C1.

- Step 1: Take k hyperspectral images of different sizes: X1 (H1, W1, N1, C1), X2 (H2, W2, N2, C2),...,Xk (Hk, Wk, Nk, Ck), with Hi, Wi, Ni, Ci represents the height, the width, the number of spectral bands, and the number of classes for the image i (respectively).
- Step 2: Choose the minimum number of bands between the k images: $N = \min (N1, N2, \dots, Nk)$.
- Step 3: Apply the dimensionality reduction algorithm PCA, on each image i that has a number of bands $Ni > N$
- Step 4: Vertically, concatenate the images obtained, to have a single image X of the following characteristics:
 - the number of pixels: $m = \sum_{i=1}^k (Hi \cdot Wi)$
 - the number of bands : $N = \min (N1, N2, \dots, Nk)$
 - the number of classes: $C = \sum_{i=1}^k (Ci)$
- Step 5: Divide the image X into two groups of data: X_train for the training and X_test for validating the prediction. Then create a spectral classification model based on the parameters of image X.

- Step 6: Train the created model on the X_train data, and we note the time taken in this step denoted t_train.
- Step 7: Test the validity of the created model on on X_test data.
- Step 8: Now, we can use the trained model to make the prediction on each image Xi: X1, X2,..., Xk separately, and we note the time in this step, denoted t_pred.
- Step 9: Visualize the classified image Xi.

A. Dimensionality reduction with PCA

The first step of the proposed algorithm is to take images of different sizes, each image Xi of size (Hi, Wi, Ni), will be converted towards the matrix format noted Mi, of size (number of lines Li = Hi x Wi and number of columns Ni). Each column j ($1 \leq j \leq Ni$) of the matrix Mi contains the pixels of the image Xi for the wavelength j and each line k ($1 \leq k \leq Li$) of the image Mi represents the values of a pixel k of Xi for all the wavelengths (see Fig. 3).

In step 2 and 3, we calculate the minimum of bands between the hyperspectral images that we will use: $N = \min (N1, N2, \dots, Nk)$, with k the number of images. The PCA reduction algorithm [17] is then applied to each image Mi, and the reduced images are concatenated to obtain the image M. For example, in Fig. 4, there is an illustration of the algorithm on k = 2 images: Pavia University and Salinas.

In the next part, we will propose a CNN spectral classification algorithm, inspired by paper [11] and which will be used for the classification of separate images (like the state-of-the-art) and also to test the proposed algorithm.

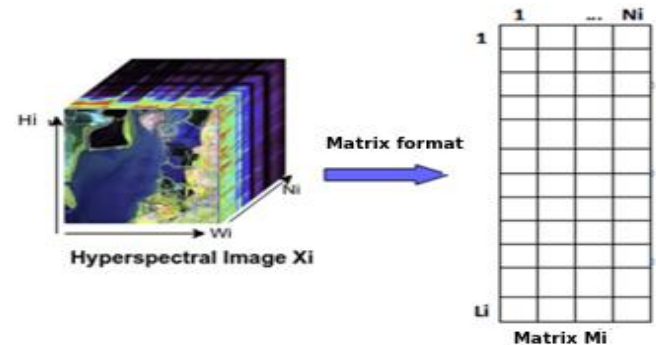


Fig. 3. Matrix Representation of the Hyperspectral Image Xi.

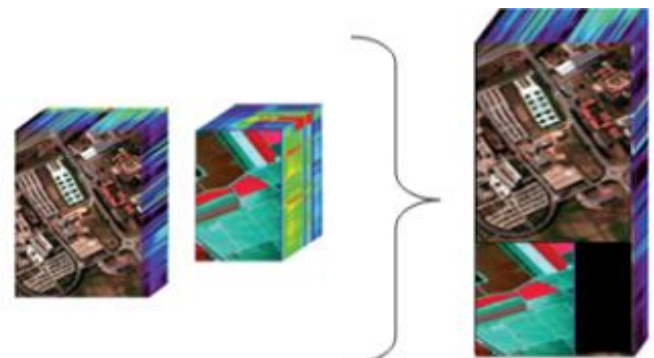


Fig. 4. Reducing the Dimensionality of Images and Concatenation to Obtain a Single Image M.

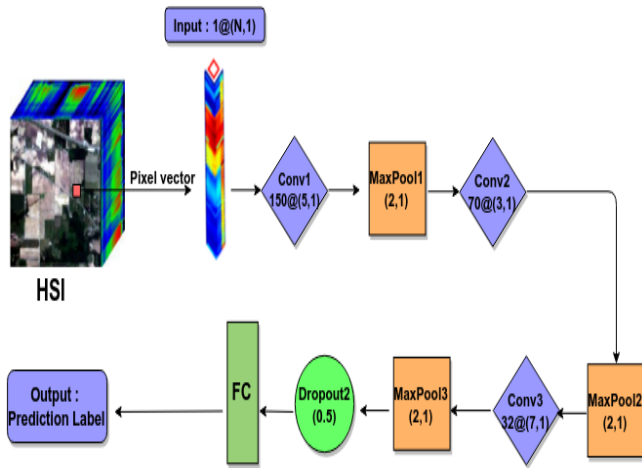


Fig. 5. Architecture of the Proposed CNN Classification Model.

B. Classification with the Spectral CNN

To classify the pixels of an HSI, we will propose a model composed of 10 layers: Input layer, 3 convolution layers, 3 Max-Pooling layers, a Dropout layer, a Fully Connected Layer and Output layer, with the following configuration (see Fig. 5).

The model takes as input a pixel vector of size N (number of bands), we apply on the pixel vector, various operations: convolution, Max-Pooling, Dropout and Fully Connected Layer according to the following parameters (Table I).

TABLE I. PARAMETERS OF THE PROPOSED MODEL

	Conv1	Conv2	Conv3
Number of filters	150	70	32
Kernel size	5 x 1	3 x 1	7 x 1
	Max-Pool 1	Max-Pool 2	Max-Pool 3
Kernel size	2 x 1	2 x 1	2 x 1

IV. EXPERIMENTAL AND COMPUTATIONAL DETAILS

A. DataSets

To classify the hyperspectral images using the proposed algorithm, we used two free datasets: Pavia University, and Salinas. For the two datasets, we took 70% pixels for training the model and 30% to test the classification model.

Pavia data: There are two types of Pavia data: Pavia Center and Pavia University. In this experiment, we use the corrected Pavia University [8], which depicts the scene of Pavia, northern Italy captured by the ROSIS (Reflective Optics System Imaging Spectrometer) sensor in 2001. The scene has a spatial dimension of 610 x 340 pixels with 103 bands of spectral reflectance in the wave range 0.43 to 0.86 μm . The scene contains 9 classes.

Salinas scene: The second dataset [8], is captured by the AVIRIS sensor on the Salinas-California valley, we find in this scene 512 x 217 pixels with 224 bands and which contains 16 classes. In this scene, 20 bands were deleted: (108-112; 154-167; 224) which represent water absorption bands.

B. Details and Results

First, we will begin by applying the proposed classification model on the two images and we note for each image, the accuracy (OA) and the time done in the training phase. Then we will build a single image from the 2 datasets as described in Section 3. And we will apply the classification model on this composite image, and note the accuracy and the training time.

Experiments are performed on a computer equipped with an Intel® Core™ processor i7-2820QM CPU @ 2.30GHz x 8, 16 GB Ram. The classification model is implemented in Python language using the deep learning library named: Keras. The following table (Table II) contains the values of the experiment:

From the paper [11], we took the accuracy (OA) and the training time of the model on the two images: PaviaU and Salinas. Then, two values were calculated: the average accuracy (OA_avg), and the total training time (T) of the model on the two images:

- $T = \text{Training time(PaviaU)} + \text{Training time (Salinas)}$
- $OA_avg = (OA(\text{PaviaU}) + OA(\text{Salinas})) / 2$

To test the efficiency and the speed of the proposed model, we made two experimentations: First, we trained the proposed model until obtaining the accuracy of the paper [11], and we noted the training time performed on each image: PaviaU and Salinas. The total training time obtained on the two images of the proposed model (695.82 s) is much smaller than the total time taken for training the model of paper [11] (3600 s). Secondly, we trained our model until we obtained the training time of paper [11], and we noted the accuracy on each image: PaviaU and Salinas. The average accuracy of the proposed model (94.71%) is better than the average accuracy of the paper [11] (92.58%).

We note that the proposed algorithm, in comparison with the algorithm of paper [11], gives a better accuracy on the two images, in a less time of training. The following graph (see Fig. 6), gives the evolution of the accuracy, as a function of training time for the proposed algorithm.

Since, the proposed model is competitive with the state-of-the-art classification model; we will use it to validate our approach to classify one image composed of several images, using a single training. Table III contains the accuracy value, the training time of the proposed algorithm on a single composite image and the OA_avg, the total of training time when the model was applied to the separate images

TABLE II. EXPERIMENTATION OF THE PROPOSED MODEL ON SEVERAL IMAGES

	Pavia U		Salinas		OA_avg (%)	T (s)
	OA (%)	Time (s)	OA (%)	Time (s)		
[11]	92.56	420	92.60	3180	92.58	3600
Proposed model (1)	92.59	117.34	92.8	578.48	92.69	695.82
Proposed model (2)	94.2	408.3	95.21	3176.07	94.71	3584.37

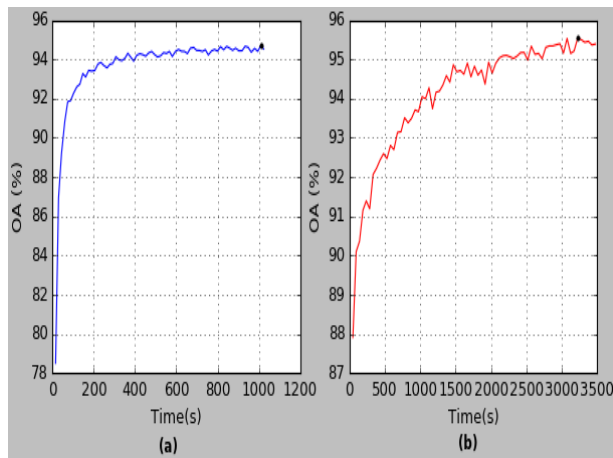


Fig. 6. The Accuracy According to the Training Time for Pavia University (a) and Salinas (b).

According to Table III, we note that the application of the proposed classification model on a single image composed of several HSI, gives a better accuracy value than applying the model on separate images and in a much less training time (see Fig. 7). The visual results of prediction are shown in Fig. 8.

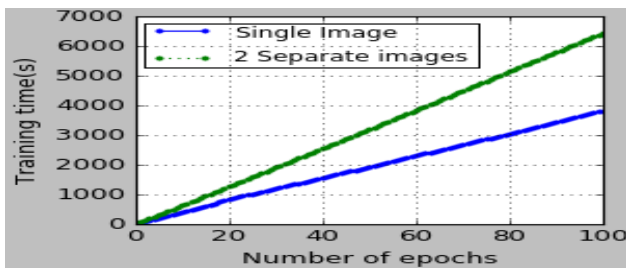


Fig. 7. The Variation of Training Time of the Proposed Model, on the Separate Images and on a Single Composed Image According to the Number of Epochs.

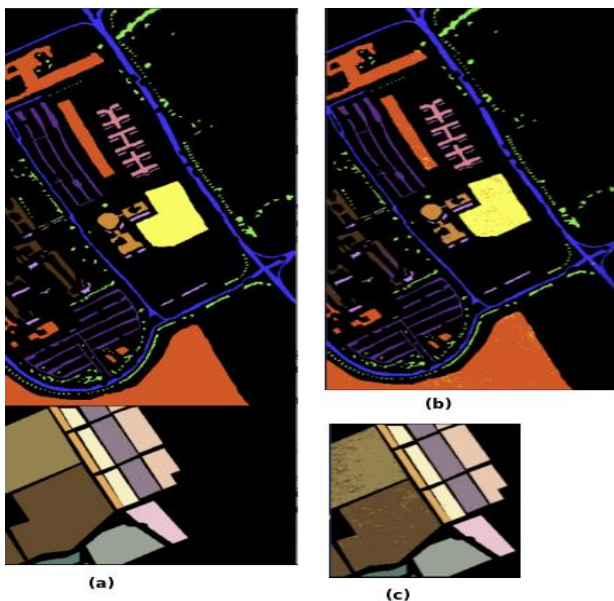


Fig. 8. RGB Composition Maps of Groundtruth for the Composite Image (a), Classification Results from the Proposed Model for : the Pavia U (b), Salinas (c).

TABLE III. EXPERIMENTATION OF THE PROPOSED MODEL ON A SINGLE IMAGE, COMPOSED OF 2 HSI

	Proposed model on : Two separate HSI		Proposed model on: A single image, composed of 2 HSI	
	OA_avg	T	OA	Training time
Test: 1	92.69 %	695.82 s	92.76 %	573.8 s
Test: 2	94.71 %	3584.37s	94.75 %	2869.41 s

V. CONCLUSIONS AND FUTURE WORK

In this paper, a new Fast classification model of a hyperspectral image composed of several HSI, using a single training has been proposed. The results of the comparison of the proposed algorithm with a state-of-the-art model [11] and even with the application of this algorithm on several HSI using multiple training, shows the speed and at the same time the performance of the proposed algorithm.

In the next work, we will increase the number of HSI that compose the image to be classified, and then we will create a CNN 3D spatial classification model that will be implemented in a distributed parallel environment.

REFERENCES

- [1] Lutz M, Biernat E. "Data science: fondamentaux et études de cas: Machine learning avec Python et R". Editions Eyrolles; Oct 2015.
- [2] Razzak, M. I., Naz, S., & Zaib, A.. Deep learning for medical image processing: Overview, challenges and the future. In Classification in BioApps (pp. 323-350). Springer, Cham.2018.
- [3] Paoletti, M. E., Haut, J. M., Plaza, J., & Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. ISPRS journal of photogrammetry and remote sensing, 145, 120-147. 2018.
- [4] Zhong, Z., Li, J., Ma, L., Jiang, H., & Zhao, H. Deep residual networks for hyperspectral image classification. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 1824-1827). IEEE. 2017.
- [5] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. 1998.
- [6] Luo, L., Liu, M., Nelson, J., Ceze, L., Phanishayee, A., & Krishnamurthy, A. Motivating in-network aggregation for distributed deep neural network training. In Workshop on Approximate Computing Across the Stack. 2017.
- [7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9). 2015.
- [8] [Online]. Available: www.ehu.es/ccwintco/uploads/e/e3/Pavia.mat; www.ehu.es/ccwintco/uploads/f/f1/Salinas.mat [Accessed: 01-May-2019].
- [9] Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing, 54(10), 6232-6251. 2016.
- [10] Mei, S., Yuan, X., Ji, J., Zhang, Y., Wan, S., & Du, Q. Hyperspectral image spatial super-resolution via 3D full convolutional neural network. Remote Sensing, 9(11), 1139. 2017.
- [11] Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 2015.
- [12] Zhang, H., Li, Y., Zhang, Y., & Shen, Q. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. Remote sensing letters, 8(5), 438-447. 2017.

- [13] Yue, J., Zhao, W., Mao, S., & Liu, H. Spectral-spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sensing Letters*, 6(6), 468-477. 2015.
- [14] Yue, J., Mao, S., & Li, M. A deep learning framework for hyperspectral image classification using spatial pyramid pooling. *Remote Sensing Letters*, 7(9), 875-884. 2016.
- [15] Lee, H., & Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10), 4843-4855. 2017.
- [16] Yang, J., Zhao, Y. Q., & Chan, J. C. W. Learning and transferring deep joint spectral-spatial features for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8), 4729-4742. 2017.
- [17] Zbakh, A., Mdaghri, Z. A., El Yadari, M., Benyoussef, A., & El Kenz, A. Proposition of a Parallel and Distributed Algorithm for the Dimensionality Reduction with Apache Spark. In *Proceedings of the Mediterranean Symposium on Smart City Applications* (pp. 490-501). Springer, Cham. 2017.