A Comparative Analysis of Deep Learning Techniques for Sub-tropical Crop Types Recognition from Multitemporal Optical/SAR Image Sequences

Jose Bermudez Castro.*, Raul Queiroz Feitosa*†, Laura Cue La Rosa*, Pedro Achanccaray Diaz* and Ieda Sanches‡
*Pontifical Catholic University of Rio de Janeiro, Brazil
Email: {bermudez, raul, lauracue, pmad9589}@ele.puc-rio.br
†Rio de Janeiro State University, Brazil
‡National Institute for Space Research
Email:ieda.sanches@inpe.br

***Abstract*—Remote Sensing (RS) data have been increasingly applied to assess agricultural yield, production and crop condition. In tropical areas, crop dynamics are complex due to multiple agricultural practices such as irrigation, non-tillage, crop rotation and multiple harvest per year. Spatial and temporal information can improve the performance in land-cover and crop type classification tasks. In this context Deep Learning (DL) have emerged as a powerful state-of-the-art technique in the RS community. This work presents a comparative analysis of traditional and DL (supervised and unsupervised) approaches for crop classification on sequences of multitemporal optical and SAR images. Three different approaches are compared: the image stacking approach, which is used as baseline, and two DL based approaches using Autoencoders (AEs) and Convolutional Neural Networks (CNNs). Experiments were carried out in two datasets from two different municipalities in Brazil, Ipuã in São Paulo state and Campo Verde in Mato Grosso state. It is shown that CNN and AE outperformed the traditional approach based on image stacking in terms of Overall Accuracy and Class Accuracy.**

***Index Terms*—Crop Recognition; Multitemporal Images; Autoencoders, Convolutional Neural Networks.**

I. Introduction

Prediction of yields, estimation of food production and precise and accurate agricultural statistics are crucial to anticipate the market behavior, create new strategies for agriculture and develop economic planning by government and private agencies. Remote sensing data has been widely used for this purpose because it provides a cost-effective tool for agricultural monitoring and management. With the launch of more satellites, high spatial and temporal resolution images with low revisit time are affordable, which allows to capture changes as crops evolve through their characteristic phenological stages. In tropical areas, crop dynamics are complex due to multiple agricultural practices such as irrigation, non-tillage, crop rotation and multiple harvest per year.

Among the proposed approaches for crop recognition, there are three main groups: Pixel-wise, Object-based and Context-based. Pixel-wise methods take information from every pixel individually and classify each image separately. Neural Networks (NNs), Support Vector Machines (SVMs) and Random Forest (RF) classifiers have been applied for this purpose [1], [2]. These methods have a major limitation, because they ignore spatial and temporal context. Object-based methods [3] partially capture spatial context by classifying segments. Their main limitation is due to the fact that conventional segmentation algorithms rely only on the data and fully disregard semantics. Context-based classification approaches take into account contextual information in the spatial and/or temporal domains. The temporal context carries information about the phenological cycles, which is essential to properly discriminate among different crop types [4], [5].

Hidden Markov Models (HMMs) has been used to model crop's phenology over time [6], [7]. Spatio-temporal Markov Random Fields (MRFs) [8] and Conditional Random Fields (CRFs) [9] have been proposed to unify both, spatial and temporal information. These approaches achieve higher accuracies than other methods at the cost of a higher computational effort and more labeled samples for a supervised training. They also require a prior feature selection analysis.

On the other hand, Deep Learning (DL) techniques have recently become very popular in the scientific community particularly for image classification. Such techniques are able to learn features automatically from non-labeled samples. Deep Belief Networks (DBNs) [10], Autoencoders (AEs) [11], [12], Convolutional Neural Networks (CNNs) [13] and Recurrent Neural Networks (RNN) [14] are the main approaches in DL. CNNs and RNNs perform a supervised training of the whole network. In contrast, DBNs and AEs train one layer at a time in an unsupervised manner, reducing the need of collecting many labeled samples.

State-of-the-art land-cover and crop type classification techniques implement DL approaches using spatial and temporal context [12], [15]. Kussul in [16] used temporal information in a multilevel DL architecture for crop type classification in a heterogeneous environment in Ukraine with scenes acquired by Landsat-8 and Sentinel-1A satellites. The crop calendar of these data is from September to July for winter crops, and

from April to October for spring and summer crops. Crop's dynamic is simpler than in the tropics with only one crop per field per season. To our knowledge this is the first work that approaches crop mapping using deep learning strategies on a database from tropical regions.

In this paper, we perform a comparative analysis of supervised and unsupervised DL techniques for crop classification on sequences of multitemporal remote sensing images. Both techniques have been tested in two datasets composed by Optical (Landsat 5/7) and SAR (Sentinel-1) image sequences from two different municipalities in Brazil, Ipuã in São Paulo state and Campo Verde in Mato Grosso state, respectively. These are representative agricultural areas in Brazil, where crops dynamic is more complicated than in temperate regions.

The remainder of this paper is organized as follows. Section II explains the fundamentals of AEs and CNNs. Section III introduces the methods evaluated in this work to extract multitemporal feature representations. Section IV presents the datasets used in our experiments, the features extracted from them and the experimental protocol. Section V shows the results obtained in our experiments and discusses them. Finally, Section VI summarizes the conclusions drawn from our results.

II. Fundamentals

A. Autoencoders (AEs)

An Autoencoder is a Neural Network architecture formed by two modules, an encoder and a decoder [17]. As shown in Figure 1, the encoder projects the d-dimensional input data $\mathbf{x}$ onto a k-dimensional (k is the number of hidden nodes) space through a nonlinear mapping function f:

$$f(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1}$$

where $\mathbf{W}$ is a $k \times d$ weight matrix, $\mathbf{b}$ is an d-dimensional bias vector and s is a nonlinear function. In contrast, the decoder projects back to the original d-dimensional input space through another mapping function h:

$$\hat{\mathbf{x}} = h(f(\mathbf{x})) = s(\mathbf{W}' f(\mathbf{x}) + \mathbf{b}') \tag{2}$$

where $\mathbf{W}'$ is usually constrained to be equal to $\mathbf{W}^T$. The parameters (the weights and bias) of the autoencoder are learned using the backpropagation algorithm by minimizing a cost function L, such as the one given by Equation 3.

$$L = \sum_i \|\mathbf{x}_i - h(f(\mathbf{x}_i))\|_2^2 + \alpha \|\mathbf{W}\|_2^2 + \beta KL(\rho||\hat{\rho}) \tag{3}$$

where $\sum_i \|\mathbf{x}_i - h(f(\mathbf{x}_i))\|_2^2$, $\|\mathbf{W}\|_2^2$ and $KL(\rho||\hat{\rho})$ correspond to the reconstruction error, weight decay and sparsity penalty terms respectively, controlled by the weighting coefficients α and β. $\|\mathbf{W}\|_2^2$ is the Frobenius norm of the weight matrix. The term $KL(\rho||\hat{\rho})$ is the Kullback-Leibler (KL) divergence between ρ and $\hat{\rho}$, the ideal and actual distribution of the average activation over all hidden units, respectively.

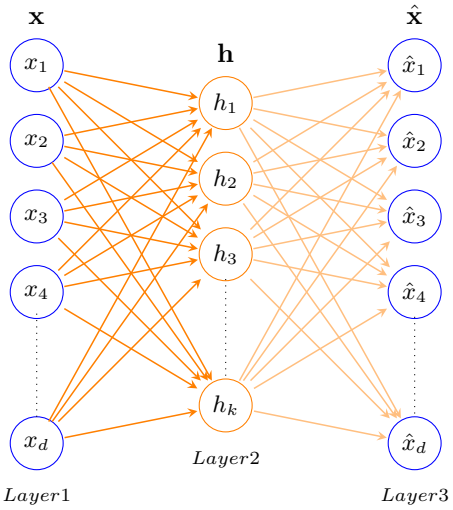

Fig. 1. Autoencoders architecture. Layer 1 corresponds to the Input data, layer 2 to the Encoder, and layer 3 to the Decoder.

Once the autoencoder's parameters have been learned, the encoder function f is used to generate a new set of features from the original data. This new set of features is expected to be more discriminative than the original ones.

B. Convolutional Neural Networks (CNNs)

CNNs pass the image through a series of several layers of small neuron/kernel collections where each one looks at an small portion of an image and gets an output. The output can be a single class or a set of class probabilities that best describes the image. Typically, CNNs use small convolutional kernels. In consequence, CNNs involve fewer parameters than fully connected neural networks [18]. The simplest CNNs architectures is shown in Figure 2 and consists of several layers. These layers can be one of the following types:

1) Convolutional: during the convolution the kernels slide over all pixels of the input image. This kernel/weights is an array with the same depth of the input. Each of these kernels can be seen as feature identifiers.
2) Pooling: this is a downsampling layer whose input is the output of a convolutional layer. The pooling layer reduces the amount of data in spatial domain (depth remains unchanged) in order to reduce the number of parameters of the subsequent layers, to reduce the computation cost and also to control overfitting.
3) Fully-Connected: takes an input volume layer (previous layer) and convert it to an one-dimensional layer, by connecting all neurons of the previous layer to every single neuron of the fully-connected layer.

The final layer in a CNNs contains a single node for each class. In our model, a softmax activation function was used to get the posterior probability for the different classes. More complex CNNs architectures usually have many convolutional and pooling layers [19]–[21].

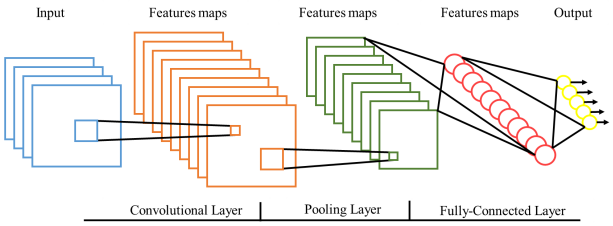

Fig. 2. CNNs basic architecture.

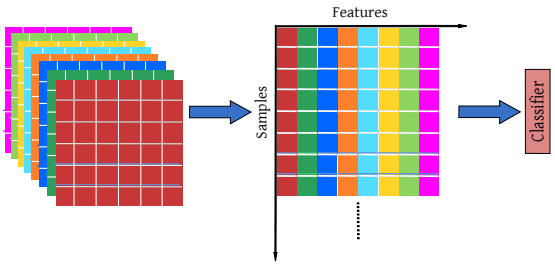

Fig. 3. Image Stacking. First, images in a multitemporal sequence are stacked. Then, a classification algorithm is trained using this stack.

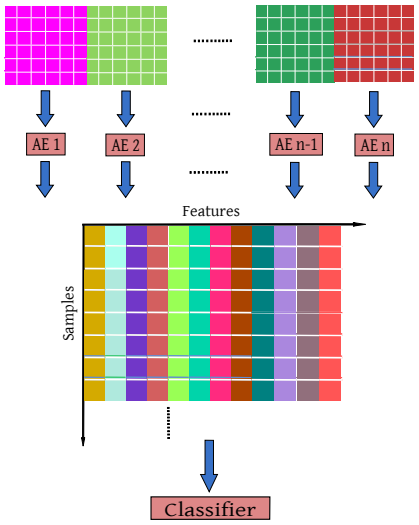

Fig. 4. Unsupervised Multitemporal based on Autoencoders approach. First, an Autoencoder is trained for each image of the multitemporal sequence. Then, these outputs are stacked. Finally, a classification algorithm is trained using this stack.

III. Methods

This section describes the methods evaluated in this work for crop recognition: *Image Stacking* and two approaches based on DL's techniques, *Unsupervised Multitemporal based on Autoencoders* and *Supervised Multitemporal based on Convolutional Neural Networks*. *Image Stacking* [22], [23] is the most widely exploited approach in the literature for multitemporal remote sensing image analysis, and in this work, we used it as the baseline. Likewise, AE and CNN are the DL methods most widely used in the literature. An explanation of each method is given in the following.

A. Image Stacking (IS)

The traditional approach consists in stacking all images of the multitemporal sequence to assemble, for each pixel location, a descriptor that comprises the features of all epochs. The representations built in this way are used to train a classifier that assigns a class label to each pixel along the sequence. In this approach no spatial context is taken into account. Figure 3 illustrates the process flow of this method.

B. Unsupervised Multitemporal based on Autoencoders (UMAE)

In this approach, temporal and spatial contextual information are exploited as part of the AE training. In this method, an AE is trained for each epoch separately. Here, the final descriptor for each pixel of the image sequence will be assembled by concatenating the corresponding new learned pixel representation of each epoch. Similar to *IS*, a classification model is built from the resulting descriptors. Figure 4 summarizes the method. The descriptor $\mathbf{x}$ of a pixel in each image is a $d \times w^2$ dimensional vector that comprises the w-by-w-by-d patch centered at that pixel, where d is the depth of each input image. The training/inference procedure consists of the following steps:

1) Select randomly for each image in the sequence, M pixel descriptors.
2) Train for each image, an AE using the corresponding set of M vectors. Then, these sets are previously standardized to zero-mean and unit-variance.
3) Compute the representation of each pixel in each image using the encoded mapping functions learned by each AE in the previous step.
4) Take as final representation of each pixel the concatenation of its single date AE representations over the whole sequence. Notice that pixels at the same location will share the same representation over all epochs.
5) Train a classifier for the target image using labeled pixel samples and their representations computed in the previous step.
6) Apply the classifier trained in the previous step to all pixels of the target image not used for training, using the representations computed in step 4.

C. Supervised Multi temporal based on Convolutional Neural Networks (SMCNN)

The *CNN* architecture tested is this study is shown in Figure 2. It consists of four layers, from left to right: convolutional, max-pooling, fully connected and softmax layer. Similarly to [16], we train a CNN to describe a pixel location taking information of the neighborhood. The descriptor $\mathbf{x}$ of a pixel in each image is a $d \times n \times w^2$ dimensional vector that comprises the w-by-w-by-d patches centered at the same position of all n images, where d is the depth of each input image. Notice that pixels at the same location will share the same descriptor over all epochs. The training/inference procedure consists of the following steps:

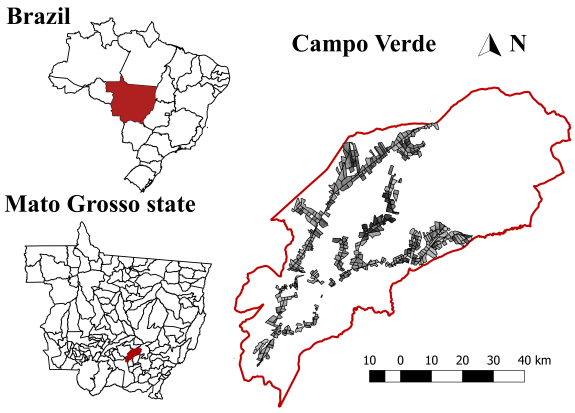

Fig. 5. Study area: Campo Verde, Mato Grosso state, Brazil.

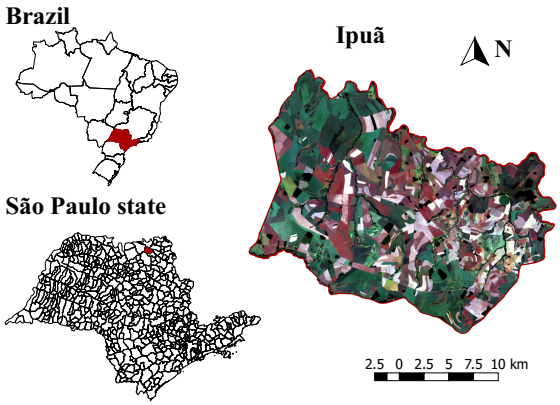

Fig. 6. Study area: Ipuã, São Paulo state, Brazil.

1) Train a CNN in a supervised fashion for the target image using labeled pixel samples and their corresponding descriptors.
2) Apply the CNN trained in the previous step to the descriptors of all pixels of the target image not used for training.

IV. Experiments

A. Datasets

1) Ipuã: Ipuã municipality in the state of São Paulo, Brazil has an extension of 465 km^2 approximately (see Figure 6). A sequence of 9 Landsat scenes, from August 2000 to July 2001, was taken, from either Landsat-5 (TM) or Landsat-7 (ETM+) with 30 m spatial resolution, each image having approximately 500K pixels. The reference for each epoch was produced manually by a human expert.

The most common crops are *Sugarcane*, *Soybean* and *Maize*. In our study, we also included two classes related to no crops: *Prepared Soil*, which corresponds to ploughing and soil grooming phases, and *Post-Harvest*, characterized by vegetation residues lying on the ground. To complete the set of classes, *Pasture*, *Riparian Forest* and *Others* were also included in our model. The last one represents minor crops as well as rivers and urban areas. Figure 7 shows the class occurrences per image in the dataset. Notice that some classes appears only in two or three epochs due to its shorter cycle (*Maize*) or the gap in the acquisition dates (*Soybean*) during cloudy months. On the other hand, *Sugarcane*, which is a long cycle crop, appears in all epochs.

2) Campo Verde: Campo Verde municipality in the state of Mato Grosso, Brazil has an extension of 4782 km^2 approximately (see Figure 5). A total of 27 Level 1 Interferometric Wide Swath (IWS) mode Ground Range Detected (GRD) Sentinel-1 products in VV and VH polarizations were used to cover all Campo Verde municipality from October 2015 to July 2016 resulting in a sequence of 14 images, with two images per month for November, December, March, May and July and only one image for October, January, February and June. These images were geometrically corrected using a Range Doppler terrain correction with a Digital Elevation Model (DEM) from SRTM, radiometrically calibrated to a backscatter coefficient (sigma nought (σ^0) in this case), converted to db, co-registered using a RapidEye mosaic (5 m spatial resolution) and georeferenced to UTM projection Zone 21S and Datum WGS84.

The main crops found in this area are: *Soybean*, *Maize* and *Cotton*. Also, there are some minor crops such as *Beans* and *Sorghum*. As *non-commercial crops* (*NCC*), *Millet*, *Brachiaria* and *Crotalaria* were considered. Other classes present in the dataset are *Pasture*, *Eucalyptus*, *Soil*, *Turfgrass* and *Cerrado*. Figure 8 shows the class occurrences per images in the dataset. Similar to Ipuã dataset, the number of crops per image changes along the whole image sequence due to the different phenological cycles of each culture.

B. Feature Extraction

Hand-crafted features were computed for each dataset and used for the experiments as well as those learned by the aforementioned methods in Section III. For *Ipuã*, the pixel spectral information from bands 1-5 and 7, and the Normalized Difference Vegetation Index (NDVI) were used as feature vector. For *Campo Verde*, features extracted from the Gray Level Co-occurence Matrix (GLCM) or pixels values were selected depending on the evaluated method. For the *IS* approach, GLCM features were employed and pixel values for

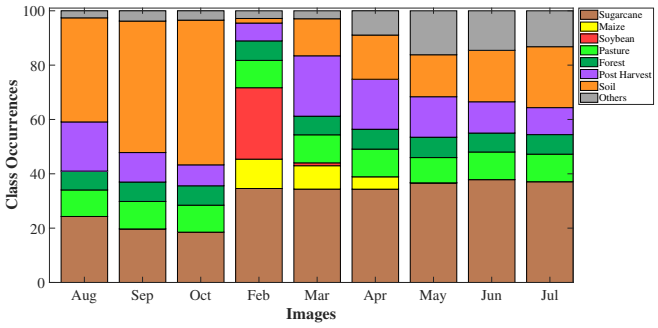

Fig. 7. Class occurrences per image in Ipuã dataset.

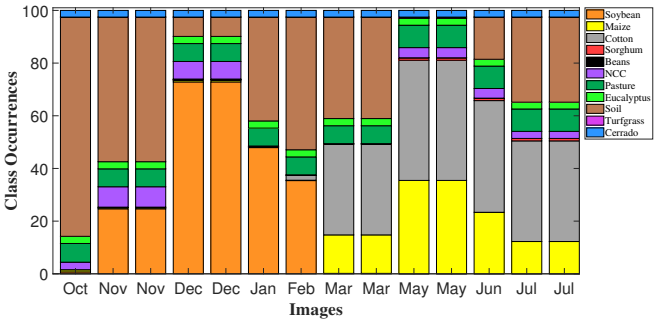

Fig. 8. Class occurrences per image in Campo Verde dataset.

the other methods based on deep learning techniques. As in [24], four features were computed for each image band from the GLCM (correlation, homogeneity, mean and variance) in four directions (0, 45, 90 and 135 degrees) using 3×3 windows. Then, each pixel was represented by a feature vector of dimensionality 32.

C. Experimental Protocol

For the *IS* and *UMAE* approaches we used a Random Forest (RF) classifier, specifically the implementation in Scikit-Learn [25]. The DL based algorithms were implemented using the Theano framework [26]. For all experiments, a manual parameter tuning procedure was carried out to get the best possible configuration. For the RF classifier, the number of random trees and its maximum depth was set to 250 and 25, respectively; for the *UMAE*, the size of patches was 3×3, the hidden layer was set to 256 neural units and the sparsity penalty parameter to 10^{-4}; for the *SMCNN*, the size of the neighborhood was 5×5, the convolutional layer was set to 256 kernels of 3×3 and the FC layer to 256 neural units. We used a dropout of 0.5 at FC layer to train the CNN.

The protocol followed in the experiments is described next.

Algorithm 1: Experimental Protocol

Input: Sequence of n images $\{I_1, I_2, I_3, ..., I_n\}$

1: **for** $t = n$ to 1 **do**
2: **for** $l = 0$ to $t - 1$ **do**
3: Stack the feature pixel representation that comprises the whole sequence of l images from I_t to I_{t-l}, according to the selected method.
4: Train and evaluate a classifier/DL approach using the stacked features and the reference for the last image in the sequence I_t, respectively.
5: **end for**
6: **end for**

For a given image sequence we classified only the image of the last epoch. We started with a single image in the sequence and repeated the experiment by adding earlier images successively (see Figure 7).

Two sequences were considered for both datasets based on the class distribution per image in Figure 7 and 8. For Ipuã dataset, these sequences were selected with focus on *Maize* and *Sugarcane*, wich come about only between February and April and between February and July for *Sugarcane*, respectively; although *Sugarcane* appears throughout the whole images sequence, only images from February on were considered because of the missing images between November, 2000 and January, 2001. Likewise, for *Campo Verde* dataset, we took one sequence from November, 2014 to February, 2015, where there is mainly *Soybean*, and another one from March to July, where *Cotton* and *Maize* are the major crops. Notice that October epoch was excluded from the analysis because there were not enough samples of any class of interest.

In order to balance the number of training samples for all classes we replicated samples of some less abundant classes on both datasets. In particular, for *Ipuã* dataset we selected 5,000 samples per class while for *Campo Verde*, 50,000 samples per class were selected.

V. Results

Results for both Ipuã sequences are shown in Figure 9, Figure 10 and Table I. Each figure summarizes, for different sequence lengths, the *Accumulated Class Accuracy (AA)* obtained by the methods described in Section III. From left to right, in each bar group, each bar corresponds to *IS*, *UMAE* and *SMCNN* methods, respectively. The maximum possible value of each bar is equal to the number of classes$\times$100%. This kind of comparison allow us to analyze the contribution of temporal information to the classification performance as the sequence length increases by adding data from the past. Similarly, Table I summarizes the results of the same experiments in terms of *Overall Accuracy (OA)*.

Figure 9 and Figure 10 show improvements on the *AA* of up to 76% and 60%, respectively for *IS* as more images were considered. Notice that performance for most classes improved as the sequence length increased. A similar behavior was observed for *UMAE* and *SMCNN*. A significant improvement was obtained when a second image was added to the sequence. As more images were added, the incremental improvements decreased. This behavior can be explained by considering that information from earlier images tend to be less relevant as we move back in time.

TABLE I
Overall Accuracy for both sequences evaluated from Ipuã dataset: From Feb to April and from Feb to Jul, classifying always the last image in each sequence.

Sequence length	*OA*(%)					
	Feb - Apr			Feb - Jul		
	IS	*UMAE*	*SMCNN*	*IS*	*UMAE*	*SMCNN*
1	78.7	82.9	85.5	72.9	76.8	80.6
2	82.9	85.2	86.5	78.0	80.0	82.2
3	84.6	85.9	86.9	79.9	81.8	81.3
4	–	–	–	82.3	82.1	82.6
5	–	–	–	82.3	82.5	83.8
6	–	–	–	82.6	81.9	84.4

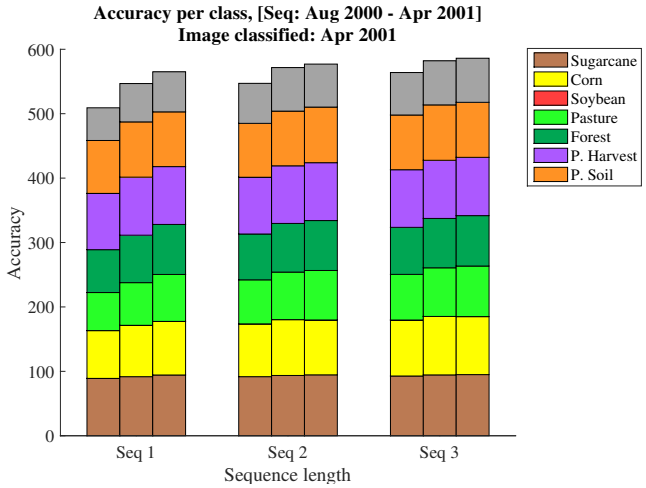

Fig. 9. *Accumulated Class Accuracy* for different sequence lengths for the first sequence from Ipuã dataset. From left to right in each bar group: *Is*, *UMAE* and *SMCNN* approaches.

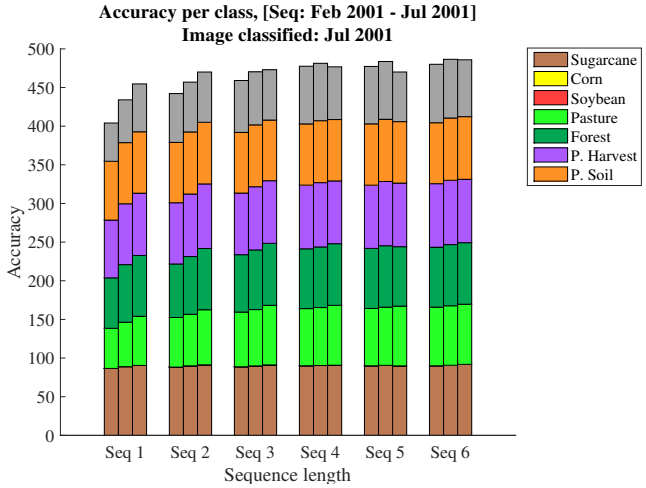

Fig. 10. Accumulated *Class Accuracy* for different sequence lengths for the second sequence from Ipuã dataset. From left to right in each bar group: *Is*, *UMAE* and *SMCNN* approaches.

TABLE II
OVERALL ACCURACY FOR BOTH SEQUENCES EVALUATED FROM CAMPO VERDE DATASET: FROM NOV TO FEB AND FROM MAR TO JUL, CLASSIFYING ALWAYS THE LAST IMAGE IN EACH SEQUENCE.

Sequence length	*OA*(%)					
	Nov - Feb			Mar - Jul		
	IS	*UMAE*	*SMCNN*	*IS*	*UMAE*	*SMCNN*
1	24.4	47.2	36.7	18.2	42.6	38.5
2	51.0	53.4	53.9	47.2	52.8	54.5
3	56.5	58.0	62.4	52.7	56.6	59.1
4	59.4	60.8	66.5	57.5	59.9	63.7
5	63.7	64.8	70.6	60.3	62.2	61.4
6	64.8	65.8	71.2	62.9	64.2	66.5
7	–	–	–	63.5	64.6	67.1

Results from Table I are consistent with the results exhibited in Figure 9 and Figure 10; *OA* improved as the sequence length increased. For instance, the results for *IS* improved up to 6% and 10% in terms of *OA* for the shorter and the longer *Ipuã* sequences, respectively.

As for the DL based techniques with respect to the *IS*, major improvements were achieved primarily in relation to the monotemporal classification (first group of bars in Figure 9 and Figure 10 or the row of sequence length equal to 1 in Table I); up to 45% for both sequences in terms of *AA* or 8% for the *OA* metric, while for longer sequences (see Figure 9), the improvement decreases to 7% for higher sequence lengths.

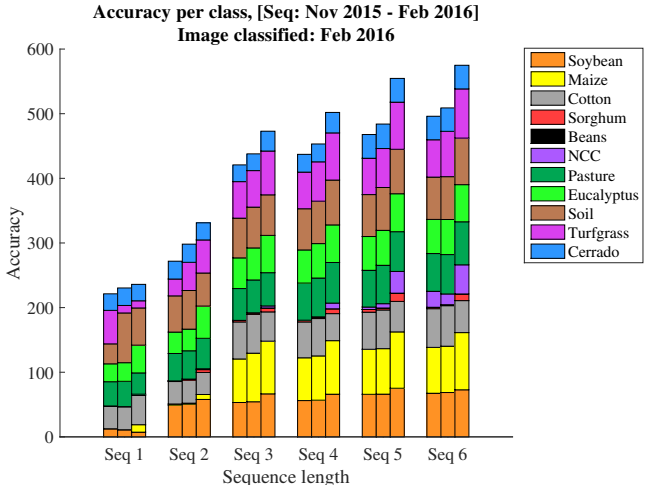

Fig. 11. Accumulated *Class Accuracy* for different sequence lengths for the first sequence from *Campo Verde* dataset. From left to right in each bar group: *IS*, *UMAE* and *SMCNN* approaches.

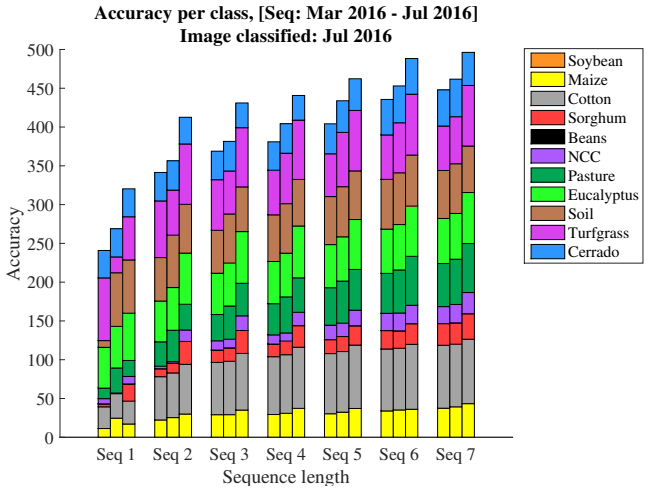

Fig. 12. Accumulated *Class Accuracy* for different sequence lengths for the second sequence from *Campo Verde* dataset. From left to right in each bar group: *IS*, *UMAE* and *SMCNN* approaches.

The results for the *Campo Verde* dataset are shown in Figure 11, Figure 12 and Table II. Similar to the results drawn from the experiments on Ipuã, temporal information helped a lot

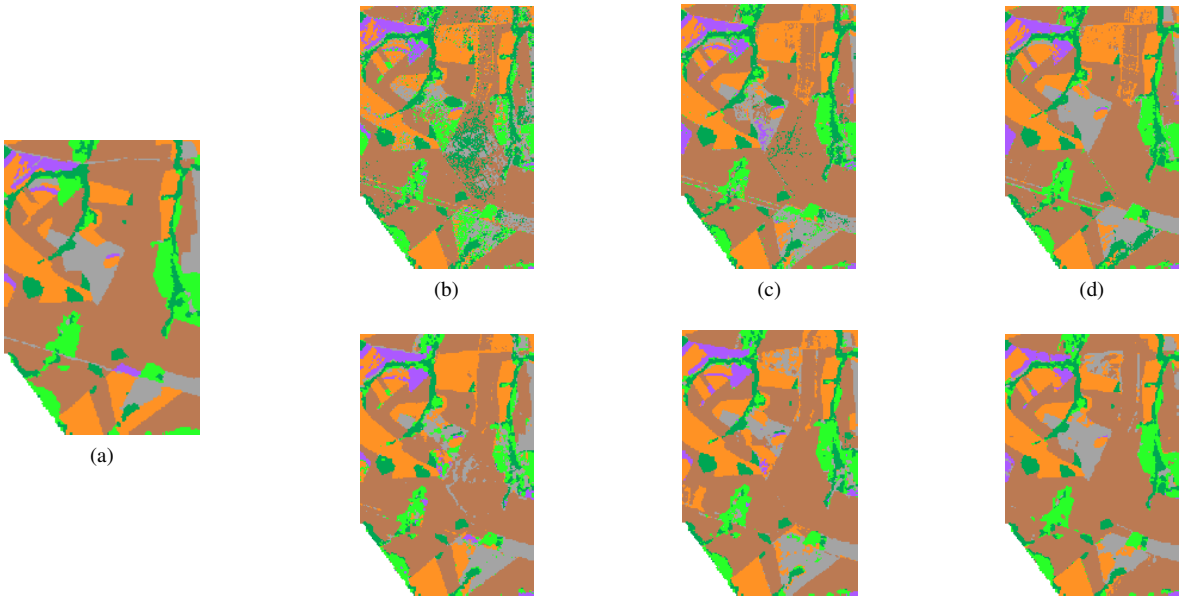

Fig. 13. Prediction maps of a selected interest region of *Ipuã* dataset. (a) is the reference map, (b), (c) and (d) are the *IS* approach prediction maps for sequence lengths of 1, 3 and 6, respectively. (e), (f) and (g) are the corresponding *SMCNN* approach predictions maps.

Fig. 14. Prediction maps of a selected interest region of *Campo Verde* dataset. (a) is the reference map, (b), (c) and (d) are the *IS* approach prediction maps for sequence lengths of 1, 3 and 6, respectively. (e), (f) and (g) are the corresponding *SMCNN* approach predictions maps.

to improve the classification performance for all evaluated approaches. This behavior becomes more evident for *Campo Verde* due to the low rates in the monotemporal classification, less than 50% *OA* (see Table II) due to classes with *Class Accuracy* close to zero. For instance, it can be seen in Figure 11 that *Maize* bars only increased for sequences comprising more than three epochs. The *Campo Verde* data set is highly unbalanced in terms of number of samples per class. Though we balanced the training set by replicating samples of minority classes, they still presented low *AA*.

Like the results for Ipuã, the results obtained with features learned by AEs were superior to the results achieved with handcrafted features. However, *SMCNN* results outperformed all other methods in almost all cases. As the *Campo Verde* dataset has more training samples than *Ipuã* dataset, a better network's parameters tuning could be achieved. So, the superiority of *SMCNN* to the other approaches was higher for *Campo Verde*.

Finally, Figure 13 and Figure 14 show snips of the reference and classification maps obtained by the *IS* and *SMCNN* approaches for *Ipuã* and *Campo Verde* (sequence 1), respectively, in three cases: monotemporal classification and multitemporal for sequence of 3 and 6 images, respectively. We see that the classification improved as more images were added to the sequence. For longer sequences, the prediction maps came closer to the reference. Comparing the performance of *IS* and *SMCNN*, we notice that, in both datasets, the salt&pepper effect was less significant for *SMCNN*. This difference can be related to the ability of *CNNs* to capture spatial context.

VI. Conclusion

In this work, we developed a comparative analysis of supervised and unsupervised DL techniques for crop recognition in tropical regions, on an optical and on a SAR dataset. Results confirmed that temporal information plays an important role.

On the other hand, Deep Learning techniques outperformed the conventional image stacking strategy in almost all experiments. In particular, *SMCNN* was the best performing among all evaluated methods, mainly for the SAR dataset.

Future works will be focused in other *CNN* architectures. In particular, *Recurrent Neural Networks* architectures will be considered in the continuation of this research.

References

[1] M. K. Mosleh, Q. K. Hassan, and E. H. Chowdhury, "Application of remote sensors in mapping rice area and forecasting its production: A review," *Sensors*, vol. 15, no. 1, pp. 769–791, 2015.

[2] R. Sonobe, H. Tani, X. Wang, N. Kobayashi, and H. Shimamura, "Discrimination of crop types with terrasar-x-derived information," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 83, pp. 2–13, 2015.

[3] X. Jiao, J. M. Kovacs, J. Shang, H. McNairn, D. Walters, B. Ma, and X. Geng, "Object-oriented crop mapping and monitoring using multitemporal polarimetric radarsat-2 data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 96, pp. 38–46, 2014.

[4] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[5] G. Moser and S. B. Serpico, "Multitemporal region-based classification of high-resolution images by markov random fields and multiscale segmentation," in *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*. IEEE, 2011, pp. 102–105.

[6] Y. Shen, L. Wu, L. Di, G. Yu, H. Tang, G. Yu, and Y. Shao, "Hidden markov models for real-time estimation of corn progress stages using modis and meteorological data," *Remote Sensing*, vol. 5, no. 4, pp. 1734–1753, 2013.

[7] S. Siachalou, G. Mallinis, and M. Tsakiri-Strati, "A hidden markov models approach for crop classification: Linking crop phenology to time series of multi-sensor remote sensing data," *Remote Sensing*, vol. 7, no. 4, pp. 3633–3650, 2015.

[8] D. Liu, K. Song, J. R. Townshend, and P. Gong, "Using local transition probability models in markov random fields for forest change detection," *Remote Sensing of Environment*, vol. 112, no. 5, pp. 2222–2231, 2008.

[9] B. K. Kenduiywo, D. Bargiel, and U. Soergel, "Higher order dynamic conditional random fields ensemble for crop type classification in radar images," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin, "Deep learning with hierarchical convolutional factor analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1887–1901, 2013.

[12] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, 2016.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] H. Wu and S. Prasad, "Convolutional recurrent neural networks forhyperspectral data classification," *Remote Sensing*, vol. 9, no. 3, p. 298, 2017.

[15] N. Kussul, A. Shelestov, M. Lavreniuk, I. Butko, and S. Skakun, "Deep learning approach for large scale land cover mapping based on remote sensing data fusion," in *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*. IEEE, 2016, pp. 198–201.

[16] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[17] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[22] A. Schneider and C. E. Woodcock, "Compact, dispersed, fragmented, extensive? a comparison of urban growth in twenty-five global cities using remotely sensed data, pattern metrics and census information," *Urban Studies*, vol. 45, no. 3, pp. 659–692, 2008.

[23] A. Schneider, "Monitoring land cover change in urban and peri-urban areas using dense time stacks of landsat satellite data and a data mining approach," *Remote Sensing of Environment*, vol. 124, pp. 689–704, 2012.

[24] B. Kenduiywo, D. Bargiel, and U. Soergel, "Crop type mapping from a sequence of terrasar-x images with dynamic conditional random fields," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 59–66, 2016.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688