DEEP LEARNING FOR THE CLASSIFICATION OF SENTINEL-2 IMAGE TIME SERIES

Charlotte Pelletier, Geoffrey I. Webb, François Petitjean

Faculty of Information Technology
25 Exhibition Walk
Monash University
3800 Melbourne - Australia

ABSTRACT

Satellite image time series (SITS) have proven to be essential for accurate and up-to-date land cover mapping over large areas. Most works about SITS have focused on the use of traditional classification algorithms such as Random Forests (RFs). Deep learning algorithms have been very successful for supervised tasks, in particular for data that exhibit a structure between attributes, such as space or time. In this work, we compare for the first time RFs to the two leading deep learning algorithms for handling temporal data: Recurrent Neural Networks (RNNs) and temporal Convolutional Neural Networks (TempCNNs). We carry out a large experiment using Sentinel-2 time series. We compare both accuracy and computational times to classify 10,980 km^2 over Australia. The results highlights the good performance of TemCNNs that obtain the highest accuracy. They also show that RNNs might be less suited for large scale study as they have higher runtime complexity.

Index Terms— satellite image time series, land cover mapping, Sentinel-2, deep learning

1. INTRODUCTION

On March 7 2017, the European Space Agency (ESA) successfully put its latest high-resolution optical satellite, Sentinel-2B, into orbit. Both Sentinel-2A and 2B are now acquiring pictures of the Earth every five days at high spatial and spectral resolutions [1]. These new satellite image time series (SITS) are a powerful tool for the management of territories and climate studies. Among these applications, we focus here on the production of accurate and up-to-date land cover maps, such as the one displayed in Figure 1.

For this task, supervised classification algorithms have shown their potential, especially traditional algorithms such as Random Forests (RFs) and Support Vector Machines (SVMs) [2]. For example, RFs are able to deal with the high dimensionality of SITS [3]. However, these methods are oblivious to the temporal structure of SITS: a shuffle of the images in the series leads to similar results. Hence, specific temporal dynamics will not be taken into account, potentially leading to a decrease in the quality of the maps.

To make the most of SITS temporal dimension, recent works have explored the potential of deep learning models. In particular, Recurrent Neural Networks (RNNs), developed initially for sequence data, have been successfully applied to multi-temporal Synthetic Aperture Radar (SAR) data [4, 5] and optical images [6]. Meanwhile, we have also proposed the use of temporal Convolutional Neural Networks (TempCNNs), where convolutions are applied in the temporal domain instead of the spatial domain, for the classification of optical SITS [7]. Both RNNs and TempCNNs have not been compared yet in the literature. Moreover, RNNs have been applied to short time series (less than 25 time stamps) and on small scale studies: fewer training instances (below 100,000) and/or small studied area (below 100 km $\times$ 100 km).

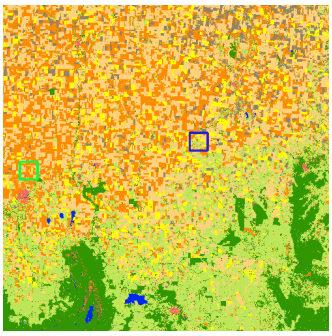

Fig. 1. An example land cover map.

Hence, this paper aims at filling the gap and studies the performance of traditional and deep learning algorithms – RFs, RNNs, and TempCNNs – for SITS classification on a large scale problem. More specifically, we carry out experiments on 60 Sentinel-2 images acquired over Australia. We perform quantitative – accuracy as well as runtime complexity – and qualitative evaluations to highlight advantages and potential drawbacks of the three algorithms. Note that this work focuses only on the use of the temporal structure, and does not cover the use of the spatial structure of SITS.

The remainder of this paper is organized as follows. The trained classification algorithms are described in detail in Section 2. Then, Section 3 introduces the data and the experimental settings used in this work. Section 4 provides some quantitative results as well as visual inspection of the produced land cover maps. Finally, conclusions are drawn in Section 5.

2. CLASSIFICATION ALGORITHMS

2.1. Random Forests

Research into time series classification for remote sensing has concluded that RFs are generally the most effective classifiers [8]. More specifically, RFs present several advantages: they handle the high dimensionality of SITS data, they are robust to a small presence of

This research was supported by the Australian Research Council under grant DE170100037. This material is based upon work supported by the Air Force Office of Scientific Research, Asian Office of Aerospace Research and Development (AOARD) under award number FA2386-18-1-4030.

mislabeled data [9], they perform well on large scale area [10], and their parameters are easy to set [3].

RFs is an ensemble method that learns a set of binary decision trees [11]. To increase the diversity among the ensemble, bootstrap instances are used to build each tree. In addition, a random subset of features is used to split the data at each node. Only the best split is used, as assessed by a split effectiveness test such as the maximization of the node purity. The tree construction ends when all the nodes are pure or when a user-defined criterion is met, *e.g* a maximum depth or a minimum node size.

2.2. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) have been proposed in remote sensing for the classification of SITS. They have shown their potential by outperforming traditional algorithms such as RFs or Support Vector Machines (SVMs) on small area case-studies (less than 100,000 training instances) [4, 12, 5, 6].

First developed for sequential data, RNN models have the specificity of sharing the learned features across different positions, *e.g.* across different words in a sentence. This makes RNNs extremely efficient to produce an output at each time step, such as in machine translation. They have a high computational cost: back-propagating the error at each time step increases drastically the training time, and may cause learning issues such as vanishing gradient. Most recent RNN architectures use Gated Recurrent Units (GRUs) that help to capture long distance connections and solve the vanishing gradient issue. They are composed of a memory cell as well as update and reset gates to decide how much new information to add and how much past information to forget. Although RNNs are appealing for time series, their potential for time series classification is less straightforward. Contrary to machine translation tasks, time series classification requires only one output for the whole time series. Hence, RNNs might be less suited for this specific task.

2.3. Temporal Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been widely applied on various computer vision tasks [13]. They also lead the state-of-the-art for some remote sensing tasks, including the classification of hyper-spectral images. Despite their huge success, they have been investigated only recently on time series classification [14]. The specificity of these networks is to apply convolutions in the temporal domain rather than the spatial domain.

3. MATERIAL AND METHODS

3.1. Sentinel-2 images

The experiments are carried out on 60 Sentinel-2 image time series extracted over Victoria in Australia (tile T54HXE). Each image is composed of ten spectral bands at 10 m spatial resolution (the sixth bands at 20 m are spatially interpolated at 10 m). The series starts in August 2017, and ends in July 2018. Figure 2 displays a false color Sentinel-2 image from October 16 2017, and its location in Victoria. This studied area is composed mainly of crops, some native forests and few cities.

Atmospheric, adjacency and slope effects are corrected by using MACCS/MAJA processing chain [15]. Images are eventually gap-filled by using a linear temporal interpolation on a regular temporal grid with a time gap of five days [10]. After the gapfilling operation, a total of 730 features describes each pixel: 73 interpolated dates multiply by 10 spectral bands.

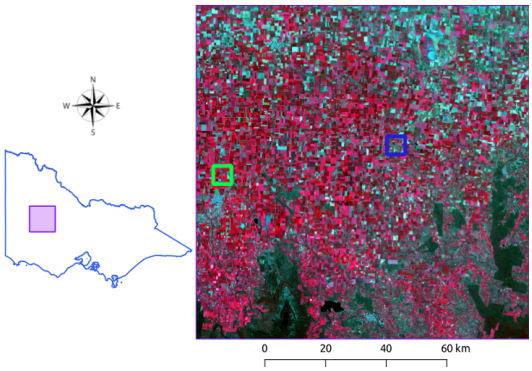

Fig. 2. Studied area over Victoria (Australia). Sentinel-2 image in false color from October 16 2017.

3.2. Reference data

The reference data has been kindly provided by Agriculture Victoria, that has collected 250 m $\times$ 250 m homogeneous areas for vegetation and water classes. For vegetation classes (bare ground, canola, legumes, cereals and grassland), we have extended the reference data at the parcel level. Finally, we have manually extracted forests, water surfaces and urban areas by photo-interpreting non-cloudy Sentinel-2 images. Table 1 displays the number of pixels and polygons (*i.e.* parcel-level for crop classes) for each land cover type. The legend used in Section 4 is displayed in the last column.

Table 1. Number of instances per class counted at pixel- and polygon-level.

Classes	Pixels	Polygons	Legend
Bare ground	69,311	9	
Canola	259,540	53	
Cereals	1,087,238	174	
Legumes	1,019,335	164	
Grassland	179,901	101	
Forests	150,470	30	
Water	31,708	8	
Urban	30,176	14	
Total	**2,827,679**	**553**	

Table 1 shows that the number of available instances varies across the different land cover classes. The most represented classes include crop classes such as cereals, legumes and canola. For the evaluation in Section 4.1, 70 % of instances are used for training (about 1.9M instances), and the remaining 30 % for testing (about 900k instances). The split is performed at the polygon-level to ensure that pixels from the same polygons are not used as train instances as well as test instances. Each is experiment is run five times: we report mean and standard deviation values.

3.3. Classification algorithm settings

To complete RF experiments of Section 4, Scikit-Learn library (Python) has been used with standard parameter settings [3]: 100 trees at the maximum depth of 25, a minimum node size of 1, and a number of randomly selected variables per node equal to the square root of the total number of features.

Concerning the architecture of the trained RNN models, we decide to stack three Gated Recurrent Units (GRUs) composed of 192

units, and a Softmax layer that outputs the predictions. The architecture is similar to the recent work in [4], but we use bidirectional GRUs to use both past and future information[1].

Following our recent work on TempCNNs [7], we use here a CNN network composed of three convolutional layers composed of 64 units, one fully-connected layer composed of 256 units, then the Softmax layer.

For both deep learning networks, weight and bias parameters are optimized by using Adam (Adaptive moment optimization) with its default parameter values [16]. Batch size is set to 64, and the number of epochs to 20. The cross-entropy loss computed on a validation set is monitored: the best model is selected by using an early stopping mechanism on the validation loss. The validation set is composed of 10 % of the training data extracted at the polygon-level.

Networks are implemented through Keras [17], with Tensorflow as the backend [18]. To facilitate others to build on this work, we have made our code available at `https://github.com/charlotte-pel/igarss2019-dl4sits`. To compare train and test times, all the experiments have been carried out on the same machine with 12 Central Processing Units (CPUs) and 256 GB of RAM. To test the computational gain of Graphical Processing Units (GPUs), we have also run deep learning experiments on an NVIDIA Tesla V100 GPU.

4. EXPERIMENTAL RESULTS

To compare the three algorithms presented in Section 2, we perform two experiments. First, we evaluate the performance of the algorithms by computing the accuracy, training and testing times. Then, we visually analyze the resulting land cover maps.

4.1. Comparison of classification algorithms

This section compares the relative performance – accuracy and runtimes – of the three algorithms presented in Section 2. Table 2 gives the Overall Accuracy (OA) values, and the train and test times on both CPUs and GPUs. Highest OA is depicted in boldface.

Table 2. Overall Accuracy (OA) and runtime for Random Forests (RFs), Recurrent Neural Networks (RNNs), and temporal Convolutional Neural Networks (TempCNNs).

	RF	**RNN**	**TempCNN**
OA	94.0±0.9 %	90.8±2.1 %	**94.5±1.0 %**
CPU train	17 min±4 min	7h27±1h29	43 min±11 min
CPU train / epoch	-	3h06±3 min	17±4 min
CPU test	3 s	9 min	1 min
GPU train	-	1h15±16 min	18 min±5 min
GPU train / epoch	-	31 min±1 min	6 min±1 min
GPU test	-	5 min	38 s

Table 2 shows that RFs and TempCNN obtain similar accuracy results, with TempCNNs obtaining the highest OA values. It also shows that RNNs has a lower OA value of 3 % with the highest runtime complexity. As it requires 9 minutes on CPU to classify about 900k test instances, it will require almost 20 hours to obtain the land cover map for the whole tile (about 120M pixels to classify), *versus* 7 minutes for RFs and 2h15 for TempCNNs, respectively.

RF low runtime is its main advantage when dealing with large scale studies. Training time for TempCNNs is on par with RF if training on GPU (with a batch size of 64). In addition, its transfer learning capability might be a solution to the lack of accurate labeled training instances [19]. Finally, we would like to stress that the results need to be reproduced for a finer nomenclature, for example one that discriminates between types of crops (*e.g.* beans *versus* lentils). We suspect that TempCNNs might outperform RFs more significantly in that case: we previously found a difference of 3 % in accuracy for a multi-crop study [7].

[1]We provide in our repository results for both mono-directional and bidirectionall RNNs: `https://github.com/charlotte-pel/igarss2019-dl4sits`.

4.2. Visual analysis of the produced land cover maps

To ease the interpretation, we limit the visual analysis to two square areas of 6 km × 6 km (600 pixels × 600 pixels), highlighted in blue and green in Figure 2. The results for the full area are available at `https://github.com/charlotte-pel/igarss2019-dl4sits`. Figure 3 shows the results for these areas. The first column shows a Sentinel-2 image in false color from September 26 2017. The other columns display the land cover maps obtained by the three algorithms: RF, RNN and TempCNN. Legend of land cover maps is displayed in Table 1.

For the first area, Figure 3 shows disagreements mainly on grassland areas (light green) where RF and RNN seem to exhibit more noise than TempCNN. RF is better able to detect the linear forest delineations (darker green). As for the second area, Figure 3 shows disagreements between bare ground (grey) and legumes (orange). A visual inspection of the corresponding discrepancies has shown that a vegetation regrowth occurs around May 2018 for those areas where bare ground is detected by RF but not by TempCNN. Both classifiers are thus correct depending on the time. RNN also made a mistake on one of those areas by classifying it as 'urban' (pink). Finally, the presence of salt and pepper noise indicates that the three algorithms could benefit from the use of spatial information.

5. CONCLUSION

This work tackles the choice of classification algorithms for the classification of new Sentinel-2 SITS over large areas. For the first time, we compared RFs to RNNs and TempCNNs. The results show good quantitative and qualitative results for RFs and TempCNNs. Conversely, RNNs seem less successful for the given classification task due to its prohibitive time complexities (especially training time) and its lower accuracy.

Acknowledgments

We would like to thank Elizabeth Morse-McNabb, Kathryn Sheffield, Rob Clark, Hayden Lewis, and Susan Robson from Agriculture Victoria for providing us with the reference data over Victoria. We would also like to thank Olivier Hagolle and the PEPS team to have helped us with the pre-processing and the collection of Sentinel-2 images.

6. REFERENCES

[1] M Drusch, U Del Bello, S Carlier, O Colin, V Fernandez, F Gascon, B Hoersch, C Isola, P Laberinti, P Martimort, A Meygret, F Spoto, O Sy, F Marchese, and P Bargellini, "Sentinel-2: ESA's optical high-resolution mission for GMES operational services," *Remote Sensing of Environment*, vol. 120, pp. 25–36, 2012.

[2] C Gómez, J C White, and M A Wulder, "Optical remotely sensed time series data for land cover classification: A review,"

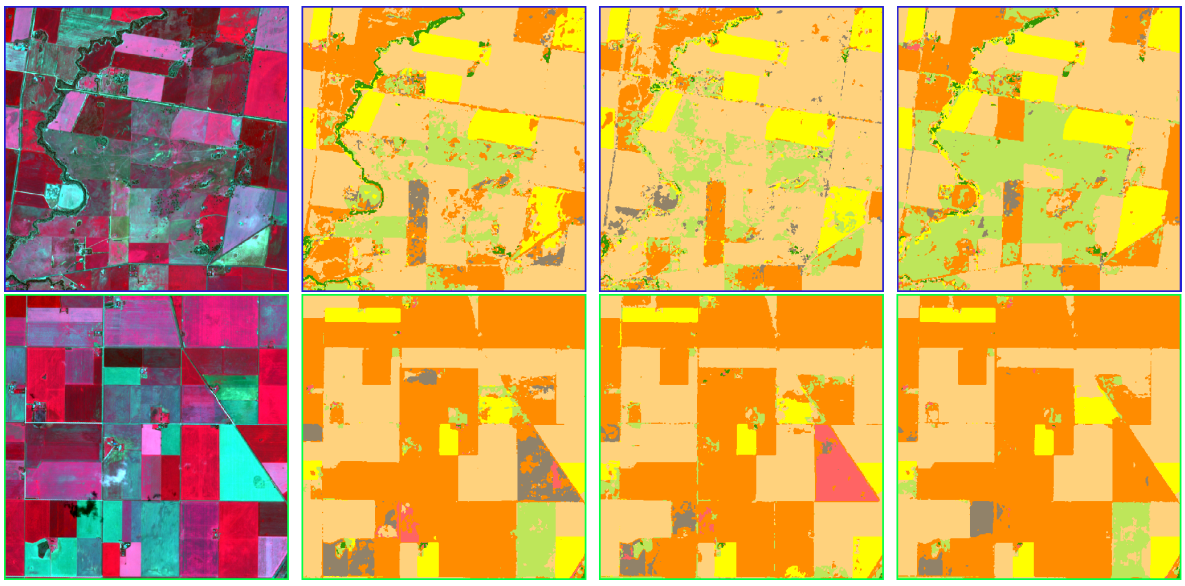

Fig. 3. Visual analysis for two areas. From left to right: Sentinel-2 image in false color from September 26 2017, Random Forest (RF) map, Recurrent Neural Network (RNN), and temporal Convolutional Neural Network (TempCNN) map.

ISPRS Journal of Photogrammetry and Remote Sensing, vol. 116, pp. 55–72, 2016.

[3] C Pelletier, S Valero, J Inglada, N Champion, and G Dedieu, "Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas," *Remote Sensing of Environment*, vol. 187, pp. 156–168, 2016.

[4] E Ndikumana, D Ho Tong Minh, N Baghdadi, D Courault, and L Hossard, "Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France," *Remote Sensing*, vol. 10, no. 8, pp. 1217, 2018.

[5] D H T Minh, D Ienco, R Gaetano, N Lalande, E Ndikumana, F Osman, and P Maurel, "Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR Sentinel-1," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 464–468, 2018.

[6] Z Sun, L Di, and H Fang, "Using Long Short-Term Memory Recurrent Neural Network in land cover classification on Landsat and cropland data layer time series," *International Journal of Remote Sensing*, pp. 1–22, 2018.

[7] C Pelletier, G I Webb, and F Petitjean, "Temporal Convolutional Neural Network for the classification of satellite image time series," *arXiv preprint arXiv:1811.10166*, 2018.

[8] M Belgiu and L Drăguţ, "Random Forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016.

[9] C Pelletier, S Valero, J Inglada, N Champion, C Marais Sicre, and G Dedieu, "Effect of training class label noise on classification performances for land cover mapping with satellite image time series," *Remote Sensing*, vol. 9, no. 2, pp. 173, 2017.

[10] J Inglada, A Vincent, M Arias, B Tardy, D Morin, and I Rodes, "Operational high resolution land cover map production at the country scale using satellite image time series," *Remote Sensing*, vol. 9, no. 1, pp. 95, 2017.

[11] L Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] D Ienco, R Gaetano, C Dupaquier, and P Maurel, "Land cover classification via multitemporal spatial data by deep Recurrent Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1685–1689, 2017.

[13] A Krizhevsky, I Sutskever, and G E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[14] Z Wang, W Yan, and T Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[15] O Hagolle, M Huc, D Villa Pascual, and G Dedieu, "A multitemporal and multi-spectral method to estimate aerosol optical thickness over land, for the atmospheric correction of formosat-2, landsat, venμs and sentinel-2 images," *Remote Sensing*, vol. 7, no. 3, pp. 2668–2691, 2015.

[16] D P Kingma and J Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2014.

[17] F Chollet et al., "Keras," 2015, `https://keras.io`.

[18] M Abadi, P Barham, J Chen, Z Chen, A Davis, J Dean, M Devin, Sa Ghemawat, G Irving, M Isard, et al., "Tensorflow: A system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[19] F Hu, G-S Xia, J Hu, and L Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.