DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn

Roberto Interdonato, Dino Ienco, Raffaele Gaetano, Kenji Ose

DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn

Roberto Interdonato*

CIRAD, UMR TETIS, Montpellier

Dino Ienco

IRSTEA, UMR TETIS
LIRMM, University of Montpellier, Montpellier

Raffaele Gaetano

CIRAD, UMR TETIS, Montpellier

Kenji Ose

IRSTEA, UMR TETIS, Montpellier

Abstract

Nowadays, modern Earth Observation systems continuously generate huge amounts of data. A notable example is represented by the Sentinel-2 mission, which provides images at high spatial resolution (up to 10m) with high temporal revisit period (every 5 days), which can be organized in Satellite Image Time Series (SITS). While the use of SITS has been proved to be beneficial in the context of Land Use/Land Cover (LULC) map generation, unfortunately, machine learning approaches commonly leveraged in remote sensing field fail to take advantage of spatio-temporal dependencies present in such data.

Recently, new generation deep learning methods allowed to significantly advance research in this field. These approaches have generally focused on a single type of neural network, i.e., Convolutional Neural Networks (CNNs) or Recur-

*Corresponding author
Email addresses: roberto.interdonato@cirad.fr (Roberto Interdonato), dino.ienco@irstea.fr (Dino Ienco), raffaele.gaetano@cirad.fr (Raffaele Gaetano), kenji.ose@irstea.fr (Kenji Ose)

rent Neural Networks (RNNs), which model different but complementary information: spatial autocorrelation (CNNs) and temporal dependencies (RNNs). In this work, we propose the first deep learning architecture for the analysis of SITS data, namely *DuPLO* (DUal view Point deep Learning architecture for time series classificatiOn), that combines Convolutional and Recurrent neural networks to exploit their complementarity. Our hypothesis is that, since CNNs and RNNs capture different aspects of the data, a combination of both models would produce a more diverse and complete representation of the information for the underlying land cover classification task. Experiments carried out on two study sites characterized by different land cover characteristics (i.e., the *Gard* site in France and the *Reunion Island* in the Indian Ocean), demonstrate the significance of our proposal.

Keywords: Satellite Image Time Series, Deep Learning, Land Cover Classification, Sentinel-2

1. Introduction

Modern Earth Observation (EO) systems produce huge volumes of data every day, involving programs that provide satellite images at high spatial resolution with high temporal revisit period. High-resolution Satellite Image Time Series (SITS) represent a practical way to organize this information, which is particularly useful for area monitoring tasks. A notable example is the Sentinel-2 mission, which is part of the Copernicus Programme, i.e., a programme developed by the European Space Agency (ESA) that involves a constellation of satellites monitoring different aspects of the Earth surface. The Sentinel-2 mission allows to monitor the entire Earth Surface at 10m of spatial resolution with a revisit period between 5 and 10 days, supplying optical information ranging from visible to near and medium infrared. One of the main advantages of this mission is that the produced data are publicly available.

For these reasons, the use of SITS is gaining increasing success in a plethora of different domains, such as ecology, agriculture, mobility, health, risk monitoring and land management planning [1, 2, 3, 4, 5, 6, 7, 8, 9]. In the context of Land Use/Land Cover (LULC) classification, exploiting SITS can be fruitful to discriminate among classes that exhibit different temporal behaviors [10], i.e., with the respect to the results that can be obtained using a single image. In [7], the authors propose to exploit SITS data to extract homogeneous land units in terms of phenological patterns and, later, for the automatic classification of land units according to their land-cover. The effectiveness of Sentinel-2 SITS to produce land cover maps at country scale has been showed in [8], demonstrating the practical interest of such data source. In [9], the authors combine multi-source optical (Landsat-8) and radar (Sentinel-1) SITS in order to improve land cover maps on the agricultural domain. Another example is supplied in [3] where optical SITS are leveraged to characterize grassland area as proxy indicator for biodiversity, food production, and global carbon cycle.

Despite the usefulness of temporal trends that can be derived from remote sensing time series, most of the strategies proposed for SITS analysis tasks [11, 12, 8, 7], directly apply standard machine learning approaches (i.e. Random Forest, SVM) on the stacked images, thus ignoring any temporal dependencies that may be discovered in the data. Indeed, such algorithms make the assumption that the information (spectral bands and timestamps) are independent from each other.

Recently, the deep learning revolution [13] has shown that neural network models are well adapted tools for automatically managing and classifying remote sensing data. The main characteristic of these models is the ability to simultaneously extract features optimized for image classification as well as the associated classifier. Moreover, unlike standard machine learning approaches, they can be used to discover spatial and temporal dependencies in SITS data. Deep learning methods can be roughly categorized in two families of techniques: convolutional neural networks [13] (CNNs) and recurrent neural networks [14] (RNNs). CNNs are well suited to model the spatial autocorrelation available in an image and they are already a well-known tool in the field of Remote Sensing [13, 9, 15]. Conversely, RNNs are specifically tailored to manage time dependencies [16] from multidimensional time series and they are recently starting to get attention in the Remote Sensing community [16, 17, 18]. Such models explicitly capture temporal correlations by recursion and they have already proved to be effective in different domains such as speech recognition [19], natural language processing [20] and image completion [21].

Considering the analysis of SITS data, few works already exist which exploit deep learning to analyze such kind of data. A CNN based strategy is employed in [9] to deal with land cover classification in the agricultural domain. The main idea is to obtain a single image by stacking all the images in an input SITS, then using it to fed a CNN-based model. The results demonstrate the quality of the proposed approach w.r.t. a standard machine learning algorithm (i.e., Random Forest classifier). In [17] a binary change detection classification task (i.e., change vs. no-change) has been addressed using a RNN model on a small time series of two dates. The authors in [16] propose a RNN-based approach for land cover classification on optical SITS data. The work evaluates deep learning methods on very high resolution and high resolution data on two different study sites. These preliminary results have paved the way to the use of such models for the analysis of Satellite image time series data. Always in the context ofh land cover classification tasks, RNN have also been used for the analysis of radar SITS data [22, 23].

Even if we acknowledge the existence of a significant body of work dealing with the use of deep learning for the analysis of SITS data, at the same time it should be observed how previous works were tied to the choice of a specific network model, i.e., focusing either on Recurrent or Convolutional Neural Networks. Even though both approaches have been shown to be effective on the analysis of SITS data, our hypothesis is that, since they capture different knowledge aspects, a combination of both would produce a more diverse and complete representation of the information. For this reason, we propose a deep learning architecture for the analysis of SITS data, namely *DuPLO* (DUal view Point deep Learning architecture for time series classificatiOn), based on the combination of Convolutional and Recurrent neural networks. To the best of our knowledge, $DuPLO$ is the first example of deep learning architecture combining RNN and CNN approaches for the analysis of satellite image time series.

The idea behind $DuPLO$ is to take advantage of the fact that the two strategies (i.e., CNN and RNN) focus on different characteristics of the data, so that addressing the problem from a dual view point will allow to exploit as much as possible the complementary information produced by such models. Experiments carried out on two study sites characterized by different land cover characteristics (i.e., the *Gard* site in France and the *Reunion Island* in the Indian Ocean), demonstrate the effectiveness of our proposal compared to state of the art approaches for the characterization of land cover mapping on SITS data. Furthermore, the quantitative and qualitative results emphasize how the combination of CNN and RNN is beneficial for the classification task compared to the use of a single neural network model.

The rest of the article is structured as follows: Section 2 introduces the proposed deep learning architecture for land cover classification from SITS data. The study sites and the associated data are presented in Section 3 while, the experimental setting and the evaluations are carried out and discussed in Section 4. Finally, Section 5 draws conclusions.

2. $DuPLO$: A DUal view Point deep Learning architecture for time series classificatiOn

Figure 1 shows a visual representation of the proposed $DuPLO$ deep learning architecture for the satellite image time series classification process. The optical time series of Sentinel-2 (S2) satellite images in input is first processed independently by the two branches consisting of two different neural networks: a CNN (cf. Section 2.1) and a RNN (cf. Section 2.2). Each branch supplies complementary information for the discriminative process since they look at the information from different points of view. The result of each branch is a feature vector summarizing the extracted knowledge. Each vector is used independently to train an auxiliary classifier (cf. Section 2.3), while the concatenation of both

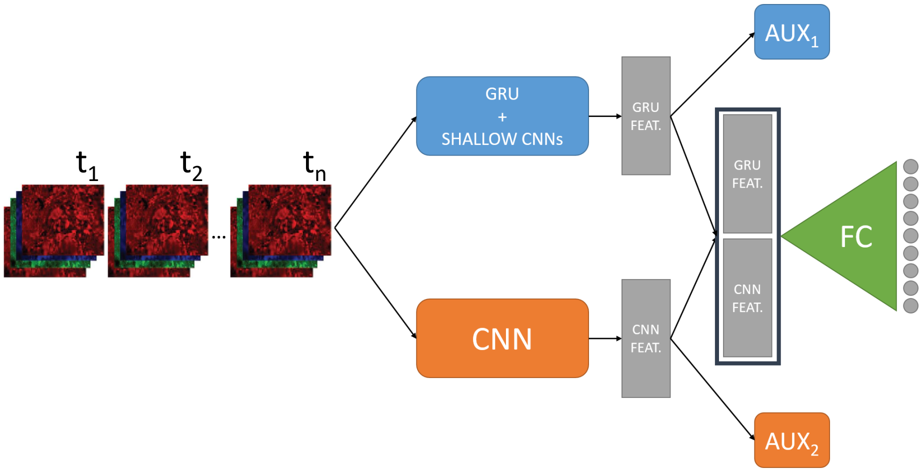


Figure 1: Visual representation of the *DuPLO* Deep Learning Architecture.

feature vectors (i.e., in a single feature vector of 2048 descriptors) is used to fed the final classifier that produces the land cover decision.

2.1. CNN Branch

Figure 2 depicts the network architecture associated to the CNN branch of *DuPLO*. For this branch, we took inspiration from the VGG model [24], a well-known network architecture usually adopted to tackle with standard Computer Vision tasks. The basic idea behind this model is to constantly increase the number of filters along the network, as long as a reasonable size of the feature maps has been reached.

Inspecting this architecture, we can observe that the first convolution has a kernel of 3×3 and it produces 256 feature maps. The second convolution still applies a kernel of size 3×3 outputting 512 feature maps while the last convolution, with kernel size 1×1, is only devoted to increase the final number of extracted features to 1024. All the convolutions are associated with a linear filter, followed by a Rectifier Linear Unit (ReLU) activation function [25] to induce non-linearity and a batch normalization step [26] that accelerates deep network training convergence by reducing the internal covariate shift.

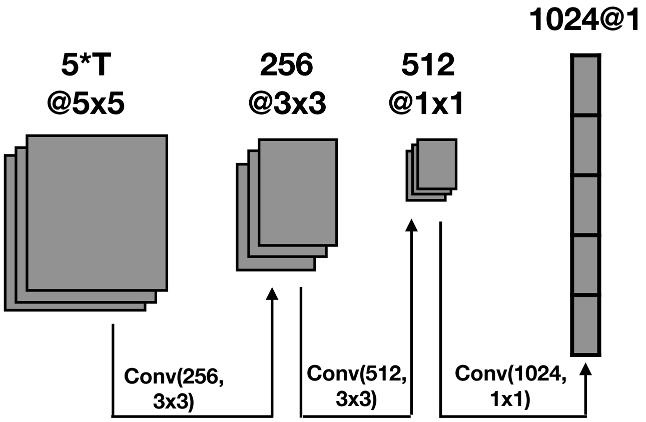


Figure 2: Visual representation of the CNN module representing the branch of the *DuPLO* model that manages the whole time series information as a stacked image. The T value indicates the number of timestamps in the satellite image time series.

The ReLU activation function is defined as follows:

$$ReLU(x) = Max(0, W \cdot x + b) \tag{1}$$

This activation function is defined on the positive part of the linear transformation of its argument ($W \cdot x + b$) where x is the input information and W and b are parameters learned by the neural network model. The choice of ReLU nonlinearities is motivated by two factors: i) the good convergence properties it guarantees and ii) the low computational complexity it provides [25].

Even though the proposed architecture shows a reasonable number of parameters, the training of such models may be difficult and the final model can suffer by overfitting [27]. In order to avoid such phenomena, following a common practice for the training of deep learning architectures, we add Dropout [27] after each batch normalization step. We set the drop rate equal to 0.4 meaning that 40% of the neurons are randomly deactivated at each propagation step, at training time.

2.2. RNN Branch

Recurrent Neural Networks are well established machine learning techniques that demonstrate their quality in different domains such as speech recognition, signal and natural language processing [28, 20]. Unlike standard feed forward networks (e.g., Convolutional Neural Networks – CNNs), RNNs explicitly manage temporal data dependencies since the output of the neuron at time t-1 is used, together with the next input, to feed the neuron itself at time t.

Recently, recurrent neural network (RNN) approaches have been successfully applied to tasks in the remote sensing field, e.g., to produce land use mappings from time series of optical images [16] and to recognize vegetation cover status using Sentinel-1 radar time series [29]. Motivated by these recent research results, we introduce a RNN module to manage information from the Sentinel-2 time series with the aim to extract an alternative representation from the data. In our model, we choose the GRU unit (Gated Recurrent Unit) introduced in [30] since it has a moderate number of parameters to learn and its effectiveness in the field of remote sensing has already been proved [16, 31]. Due to the fact that we consider patches of satellite images, centered around a central pixel, we do not use the GRU unit directly on the radiometric information. First, we use a shallow CNN, we name $SCNN$, to process the patches at each timestamp and, subsequently, we feed the RNN model with the information extracted by the convolutional models. Finally, we couple the Gated Recurrent Unit with an *attention* mechanism [32].

The structure of the $SCNN$is reported in Figure 3. This shallow network is composed only by two convolutional layers, with 32 and 64 filters respectively, producing an output vector composed by 64 features. This step allows to extract the information carried out by the spatial neighborhood of the considered pixel before considering the temporal behavior of the satellite image time series. Also in this case, each convolution is associated with a linear filter, followed by a Rectifier Linear Unit (ReLU) activation function [25] to induce non-linearity and a batch normalization step [26].

Once the $SCNN$is applied on the patches describing the time series, the

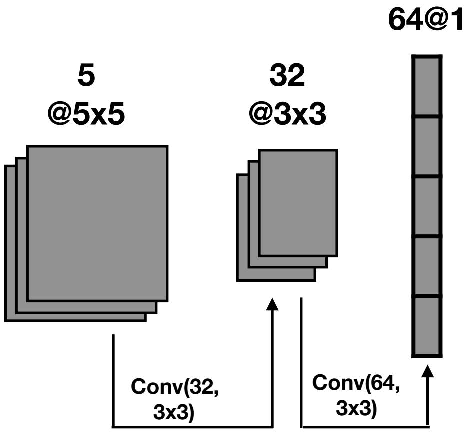


Figure 3: Visual representation of the Shallow CNN applied at each timestamp of the satellite image time series. The results of this non-linear transformation is successively used to feed the Gate Recurrent Unit.

input of a RNN unit is a sequence (x_{t_1},..., x_{t_T}) where a generic element x_{t_i} is a feature vector of cardinality equals to 64 (extracted by $SCNN$) and t_i refers to the corresponding time stamp.

Equations 2, 3 and 4 formally describe the GRU neuron.

$$z_{t_i} = \sigma(W_{zx}x_{t_i} + W_{zh}h_{t_{i-1}} + b_z) \tag{2}$$

$$r_{t_i} = \sigma(W_{rx}x_{t_i} + W_{rh}h_{t_{i-1}} + b_r) \tag{3}$$

$$\begin{aligned} h_{t_i} = {} & z_t \odot h_{t-1} + \\ & (1 - z_{t_i}) \odot \tanh(W_{hx}x_t + W_{hr}(r_t \odot h_{t_{i-1}}) + b_h) \end{aligned} \tag{4}$$

The $\odot$ symbol indicates an element-wise multiplication while σ and tanh represent Sigmoid and Hyperbolic Tangent function, respectively.

The GRU unit has two gates, update (z_t) and reset (r_t), and one cell state, i.e., the hidden state (h_t). Moreover, the two gates combine the current input (x_t) with the information coming from the previous time stamp (h_{t-1}). The update gate effectively controls the trade off between how much information from the previous hidden state will carry over to the current hidden state and how much information of the current time stamp needs to be kept. On the other hand, the reset gate monitors how much information from the previous timestamps needs to be integrated with current information. As all hidden units have separated reset and update gates, they are able to capture dependencies over different time scales. Units more prone to capture short-term dependencies will tend to have a frequently activated reset gate, but those that capture longer-term dependencies will have update gates that remain mostly active [30]. This behavior enables the GRU unit to remember long-term information.

Attention mechanisms [32] are widely used in automatic signal processing (1D signal or language) and they allow to join together the information extracted by the GRU model at different time stamps. The output returned by the GRU model is a sequence of learned feature vectors for each time stamp: (h_{t_1},..., h_{t_N}) where each h_{t_i} has the same dimension d. Their matrix representation $H \in \mathbb{R}^{T,d}$ is obtained vertically stacking the set of vectors. The attention mechanism allows us to combine together these different vectors h_{t_i}, in a single one rnn_{feat}, to attentively combine the information returned by the GRU unit at each of the different time stamps. The attention formulation we use, starting from a sequence of vectors encoding the learned descriptors (h_{t_1},..., h_{t_T}), is the following one:

$$v_a = tanh(H \cdot W_a + b_a) \tag{5}$$

$$\omega = SoftMax(v_a \cdot u_a) \tag{6}$$

$$rnn_{feat} = \sum_{i=1}^{T} \lambda_i \cdot h_{t_i} \tag{7}$$

where matrix $W_a \in \mathbb{R}^{d,d}$ and vectors $b_a, u_a \in \mathbb{R}^d$ are parameters learned during the process. These parameters allow to combine the vectors contained in matrix H. The purpose of this procedure is to learn a set of weights (ω_{t_1},..., ω_{t_T}) that allows the contribution of each time stamp to be weighted by h_{t_i} through a linear combination. The $SoftMax(\cdot)$ [16] function is used to normalize weights ω so that their sum is equal to 1. The output of the RNN module is the feature vector rnn_{feat}: it encodes temporal information related to ts_i for the pixel i.

With the aim to supply an equal amount of information from each branch of analysis of the *DuPLO* model, we set the value of d (the size of the learned feature vector) equal to 1024.

The features extracted by each branch of the architecture are combined by concatenation, resulting in a feature vector of 2048 descriptors, i.e., 1024 from the CNN branch and 1024 from the RNN branch. In this way we assure an equal contribution from each branch to the final decision. Many other combination techniques are possible (e.g., sum, gating, and so on) but we rely on standard concatenation following recent practices introduced by working on multi-source remote sensing analysis [18, 33].

2.3. Training of DuPLO model

One of the advantages of deep learning approaches, compared to standard machine learning methods, is the ability to link, in a single pipeline, the feature extraction step as well as the associated classifier [13]. This ability is particularly important when different flows of information need to be combined together, such as in our scenario where we need to couple different representations of the same data source. In addition, the different features learned via multiple non-linear combination of the radiometric information are optimized for the specific task at hand, i.e., land cover mapping.

To further strengthen the complementarity as well as the discriminative power of the learned features for each branch, we adapt the technique proposed in [34] to our problem. In [34], the authors propose to learn two complementary representations (using two convolutional networks) from the same image. The discriminative power is enhanced by two auxiliary classifiers, linked to each group of features, in addition to the classifier that uses the merged information. The complementarity is enforced by alternating the optimization of the parameters of the two branches. In our case, we still have a unique source of information (an optical Sentinel-2 time series of satellite images) but we manage it via two processing branches that differ from each other regarding the particular deep learning strategy we employ.

In detail, the classifier that exploits the full set of features is fed by concatenating the output features of both CNN (cnn_{feat}) and RNN (rnn_{feat}) modules together. Empirically, we have observed that the RNN module overfits the data. To alleviate this problem, we add a Dropout layer [27] on rnn_{feat} with a drop-rate equals to 0.4. The learning process involves the optimization of three classifiers at the same time, one specific to rnn_{feat}, a second one related to cnn_{feat} and the third one that considers $[rnn_{feat}, cnn_{feat}]$.

The cost function associated to our model is :

$$\begin{aligned} L_{total} &= \alpha_1 * L_1(rnn_{feat}) + \\ &= \alpha_2 * L_2(cnn_{feat}) + \\ &= L_{fus}([cnn_{feat}, rnn_{feat}]) \end{aligned} \tag{8}$$

where

$L_i(feat)$ is the loss function associated to the classifier inputed with the features $feat$. In our case, for all the classifiers (the auxiliary and the main ones) we adopt two fully connected layers of 1024 neurons with ReLU activation function plus a final output layer with as many neurons as the number of land cover classes to predict. The SoftMax activation function is finally applied [13] on the output layer with the aim to produce a kind of probability distribution over the class labels.

Each cost function is modeled through categorical cross entropy, a typical choice for multi-class supervised classification tasks [16].

L_{total} is optimized end-to-end. Once the network has been trained, the prediction is carried out only by means of the classifier involving the concatenated features $[cnn_{feat}, rnn_{feat}]$. The cost functions L_1 et L_2, as highlighted in [34], operate a kind of regularization that forces, within the network, the features extracted to be discriminative independently.

3. Data

The analysis was carried out on the *Reunion Island*, a French overseas department located in the Indian Ocean and on part of the *Gard* department in the South of France. The *Reunion Island* dataset consists of a time series of 34 Sentinel-2 (S2) images acquired between April 2016 and May 2017 while the *Gard* dataset consists of a time series of 37 S2 images acquired between December 2015 and January 2017. All the S2 images used are those provided at level 2A by the THEIA pole [1]. We only use bands at 10m Blue, Green, Red and Near Infrared (resp. B2, B3, B4 and B8). A preprocessing was performed to fill cloudy observations through a linear multi-temporal interpolation over each band (cf. *Temporal Gapfilling* [8]), and the NDVI radiometric indice was calculated for each date [8, 36]. A total of 5 variables (4 surface reflectances plus 1 indice) is considered for each pixel of each image in the time series. The spatial extent of the *Reunion Island* site is 6 656 × 5 913 pixels while the extent for the *Gard* site is 4 822 × 6 748 pixels. Considering the former benchmark, the field database was built from various sources: (i) the *Registre parcellaire graphique* (RPG)[2] reference data of 2014, (ii) GPS records from June 2017 and (iii) photo interpretation of the VHSR image conducted by an expert, with knowledge of the territory, for distinguishing between natural and urban spaces. Also for the latter benchmark, the field database was built from various sources: (i) the *Registre parcellaire graphique* (RPG)[2] reference data of 2016 and (ii) photo interpretation of the VHSR image conducted by an expert, with knowledge of the territory, for distinguishing between natural and urban areas.

[1] Data are available via `http://theia.cnes.fr`, preprocessed in surface reflectance via the *MACCS-ATCOR Joint Algorithm* [35] developed by the National Centre for Space Studies (CNES).

[2] RPG is part of the European Land Parcel Identification System (LPIS), provided by the French Agency for services and payment

Class	Label	# **Objects**	# **Pixels**
1	*Crop Cultivations*	380	12090
2	*Sugar cane*	496	84136
3	*Orchards*	299	15477
4	*Forest plantations*	67	9783
5	*Meadow*	257	50596
6	*Forest*	292	55108
7	*Shrubby savannah*	371	20287
8	*Herbaceous savannah*	78	5978
9	*Bare rocks*	107	18659
10	*Urbanized areas*	125	36178
11	*Greenhouse crops*	50	1877
12	*Water Surfaces*	96	7349
13	*Shadows*	38	5230

Table 1: Characteristics of the *Reunion Island* Dataset

3.1. Ground Truth Statistics

Considering both datasets, ground truth comes in GIS vector file format containing a collection of polygons each attributed with a unique land cover class label. To ensure a precise spatial matching with image data, all geometries have been suitably corrected by hand using the corresponding Sentinel-2 images as reference. Successively, the GIS vector file containing the polygon information has been converted in raster format at the Sentinel-2 spatial resolution (10m).

The final ground truths are constituted of 322 748 pixels (resp. 2 656 objects) distributed over 13 classes for the *Reunion Island* dataset (Table 1) and 1 157 260 pixels (resp. 2 538 objects) distributed over 8 classes for the *Gard* benchmark (Table 2). We remind that the ground truth, in both cases, was collected over large areas.

Class	Label	**# Objects**	**# Pixels**
1	*Corn Crop*	260	115264
2	*Barley Crop*	76	12279
3	*Other Crops*	430	127236
4	*Rice*	208	124943
5	*Orchards*	242	26142
6	*Olive Trees*	203	9367
7	*Meadow*	230	59018
8	*Vineyard*	259	76815
9	*Forest*	171	73915
10	*Urbanized areas*	266	10584
11	*Water Surfaces*	193	521873

Table 2: Characteristics of the Gard Dataset

4. Experiments

In this section, we present and discuss the experimental results obtained on the study sites introduced in Section 3. We carried out several experimental stages, in order to provide a complete analysis of the behavior of $DuPLO$:

- we provide an ablation study in which we evaluate the importance of the different components of $DuPLO$ (Section 4.2);
- we perform an extensive quantitative analysis, comparing the global classification performances and the per-class results obtained by $DuPLO$ w.r.t. competing methods and baseline approaches (Section 4.3);
- we provide a qualitative discussion considering the land cover maps produced by our framework compared to those produced by competing methods (Section 4.4).

4.1. Experimental Settings

For our analysis, we selected as competing methods two state of the art techniques commonly employed for the classification of SITS. As state of the art standard machine learning tool, we selected a Random Forest (RF) classifier [8, 36], while as regards deep learning techniques we selected the Recurrent Neural Network approach ($LSTM$) recently proposed in [16], since it demonstrates interesting performances in the classification of SITS. Furthermore, similarly to what proposed in [16], we also investigate the possibility to use the deep learning architecture to obtain a new data representation for the classification task. To this end, we feed a Random Forest classifier with the features extracted by $DuPLO$, naming this approach RF(DuPLO).

All the Deep Learning methods (including $DuPLO$) are implemented using the Python Tensorflow library. During the learning phase, considering both $DuPLO$ and $LSTM$, we use the Adam method [37] to learn the model parameters with a learning rate equal to $2 \cdot 10^{-4}$. The training process is conducted over 300 epochs with a batch size equals to 128. The number of hidden units for the RNN module is fixed to 1 024.

As concerns $DuPLO$, we perform a preprocessing phase in order to associate each pixel to its surrounding area (i.e., to force the learning process to take into account the spatial context). We consider patches with a spatial extent equals to 5 × 5, where each patch represents the spatial context of the pixel in position (2,2). This means that for each timestamp we have a cube of information of size (5 × 5 × 5), since 5 is the number of raw bands and indices involved in the analysis. Finally, all the patches are associated to a land cover label that corresponds to the label of the central pixel. The values are normalized, per band (resp. indices) considering the time series, in the interval $[0, 1]$. Regarding the $LSTM$ and RF approaches, according to standard literature [16], the input is represented by the time series information associated to each pixel.

We divide the dataset into three parts: training, validation and test set. Training data are used to learn the model while validation data are exploited for model selection varying the parameters of each method. Finally, the model that achieves the best accuracy on the validation set is successively employed to perform the classification on the test set. More in detail, we use 30% of the objects for the training phase, 20% of the objects for the validation set while the remaining 50% are employed for the test phase. We impose that all the pixels of the same object belong exclusively to one of the splits (training, validation or test) to avoid spatial bias in the evaluation procedure [8]. Considering the *RF* models, we optimize the model via two parameters: the maximum depth of each tree and the number of trees in the forest. For the former parameter, we vary it in the range {20,40,60,80,100} while for the latter one we takes values in the set {100, 200, 300,400,500}.

Experiments are carried out on a workstation with an Intel (R) Xeon (R) CPU E5-2667 v4@3.20Ghz with 256 GB of RAM and four TITAN X GPU. The assessment of the classification performances is done considering global precision (*Accuracy*), *F-Measure* [16] and *Kappa* measures.

It is known that, depending on the split of the data, the performances of the different methods may vary as simpler or more difficult examples are involved in the training or test set. To alleviate this issue, for each dataset and for each evaluation metric, we report results averaged over ten different random splits performed with the previously presented strategy.

4.2. Ablation Analysis

In this set of experiments we investigate the interplay among the different components of $DuPLO$, setting up an ablation analysis in which we disentangle the benefits of the different parts of our framework. To this end, we compare $DuPLO$ with three variants of the original model: i) excluding the use of the auxiliary classifiers ($DuPLO_{noAux}$), ii) considering only the Convolutional Neural Network branch (*Cbranch*) and iii) considering only the Recurrent Neural Network branch (*Rbranch*). The results are reported in Table 3 (resp. Table 4) for the *Reunion Island* (reps. *Gard*) study site. In both cases we can note that $DuPLO$ outperforms the other variants. This fact underlines that: all the different components play a role in the classification process and support our intuition that combining different models would produce a more diverse and complete representation of the information. The second finding we can underline is that *Cbranch* and *Rbranch* behave similarly, showing no real difference in terms of obtained *Accuracy*, *F-Measure* and *Kappa* measure. Even tough the two variants transform the original information differently, the extracted multifaceted knowledge lets them achieve similar performances. Finally, the use of auxiliary classifiers to boost the discriminative power of each branch independently is confirmed to be effective. It can be observed how for both datasets $DuPLO$ outperforms the variant without auxiliary classifiers ($DuPLO_{noAux}$) as well as the variants involving only one branch of the proposed architecture (i.e., *Cbranch* and *Rbranch*).

	Accuracy	*F-Measure*	*Kappa*
$DuPLO$	83.72% ± 1.08%	83.73% ± 1.03%	0.8089 ± 0.0122
$DuPLO_{noAux}$	80.28% ± 0.68%	80.25% ± 0.68%	0.7685 ± 0.0075
Cbranch	79.50% ± 1.00%	79.48% ± 1.11%	0.7594 ± 0.0118
Rbranch	79.11% ± 1.66%	78.97% ± 1.71%	0.7547 ± 0.0191

Table 3: Accuracy, F-Measure, Kappa considering different ablation of $DuPLO$ on the *Reunion* dataset

	Accuracy	*F-Measure*	*Kappa*
$DuPLO$	96.36% ± 0.52%	96.28% ± 0.55%	0.9513 ± 0.0067
$DuPLO_{noAux}$	95.86% ± 0.34%	95.78% ± 0.33%	0.9446 ± 0.0041
Cbranch	96.01% ± 0.42%	95.91% ± 0.40%	0.9466 ± 0.0056
Rbranch	96.02% ± 0.43%	95.91% ± 0.40%	0.9466 ± 0.0056

Table 4: Accuracy, F-Measure, Kappa considering different ablation of $DuPLO$ on the *Gard* dataset

4.3. Comparative Analysis

Table 5 and Table 6 report the results obtained by *RF*, $LSTM$, $DuPLO$ and *RF*(DuPLO) on the *Reunion Island* and *Gard* study sites, respectively. We can observe how on both benchmarks $DuPLO$ outperforms both state of the art competing methods. However, we can also note that using $DuPLO$ as feature extractor (RF(DuPLO)) provides always the best average performances in terms of *Accuracy*, *F-Measure* and *Kappa* measure. This is in line with recent work in remote sensing [29, 38] and it is due to the fact that, once the newly extracted features are optimized for a particular task, classical machine learning techniques are able to efficiently leverage the information richness carried out by such features.

	Accuracy	*F-Measure*	*Kappa*
RF	82.99% ± 1.04%	82.40% ± 1.09%	0.7989 ± 0.0119
LSTM	76.66% ± 1.21%	76.57% ± 1.11%	0.7260 ± 0.0140
DuPLO	83.72% ± 1.08%	83.73% ± 1.03%	0.8089 ± 0.0122
RF(DuPLO)	**86.12**% ± 1.21%	**86.00**% ± 1.24%	**0.8366** ± 0.0143

Table 5: REUNION

	Accuracy	*F-Measure*	*Kappa*
RF	96.04% ± 0.40%	95.71% ± 0.44%	0.9469 ± 0.0046
LSTM	95.05% ± 0.55%	94.81% ± 0.59%	0.9338 ± 0.0066
DuPLO	96.36% ± 0.52%	96.28% ± 0.55%	0.9513 ± 0.0067
RF(DuPLO)	**96.78**% ± 0.50%	**96.70**% ± 0.51%	**0.9569** ± 0.0061

Table 6: GARD

4.3.1. Per-Class Analysis on the Reunion Island benchmark

Table 7 and Table 8 summarize the per class *F-Measure* performances of the different methods for the *Reunion Island* and *Gard* study site, respectively.

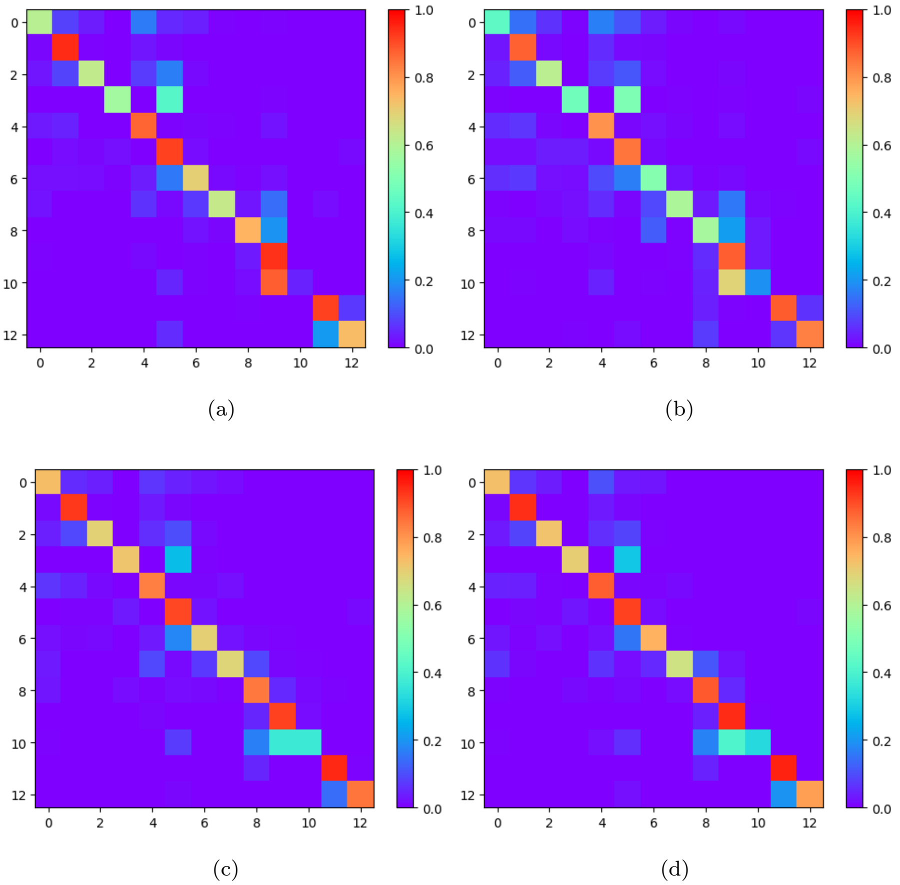
Considering the *Reunion Island* benchmark (Table 7), we can observe that, considering the main competing approaches (RF, $LSTM$ and $DuPLO$), our framework supplies the best classification results on nine over thirteen land cover classes. These classes are: (0), (1), (2), (3), (8), (9), (10), (11) and (12) (resp. *Crop Cultivations*, *Sugar cane*, *Orchards*, *Forest plantations*, *Herbaceous Savannah*, *Bare rocks*, *Urbanized areas*, *Greenhouse crops*, *Water surfaces* and *Shadows*). The highest gains are obtained in correspondence of *Bare rocks*, *Urbanized areas* and *Greenhouse crops* classes with an improvement of 8, 9 and 20 point of *F-Measure*, respectively. Considering the characteristics of such classes, the significant gains obtained by $DuPLO$ are the results of the effectiveness of our approach to exploit the temporal behavior supplied by the time series as well as the fact that our approach integrates a small amount of spatial context via the patch (a 5×5 image grid) centered around the analyzed pixel. Regarding all the other land cover classes (*Forest Plantations*, *Meadow*, *Forest* and *Shrubby savannah*) the performances of $DuPLO$ are comparable with the results obtained by the other approaches. Looking at the RF(DuPLO) method, we can observe that this solution provides the best performances over all the land cover categories and, in particular, we can note that the RF algorithm largely benefits from the feature extracted by the deep learning architecture increasing its average results of more than 3 points. We also investigate the confusion between each pair of classes and we report, for each competing method, the confusion matrix in Figure 4. The visual results support the previous discussion, since a closer look at the heat maps representing the confusion matrix points out that $DuPLO$ and RF(DuPLO) are more precise than the competitors. This consideration emerges from the fact that the corresponding heat maps (Figure 4c and Figure 4d) have a more visible diagonal structure (the dark red blocks concentrated on the diagonal). This is not the case for Random Forest (Figure 4a) and $LSTM$ (Figure 4b) where the distinction between different classes is less sharp.

Method	0	1	2	3	4	5	6	7	8	9	10	11	12
RF	61.67%	91.94%	70.12%	65.63%	83.10%	85.91%	73.23%	67.47%	73.96%	82.98%	10.87%	92.53%	88.40%
LSTM	42.68%	88.20%	64.20%	53.56%	76.51%	79.51%	59.01%	60.53%	70.86%	81.61%	18.23%	92.16%	86.55%
DuPLO	62.36%	92.09%	73.24%	70.40%	82.88%	84.59%	70.29%	63.40%	82.02%	90.47%	40.31%	93.26%	**90.76%**
RF(DuPLO)	**65.72%**	**92.98%**	**75.39%**	**73.22%**	**85.40%**	**87.30%**	**75.76%**	**67.97%**	**86.32%**	**92.05%**	**43.88%**	**93.87%**	90.29%

Table 7: Average *F-Measure* per class for the *Reunion Island* Dataset

4.3.2. Per-Class Analysis on the Gard benchmark

Considering the *Gard* benchmark (Table 8), we can observe that all the main competing methods achieve similar performances on all the land cover classes with the exception of the *Barley Crop* land cover category. While the results on all other land cover classes are satisfactory regarding all the approaches, con-



(a)

(b)

(c)

(d)

Figure 4: Heat Maps representing the confusion matrices of: (a) *RF*, (b) *LSTM*, (c) *DuPLO* and (d) *RF*(DuPLO) on the *Reunion Island* study site.

versely, on this class we can observe a clear different behavior between *DuPLO* and the other algorithms. *RF* and *LSTM* have a poor behavior on *Barley Crop* with a maximum (average) *F-Measure* of 31.55%. On the other hand, we can underline that our strategy is capable to reach an *F-Measure* of 59.30% with an increase of more than 27 percentage points compared to the best state of the art method. Inspecting deeply the results, we have seen that confusion arises between the *Corn Crop* and *Barley Crop* classes. This is due to a very similar behavior between such classes, which makes it difficult to discriminate between them. The ability of *DuPLO* to disentangle and characterize the temporal behavior of each class efficiently supports the classification of such classes. We can also observe that the *RF*(DuPLO) strategy provides the best performances over all the land cover categories with the exception of the *Forest* class, where its performance is similar to the one of the best approach, namely *DuPLO* (97.60% vs 97.71%).

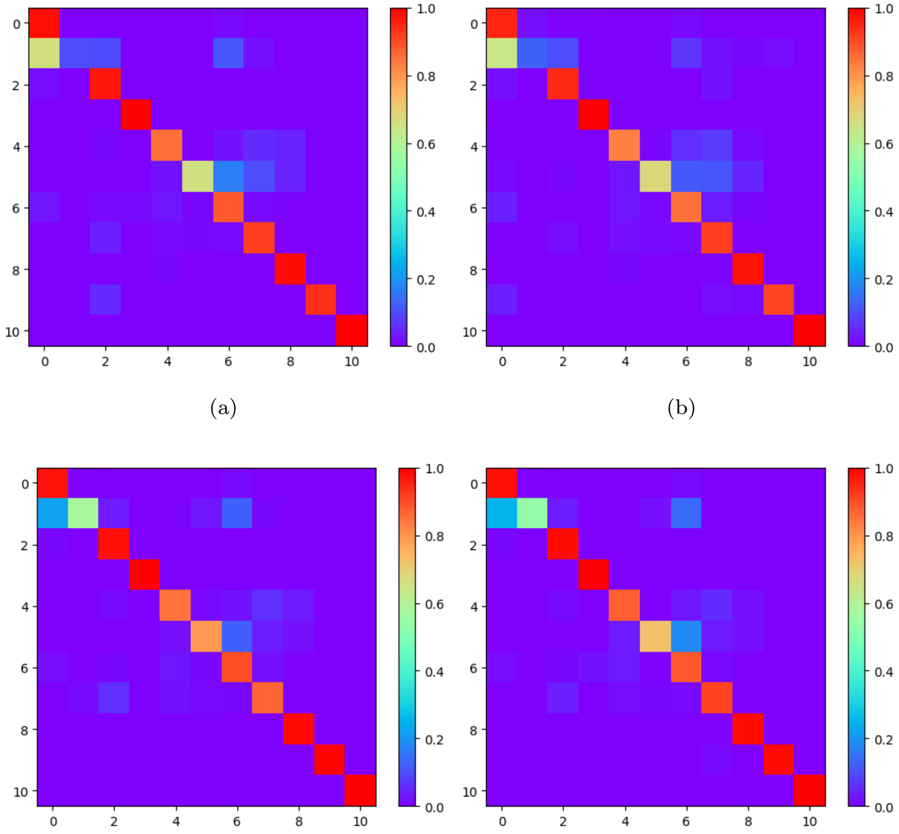
Figure 5 shows the heat maps representing the confusion matrices on the *Gard* study site. Also in this case, the visual results support the previous discussion. The heat maps representing the confusion matrices pinpoint that *DuPLO* and *RF*(DuPLO) avoid confusion on some particular classes on which the competitors fail. More specifically, we can observe this phenomenon in two different scenarios: the first one is related to the confusion between the *Corn Crop* and the *Barley Crop* land cover classes while the second case involves the *Olive Trees* class. In the former case, *DuPLO* and *RF*(DuPLO) are more precise on the *Barley Crop* while in the latter case, our approaches make some confusion between the *Olive Trees* and the *Meadow*, while the other approaches confuse the *Olive Trees* class not only with *Meadow*, but also with *Vineyard* and *Forest*.

Method	0	1	2	3	4	5	6	7	8	9	10
RF	93.40%	31.55%	96.33%	98.69%	81.19%	72.70%	82.63%	92.53%	96.74%	97.15%	99.80%
LSTM	91.30%	27.58%	95.81%	98.48%	75.37%	69.93%	78.95%	89.48%	96.72%	93.60%	99.82%
DuPLO	94.92%	59.30%	96.90%	98.99%	80.47%	70.93%	83.28%	91.69%	**97.71%**	96.77%	99.81%
RF(DuPLO)	**95.26%**	**62.65%**	**97.43%**	**99.10%**	**82.18%**	**74.64%**	**84.66%**	**93.39%**	97.60%	**98.63%**	**99.86%**

Table 8: Average *F-Measure* per class for the Gard Dataset

4.4. Qualitative Inspection of Land Cover Maps

In Figure 6 and 7 we report some representative map classification details on the *Gard* and *Reunion Island* datasets considering the *RF*, *LSTM* and *DuPLO*, respectively. With the purpose to supply a reference image with natural colors for the map classification details, we have used the multispectral information obtained from a Very High Spatial Resolution image acquired on the same area



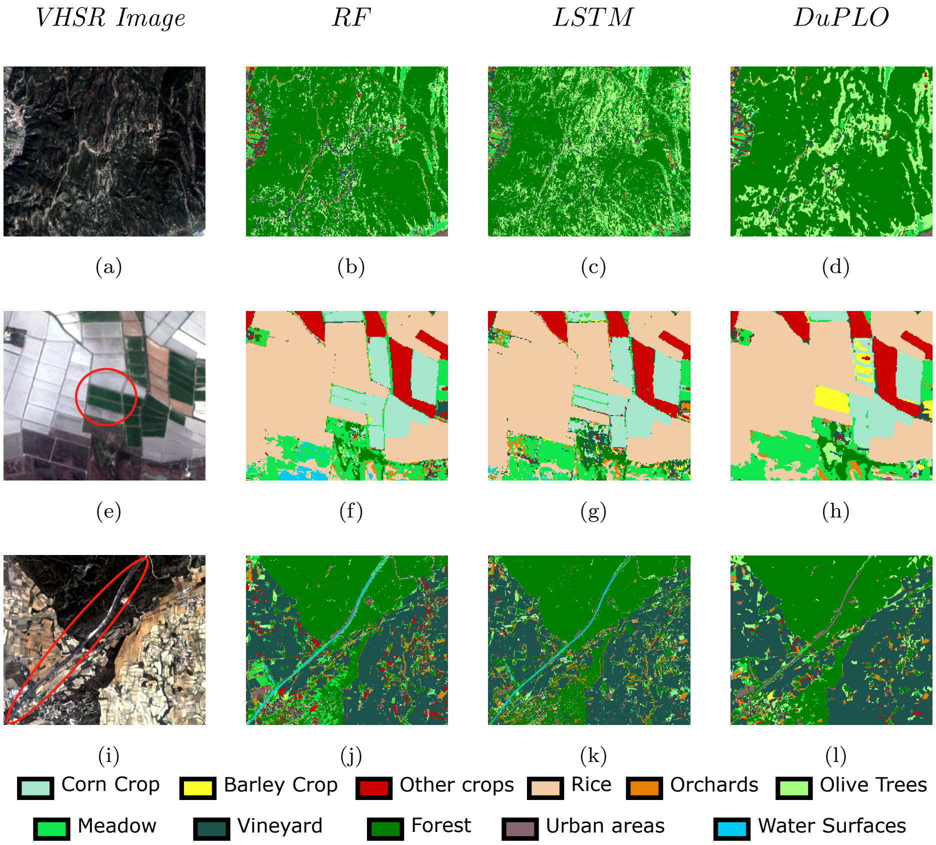
(a) (b)

(c) (d)

Figure 5: Heat Maps representing the confusion matrices of: (a) *RF*, (b) *LSTM*, (c) *DuPLO* and (d) *RF*(DuPLO) on the *Gard* study site.

in the interval spanned by the time series. More in detail, for each study site, we used the multispectral bands of a SPOT7 image at a spatial resolution of 6m.

Regarding the *Gard* study site, the first example (Figures 6a, 6b, 6c and 6d) depicts an area mainly characterized by forest, meadows and olive tress. On this area, we can observe that both *RF* and *LSTM* present confusion between these three classes and do not preserve the geometry of the scene. This is underlined by the salt and pepper error presents in their land cover maps. Conversely, we



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 6: Qualitative investigation of Land Cover Map details produced on the *Gard* study site by*RF*, *LSTM* and *DuPLO* on three different zones (from the top to the bottom): i) mixed area (forest, meadows and olive trees); ii) rural area and iii) mixed area (urban, rural area and forest).

can observe that *DuPLO* supplies a sharper (i.e., more homogeneous) characterization of the three land cover classes, especially concerning the forest class.

The second example (Figures 6e, 6f, 6g and 6h) represents a rural area principally characterized by different crop types. In this case we highlight a barley crop, in order to confirm the strong improvement in performance of *DuPLO* upon *RF* and *LSTM* on this class, already emerged from the quantitative analysis (cf. Table 8). It is evident how both *RF* and *LSTM* fail to correctly classifying the *Barley Crop*, mistakenly identifying it as *Corn Crop* (interleaved by some *Meadow*). Conversely, it can be seen how *DuPLO* correctly identifies the correct extent of the *Barley Crop*.

The third example (Figures 6i, 6j, 6k and 6l) involves an urban area, mixed with rural areas and forest. We highlight a large asphalted street which diagonally cuts the selected area. It can be noted how both *RF* and *LSTM* classify the street as water while, *DuPLO* correctly classifies it as *Urban Areas.* We remind that in our nomenclature we do not have a land cover class for street and, for this reason, detect a street as *Urban Areas* is a reasonable compromise in our scenario.

Summarizing, on this study area, the qualitative analysis of the land cover maps demonstrates the effectiveness of *DuPLO* compared to the other approaches on some specific classes like *Barley Crop* and *Corn Crop*, confirming the quantitative results reported in Table 8. The analysis also shows how *DuPLO* provides sharper and spatially coherent classification in mixed areas with respect to competitors which provide land cover maps affected by evident salt and pepper errors.

As concerns the *Reunion Island* study site, the first example (Figures 7a, 7b, 7c and 7d) depicts a forest area. It can be noted how both *RF* and *LSTM*, similarly to what observed for the first *Gard* example (Figures 6a– 6d), are not able to preserve the geometry of the scene, erroneously placing sugar cane and orchards areas among the forest ones. Conversely, *DuPLO* confirms its ability to produce a sharper and homogeneous demarcation of the forest.

The second example (Figures 7e, 7f, 7g and 7h) shows a coastal area, where we highlight two urban settlements. The salt and pepper error that has been found to characterize the land cover maps produced by *RF* and *LSTM* can be clearly observed, with spots of meadow, orchards, greenhouse crops and sugar cane in the middle of urban areas, while the land cover map produced by *DuPLO* depicts far more clear urban areas.

In the third example (Figures 7i, 7j, 7k and 7l) a bare rocks area can be observed, produced by a lava flow of the *Piton de la Fournaise* volcano on the eastern side of the island. It can be seen how *RF* and *LSTM* place urban areas

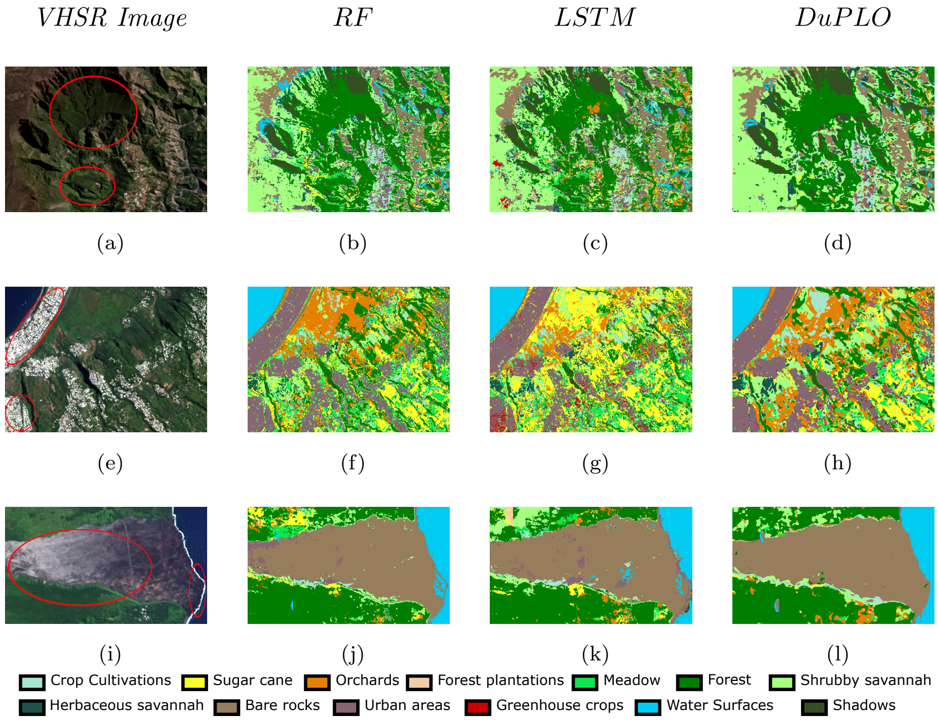


Figure 7: Qualitative investigation of Land Cover Map details produced on the *Reunion Island* study site by*RF*, *LSTM* and *DuPLO* on three different zones (from the top to the bottom): i) forest area; ii) urban area and iii) bare rocks.

and water in the middle of the bare rocks while, *DuPLO* correctly identifies all the rocky area. It is interesting to observe how *DuPLO* correctly identifies the border between the ocean and the mainland, while both competitors fail.

To sum up, the qualitative inspection of the land cover maps produced for the *Reunion Island* study site confirms the quantitative results discussed in Section 4.3 and, consolidates the observations drawn when discussing the land cover maps produced for *Gard*, especially for what concerns the ability of *DuPLO* to produce sharper and more spatially coherent land cover maps with respect to both competitors.

In order to have a wider example of how the land cover classification produced by *DuPLO* differs from the ones provided by competing methods, the land cover maps produced by *DuPLO*, *RF* and *LSTM* on the *Reunion Island* study site can be explored on our website [3].

5. Conclusion

In this paper, a novel Deep Learning architecture to deal with optical Satellite Image Time Series classification has been proposed. The approach, named *DuPLO*, leverages complementary representation of the remote sensing data to obtain a set of descriptors able to well discriminate the different land cover classes. The two-branch architecture involves a CNN and a RNN branch that process the same stream of information and, due to the difference in their structures, produces a more diverse and complete representation of the data. The final land cover classification is achieved by concatenating the features extracted by each branch. The framework is learned end-to-end from scratch.

The evaluation on two real-world study sites has shown that *DuPLO* achieves better quantitative and qualitative results than state of the art classification methods for optical SITS data. In addition, the visual inspection of the land cover maps has advocated the effectiveness of our strategy.

As future work, we plan to extend the proposed approach towards the integration of other type of remote sensing data considering a multi-source scenario. For instance, our Deep Learning strategy can be extended to combine optical and radar SITS (i.e. Sentinel-2 and Sentinel-1 data) for land cover classification.

6. Acknowledgements

This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004 (DigitAg), the GEOSUD project with reference ANR-10-EQPX-20, the Programme

[3] `https://193.48.189.134/index.php/view/map/?repository=duplo&project=duplo`
Be aware that predictions are not available on some areas of the map (i.e. on the right side of the volcano) where there were not enough data available to perform the temporal gapfilling preprocessing on the SITS due to cloud issues (cf. Section 3).

National de Télédétection Spatiale (PNTS, `http://www.insu.cnrs.fr/pnts`), grant n^{o}PNTS-2018-5, as well as from the financial contribution from the French Ministry of agriculture "Agricultural and Rural Development" trust account.

References

[1] A. Bégué, D. Arvor, B. Bellón, J. Betbeder, D. de Abelleyra, R. P. D. Ferraz, V. Lebourgeois, C. Lelong, M. Simões, S. R. Verón, Remote sensing and cropping practices: A review, Remote Sensing 10 (1) (2018) 99.

[2] S. Olen, B. Bookhagen, Mapping damage-affected areas after natural hazard events using sentinel-1 coherence time series, Remote Sensing 10 (8) (2018) 1272.

[3] N. Kolecka, C. Ginzler, R. Pazur, B. Price, P. H. Verburg, Regional scale mapping of grassland mowing frequency with sentinel-2 time series, Remote Sensing 10 (8) (2018) 1221.

[4] L. Chen, Z. Jin, R. Michishita, J. Cai, T. Yue, B. Chen, B. Xu, Dynamic monitoring of wetland cover changes using time-series remote sensing imagery, Ecological Informatics 24 (2014) 17–26. `doi:10.1016/j.ecoinf.2014.06.007`.
URL `https://doi.org/10.1016/j.ecoinf.2014.06.007`

[5] L. Khiali, D. Ienco, M. Teisseire, Object-oriented satellite image time series analysis using a graph-based representation, Ecological Informatics 43 (2018) 52–64.

[6] F. Guttler, D. Ienco, J. Nin, M. Teisseire, P. Poncelet, A graph-based approach to detect spatiotemporal dynamics in satellite image time series, ISPRS Journal of Photogrammetry and Remote Sensing 130 (2017) 92–107.

[7] B. Bellón, A. Bégué, D. L. Seen, C. A. de Almeida, M. Simões, A remote sensing approach for regional-scale mapping of agricultural land-use systems based on NDVI time series, Remote Sensing 9 (6) (2017) 600.

[8] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, I. Rodes, Operational high resolution land cover map production at the country scale using satellite image time series, Remote Sensing 9 (1) (2017) 95.

[9] N. Kussul, M. Lavreniuk, S. Skakun, A. Shelestov, Deep learning classification of land cover and crop types using remote sensing data, IEEE Geosci. Remote Sensing Lett. 14 (5) (2017) 778–782.

[10] N. A. Abade, O. A. d. C. Júnior, R. F. Guimarães, S. N. de Oliveira, Comparative analysis of modis time-series classification using support vector machines and methods based upon distance and similarity measures in the brazilian cerrado-caatinga boundary, Remote Sensing 7 (9) (2015) 12160–12191.

[11] R. Flamary, M. Fauvel, M. D. Mura, S. Valero, Analysis of multitemporal classification techniques for forecasting image time series, IEEE Geosci. Remote Sensing Lett. 12 (5) (2015) 953–957.

[12] I. Heine, T. Jagdhuber, S. Itzerott, Classification and monitoring of reed belts using dual-polarimetric terrasar-x time series, Remote Sensing 8 (7).

[13] L. Zhang, B. Du, Deep learning for remote sensing data: A technical tutorial on the state of the art, IEEE Geoscience and Remote Sensing Magazine 4 (2016) 22–40.

[14] Y. Bengio, A. C. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE TPAMI 35 (8) (2013) 1798–1828.

[15] X. Zhu, D. Tuia, L. Mou, G. X. L. Zhang, F. Xu, F. Fraundorfer, Deep learning in remote sensing: A comprehensive review and list of resources, IEEE Geosci. Remote Sens. Mag. 5 (2017) 8–36.

[16] D. Ienco, R. Gaetano, C. Dupaquier, P. Maurel, Land cover classification via multitemporal spatial data by deep recurrent neural networks, IEEE GRSL 14 (10) (2017) 1685–1689.

[17] H. Lyu, H. Lu, L. Mou, Learning a transferable change rule from a recurrent neural network for land cover change detection, Remote Sensing 8 (6).

[18] P. Benedetti, D. Ienco, R. Gaetano, K. Ose, R. G. Pensa, S. Dupuy, M3fusion: A deep learning architecture for multi-{Scale/Modal/Temporal} satellite data fusion, CoRR abs/1803.01945.

[19] A. Graves, A. Mohamed, G. E. Hinton, Speech recognition with deep recurrent neural networks, in: ICASSP, 2013, pp. 6645–6649.

[20] T. Linzen, E. Dupoux, Y. Goldberg, Assessing the ability of lstms to learn syntax-sensitive dependencies, TACL 4 (2016) 521–535.

[21] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, A. Graves, Conditional image generation with pixelcnn decoders, in: NIPS, 2016, pp. 4790–4798.

[22] D. H. T. Minh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, P. Maurel, Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR sentinel-1, IEEE Geosci. Remote Sensing Lett. 15 (3) (2018) 464–468.

[23] E. Ndikumana, D. H. T. Minh, N. Baghdadi, D. Courault, L. Hossard, Deep recurrent neural network for agricultural classification using multitemporal SAR sentinel-1 for camargue, france, Remote Sensing 10 (8) (2018) 1217.

[24] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556.
URL `http://arxiv.org/abs/1409.1556`

[25] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: ICML10, 2010, pp. 807–814.

[26] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: ICML, 2015, pp. 448–456.

[27] G. E. Dahl, T. N. Sainath, G. E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout, in: ICASSP, 2013, pp. 8609–8613.

[28] K. Soma, R. Mori, R. Sato, N. Furumai, S. Nara, Simultaneous multichannel signal transfers via chaos in a recurrent neural network, Neural Computation 27 (5) (2015) 1083–1101.

[29] D. H. T. Minh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, P. Maurel, Deep recurrent neural networks for winter vegetation quality mapping via multitemporal sar sentinel-1, IEEE GRSL Preprint (-) (2018) –.

[30] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: EMNLP, 2014, pp. 1724–1734.

[31] L. Mou, P. Ghamisi, X. X. Zhu, Deep recurrent neural networks for hyperspectral image classification, IEEE TGRS 55 (7) (2017) 3639–3655.

[32] D. Britz, M. Y. Guan, M. Luong, Efficient attention using a fixed-size memory representation, in: EMNLP, 2017, pp. 392–400.

[33] R. Gaetano, D. Ienco, K. Ose, R. Cresson, Mrfusion: A deep learning architecture to fuse pan and ms imagery for land cover mapping, CoRR abs/1806.11452.

[34] S. Hou, X. Liu, Z. Wang, Dualnet: Learn complementary features for image recognition, in: IEEE ICCV, 2017, pp. 502–510.

[35] O. Hagolle, M. Huc, D. Villa Pascual, G. Dedieu, A Multi-Temporal 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 7 (3) (2015) 2668–2691.

[36] V. Lebourgeois, S. Dupuy, E. Vintrou, M. Ameline, S. Butler, A. Bégué, A combined random forest and OBIA classification scheme for mapping smallholder agriculture at different nomenclature levels using multisource data (simulated sentinel-2 time series, VHRS and DEM), Remote Sensing 9 (3) (2017) 259.

[37] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980.

[38] R. Gaetano, D. Ienco, K. Ose, R. Cresson, Mrfusion: A deep learning architecture to fuse PAN and MS imagery for land cover mapping, CoRR abs/1806.11452. arXiv:1806.11452.
URL http://arxiv.org/abs/1806.11452