International Journal of Pattern Recognition and Artificial Intelligence


# Influence of data representations and deep architectures in image time series classification

Mohamed Chelali*, Camille Kurtz*, Anne Puissant+ and Nicole Vincent*

*LIPADE, Université de Paris, Paris, France.
{firstname.lastname}@u-paris.fr
+LIVE (CNRS UMR 7362), Université de Strasbourg, Strasbourg, France.
{firstname.lastname}@unistra.fr

Image time series, such as Satellite Image Time Series (SITS) or MRI functional sequences in the medical domain, carry both spatial and temporal information. In many pattern recognition applications such as image classification, taking into account such rich information may be crucial and discrimative during the decision making stage. However, the extraction of spatio-temporal features from image time series is difficult to handle due to the complex representation of the data cube. In this article, we present a strategy based on Random Walk to build a novel segment-based representation of the data, passing from a $2D+t$ dimension to a $2D$ one, more easily manipulable and without losing too much spatial information. Such new representation is then used to feed a classical Convolutional Neural Network (CNN) in order to learn spatio-temporal features with only $2D$ convolutions and to classify image time series data for a particular classification problem. The influence of the way the $2D+t$ data are represented, as well as the impact of the network architectures on the results, are carefully studied. The interest of this approach is highlighted on a remote sensing application for the classification of complex agricultural crops.

*Keywords*: Image Time Series; spatio-temporal features; Random Walk; Convolutional Neural Networks; Remote Sensing; Satellite images.

## 1. Introduction

An image time series is an ordered set of images taken from the same scene at different dates. Such data provide rich information with the temporal evolution of the studied area. In remote sensing applications, many constellations of satellites acquire images with a high spatial, spectral and temporal resolution around the world leading to Satellite Image Time Series (SITS). For example, the Sentinel-2 sensors produce optical SITS with a revisit time of 5 days and a spatial resolution of 10 – 20 meters.

Such series of images help understanding environmental evolution, studying the causes of various changes, and predicting future evolution. Temporal information, integrated with spectral and spatial dimensions, enables in particular, the analysis of complex patterns involved in applications related to land cover mapping (e.g. agricultural zones, urban areas) or to the identification of land use changes (e.g.

urbanization, deforestation) and the production of accurate land-cover maps of a territory.[11]

A major issue when analyzing image time series is to consider simultaneously the temporal and the spatial dimensions of the $2D + t$ data-cube. In this context, state-of-the-art methods for SITS analysis are actually mainly based on temporal information[16] at pixel level. But in some specific applications, this may not be sufficient to get satisfactory results. Taking both temporal and spatial aspects into account at the same time can, for example, make it easier to discriminate between different complex land cover classes (e.g. agricultural practices for a specific crop, urban vs. peri-urban areas). Note that here, our objective is to classify complex land-cover classes prone to confusions when a single date image is used.

This article focuses on the problem of spatio-temporal feature extraction for the classification of image time series, using a deep learning strategy. In this context, we define a novel spatio-temporal representation of image time series that makes it possible to consider classical Convolutional Neural Network (CNN) frameworks, originally proposed for the analysis of $2D$ images. Our methodological contribution is the proposal of a transformation to represent $2D + t$ data as $2D$ images without losing too much spatial information. It relies on the construction of sets of $(1D)$ segments using a Random Walk paradigm to decrease the spatial dimension of the data. This new data representation is then used to feed a CNN in order: (1) to learn spatio-temporal features with only $2D$ filters, involving at the same time temporal and spatial information, and (2) to classify image time series data according to a particular thematic problem.

We also study the influence, on the decision made by the system, of the way the $2D + t$ data are represented (temporal vs. spatial information), and the impact of the network architectures (number of parameters to be optimized), on the learned spatio-temporal convolutional features.

The remainder of this article is organized as follow. Section 2 presents some related works for SITS analysis. Section 3 describes the proposed representation of the image time series for a CNN-based analysis. An experimental study, focusing on the classification of agricultural crops in the remote sensing domain is described in Section 4. Section 5 discusses the obtained results with our approach and comparative methods. Finally, conclusion and perspectives will be found in Section 6.

## 2. Related works on SITS analysis

SITS enable the observation of the Earth surface at multiple instants. Such data improves our knowledge and understanding of environmental evolution and changes, which may be of different types, origins and duration. For a detailed survey, see.[5]

Pioneer methods processed single images from image stacks. On each image, different measurements per pixel were considered as independent features and involved in classical machine learning-based procedures. In such approaches, the date of the measurements was ignored in the feature space. Methods designed for bi-temporal

analysis locate and study abrupt changes occurring between the two observations. These methods include image differencing,[3] ratio-ing[13] or vector change analysis.[14]

Another family of methods is more directly dedicated to the analysis of image time series. Most of them are based on multi-date classification. Among them, we find radiometric trajectory analysis.[24] These methods exploit the evolution of land cover (e.g. seasons, vegetation evolution[21]), and take into account the chronology by using dedicated time series analysis methods.[2] Every pixel is considered as temporally ordered (and aligned) series of measurements, and the changes in the measurement values through time are analyzed to find (temporal) patterns, using statistical or symbolic approaches.

Some methods first propose a new representation of the SITS into a new space. We can cite "frequency-domain" approaches that include spectral analysis, wavelet analysis.[1] Other methods extract more discriminative "hand-crafted" features from a new enriched space.[4,18,19] Concerning the classification step, the classical approaches measure similarity between any incoming sample (that can be enriched with the "hand-crafted" features) and the training set. They assign the label of the most similar class using e.g. the Euclidean distance based on a nearest neighbor algorithm or / and the Dynamic Time Wrapping method.[17]

More recently, deep learning paradigms have been considered to classify remote sensing images and generate land-cover maps. In general, Convolutional Neural Networks (CNN) are used to deal with the spatial domain of the data by applying $2D$ convolutions.[8] When dealing with image time series, convolutions can be applied in the temporal domain.[16] Another type of deep learning architecture that is designed for temporal data is Recurrent Neural Network (RNN) such as Long-Short Term Memory (LSTM), used successfully in.[10,20] In this context, deep learning approaches outperform traditional classification algorithms such as Random Forest,[12] but they do not directly take into account the spatial dimension of the data as they consider pixels in an independent way. Some approaches have been proposed to consider both the temporal and the spatial dimensions of the $2D + t$ data-cube. A common strategy is to train two models, one for spatial dimensions and one for the temporal dimension, then to fuse their results at the decision level.[6] In video analysis, spatio-temporal features are learned directly using deep $3D$ convolutional networks[23] but such strategy requires the learning of a huge number of parameters to define a good model.

In this paper, our strategy is to classify a SITS using a classical $2D$ CNN model, thanks to a new representation of image time series embedding simultaneously temporal and spatial information of the data-cube. We compare our results with competitive approaches, including the use of $1D$ convolutions applied in the temporal domain[16] to classify temporal pixels (which is the current state-of-the-art). This enables to assess the plus-value of considering spatio-temporal features instead of solely temporal ones when classifying image time series.

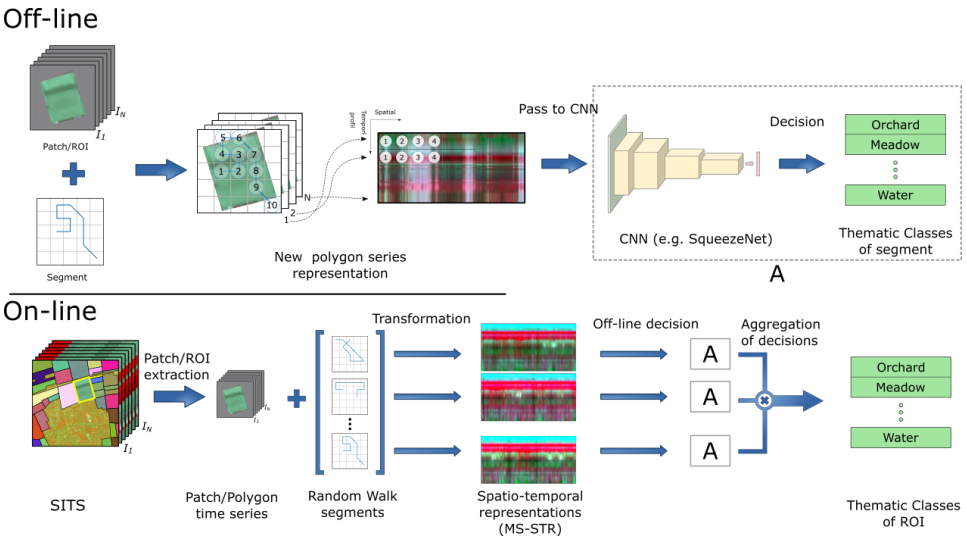



Fig. 1: Flowchart of our method for image time series deep classification based on a planar spatio-temporal data representation obtained from Random Walk based segments: (top) off-line (i.e. learning) phase and (bottom) on-line (i.e. testing) phase of the classification process.

## 3. Multi-Segment Spatio Temporal Representation (MS-STR) for SITS analysis

The proposed method aims to classify image time series from spatio-temporal features. The underlying strategy is to use a $2D$ input in a classical deep neural network architecture in order to learn a spatio-temporal model from the $2D + t$ data. Figure 1 illustrates the global workflow of our system, with the traditional off-line (i.e. learning) and on-line (i.e. testing) phases of a classification process. Since our system is dedicated to classification of objects of interest (e.g. agricultural crops from satellite images), the initial input data may be an image centered over a specific object, an image patch, or only the connected pixels of a region of interest (ROI), modeled as a polygon. In any case, we will use the term "image" for the input data.

We manage to consider some $2D$ elements to perform the learning phase in the off-line process. In this way, we differ from other approaches considering a $1D$ structure[16] or a $3D$ one[23] as we find in the state-of-the-art methods.

To this end, we start by transforming the original $2D+t$ data into planar entities containing spatio-temporal data built from $1D$ spatial segments over time. We refer to our method as Multi-Segment Spatio Temporal Representation (MS-STR). Such a representation is then transferred as the input of a CNN to achieve a classification of the segments that are built in off-line and the classifier is then used in the on-line process. The network is trained in order to learn the labels from both the spatial,

as well as the temporal information contained in the data.

### 3.1. *Data transformation*

We first explain how to transform the original $2D + t$ data to less complex $2D$ representations that contain spatio-temporal data built from $1D$ spatial segments.

**From pixels to segments.** Traditional methods that only handle temporal information consider the $2D$ domain as a set / bag of pixels, i.e. $0D$ entities. The pixels are generally characterized by the temporal series of the pixel intensities. In our case, we include some spatial information, leading to the notion of segments which are $1D$ spatial entities. An input image is then replaced by a set of $1D$ segment entities, where $L$ is the length of the segments included in the input image.

In a $1D$ segment each pixel has 2 neighbors, except for the two extreme pixels. Our transformation will then decrease the spatial information with keeping only 2 nearest neighbors.

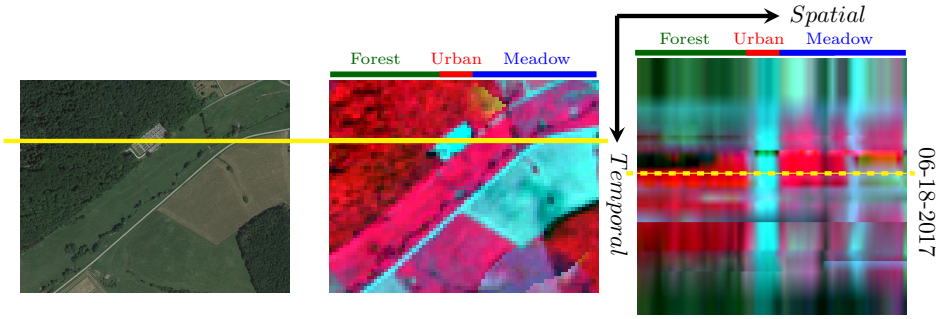
Different strategies to define $1D$ segments in the original $2D$ space are studied and compared in this work. For each chosen strategy, we apply the process $N_p$ times from an input data, producing $N_p$ different segments, in order to keep enough neighbors; the pixel orders are then chosen according to a parametrization of these segments. In this way, the spatial representation complexity of the images is decreased, from $2D$ to $1D$ segments.

Next, the segments characterized with temporal information, leading to the notion of $2D$ spatio-temporal data, are classified.

**From segments to $2D$ representations.** For a given series composed of $N$ images (i.e. $N$ temporal acquisitions), segments are first extracted. They are used for the learning of the classification model. The segment pixels are spatially represented by the pixel index within the segment. These $1D$ spatial segments will now be enriched with temporal information to build $2D$ spatio-temporal data (see top of Figure 1).

With each of the $N_p$ segments, we associate a $2D$ structure. In the abscissa, is considered the index of the pixel in the segment (from the initial pixel) and in the ordinate, is considered the evolution of the intensity of the pixels over time. This leads to a novel $2D$ representation composed of $N$ rows ($N$ is the number of images in the SITS) in the temporal domain and $L$ columns ($L$ is the length of the considered segment) in the spatial domain. This image can then be interpreted as a partial spatio-temporal $2D$ representation of the input $2D + t$ image time series.

When applying the transformation process to the $N_p$ segments, we finally obtain $N_p$ spatio-temporal $2D$ representations from the original image time series, called Multi-Segment Spatio Temporal Representation (MS-STR). These representations will be used as input of a learning process, the segment classes are the classes of the annotated input image they belong to.




(a) Google Earth image (1665 × 2056 pixels)

(b) Sentinel-2 image at 06-18-2017 (62 × 78 pixels)

(c) Spatio-temporal representation associated with the yellow segment in (b)

Fig. 2: Spatio-temporal representation from an image time series: (a) a high resolution image taken from Google Earth sensed over a particular agricultural area; (b) a Sentinel-2 image sensed on June 18, 2017 in false color (near-infearned NIR, red R and green G). This image belongs to a SITS sensed during the year 2017 over the same region as the Google Earth image; (c) A spatio-temporal representation created from the yellow segment in the Sentinel-2 image.

For illustrative purpose, Figure 2(a) displays a Google Earth high resolution image of a specific agricultural area while Figure 2(b) shows a Sentinel-2 image of the same region acquired on June 18, 2017. The Sentinel-2 image belongs to a SITS sensed over the year 2017. This image is depicted in false color (NIR, R, G that represent respectively the near-infrared, red and green spectral bands). The yellow segment drawn on these two images is passing through different zones, on the left part is a forest, then in the middle is a house and finally on the right are two meadows that are not of the same type. Figure 2(c) finally shows the spatio-temporal $2D$ representation created from this horizontal yellow segment. In this representation, the dashed yellow segment is showing the date where the Sentinel-2 image in (b) has been acquired.

In the left part of the representation, we can observe that a zone rather homogeneous on the horizontal direction is associated with the forest zone, the red zone indicates the presence of active chlorophyll from June to October. Then, in the middle a long and narrow bluish rectangle is associated with the house with low value in the NIR that is more generally linked to vegetation. In the right part is an other region where can be distinguished two behaviors, a first meadow with a red temporal interval indicating that the grass is growing, and the second meadow where the vegetation growing occurs during a shorter temporal period. This may indicate that the meadow has been cropped in the late Spring, the vegetation is then stopped and a blue zone is present. We can also notice that the grass is growing before the tree leafs.

### 3.2. *Segment construction strategies*

In this study, two different strategies were considered to build segments:

- **Scanning strategy (scan).** Here we consider all the rows and columns of pixels in the input image to build $1D$ segments. The dimensions of the input image limit both the number and the lengths of possible segments. To guarantee similar lengths $L$ for each segment, it is needed to replicate the values on segments shorter than $L$ (this may correspond to segments extracted from the borders of the image). In this way, the pixels are considered only twice in the new representations, and for each pixel, only 4 neighbors are considered, the 4 nearest neighbors.
- **Random Walk (RW) based segment.** A Random Walk[7] is a mathematical process based on a random iterative process. Each iteration is a step with Markovian properties. Here, the Random Walk is used to generate a random segment in a $2D$ image space with length $L$, noted $RW(L)$. The first point of the segment is chosen randomly on the $2D$ image and for next point, 8 directions are possible.

  Given an input image, we proceed to $N_p$ initializations of $N_p$ Random Walk segments. For each one, a $2D$ image is then built, where the rows correspond to the pixel values of the pixels in the segment extracted from the different images of the series. The chronology is related to the line number. The middle of the on-line part of Figure 1 illustrates the spatio-temporal representations from three different segments built from an input image.

These two types of segments present several differences. On the one hand, RWs consider a larger number of neighbors for each pixel. Then the variety of information on the spatial configurations is larger than in the scan approach. On the other hand, in the scan approach, the number of segments is limited whereas in the other case, the segments are randomly initiated.

### 3.3. *CNN model (architecture)*

Convolutional Neural Networks (CNN) refer to the family of deep learning algorithms. CNN-based systems are generally composed of two parts. The first one is designed to feature extraction, it has many neuron layers that apply convolutions on the previous ones. The neurons of each layer are activated by non-linear functions (e.g. sigmoïde, ReLU) in order to keep the most representative features (high order features). We find also max-pooling layers between convolutional layers to reduce progressively the quantity of the inputs and the number of the parameters to be computed to define the network, and hence to also control over-fitting. The second part may be a classifier. Generally, it is fully connected layers that provide a probability vector, on which is plugged a softmax function to predict the class label of input data.

We have chosen the SqueezeNet model[9] as baseline CNN but any other $2D$ CNN model can be used and we will achieve some comparisons in order to discuss the complexity of the concrete problem that will be used to illustrate our methodology. SqueezeNet has interesting theoretical properties, few parameters, and reaches the same accuracy level as the AlexNet model on the ImageNet dataset. The training of the model is then faster. The architecture of SqueezeNet introduces a new module called Fire composed of a squeeze layer using $1 \times 1$ convolution filters followed by expand layer that contains a mix of $1 \times 1$ and $3 \times 3$ convolution filters. Also, its classifier is based on a global average pooling over feature maps, potentially decreasing the overfitting effect. The CNN model, whatever it is, is trained with the $2D$ spatio-temporal representations, i.e. the MS-STR, obtained from each input image time series from the training set.

### 3.4. *Decision making at polygon level*

As already mentioned, our input data are polygons representing objects of interest in SITS. Each input data is associated with a set of $N_p$ segments, leading to a MS-STR. $N_p$ is consequently a parameter of the method. With each segment is associated a $2D$ planar spatio-temporal representation. Thanks to the classifier described in Section 3.3, a class label is predicted for each $2D$ spatio-temporal representation (i.e. for each segment) with some probability. To classify a polygon, we proceed by taking average of the returned probabilities by the model for the $N_p$ segments of the polygon and we affect the class label with the highest probability, ensuring ultimately a unique decision per MS-STR, and consequently per input image.

### 3.5. *Implication of temporal and spatial information in the process*

In order to show the capacity of the proposed representation to carry both types of information (spatial or temporal), we propose different approaches. First, we can compare the results obtained when considering the SITS as a whole, by considering our MS-STR approach, or when considering only the set of temporal pixels, as proposed by most of the state-of-the-art methods such as TempCNN.[16] This will allow to assess the importance of spatial information.

Second, we propose to analyze the convolutional filters learned in the CNN learning process. In the first layers of the CNN, the convolution kernels are processing in vertical the temporal aspect and in horizontal the spatial aspect. Our aim is to analysis the type of information (temporal vs. spatial) captured by each filter. For this, we first build synthetic images with only temporal information $I_t$ and with only spatial information $I_s$. These fake images are provided as input to the CNN, learned in the off-line process. For each filter ($k$ index) of the first layer, the energy of the answers is computed and noted $E_k(I_t)$ and $E_k(I_s)$. The computation of the spatio-temporal ratio $R_{st}(k)$ between the two energies indicates which aspect

(spatial or temporal) is more or less associated with the $k$ filter:

$$R_{st}(k) = \frac{E_k(I_s)}{E_k(I_t)} \tag{1}$$

We can consider three types of filters:

- **spatial filters** are those that are more linked to the spatial variations: the ratio $R_{st}(k)$ is greater than $1 + \mu$;
- **temporal filters** are those that are more linked to temporal variations: the ratio $R_{st}(k)$ is lower than $1 - \nu$;
- **spatio-temporal filters** are those where the ratio $R_{st}(k)$ is between $1 - \nu$ and $1 + \mu$; they are linked both to temporal and spatial variations.

Here, $\mu$ and $\nu$ are parameters to be chosen.

## 4. Experimental study

The experimental study is focused on a remote sensing application, the classification of agricultural crop fields from a SITS. We consider a binary classification task, where the goal is to discriminate within two agricultural thematic classes: traditional vs. intensive orchards. The automatic identification of these classes is a complex task since orchards are subject to many agricultural practices depending on the season and the territory management policy. In order to differentiate these two classes, spatio-temporal features carry useful information to discriminate the agricultural practices.

### 4.1. *Material*

We dispose of a SITS provided by the satellite Sentinel-2, containing $N = 50$ optical images sensed in 2017 over the same geographical area (East of France – tile 32ULU). Figure 3 displays the temporal distribution of the images belonging to the SITS. The images have been corrected and orthorectified by the French Theia program[a] to be radiometrically comparable. We also dispose of the cloud, shadow and saturation masks associated with each image. A preprocessing step was applied on the images with a linear interpolation on masked pixels to fill the missing values in the SITS.

For each image, only three bands are kept which are near-infrared (Nir), red (R) and green (G). The blue band (B) is considered as useless in the literature to discriminate different kinds of agricultural fields and is also sensitive to atmospheric effects. All these bands have a spatial resolution of 10 meters.

The used reference data are extracted from the (freely distributed) RPG[b], which is the official agricultural parcel delineations (in our context orchards). Some examples of polygons are represented in Figure 1. These polygons have been corrected

[a] `https://theia.cnes.fr/`
[b] `http://professionnels.ign.fr/rpg`

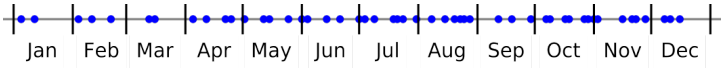



Fig. 3: Temporal distribution of the images from the SITS over the year 2017.

Table 1: Summary of the data: (first col.) Initial number of polygons per class; (two last col.) Number of spatio-temporal segments depending on the segment construction strategy.

| **Classes** | **# polygons** | **# Spatio-temp. rep. for** *scan* | **# Spatio-temp. rep. for** *RW* |
|---|---|---|---|
| **Int. orchards** | 137 | 2 998 | 10 609 |
| **Trad. orchards** | 193 | 2 024 | 12 555 |
| **Total** | 330 | 5 022 | 23 164 |

by photo-interpretation to ensure a good delimitation of the parcels. The reference data used in our experiment are the semantic labels of these polygons (traditional or intensive orchards). These polygons are leading to a new time series of polygons, noted Polygon Image Time Series (PITS).

### 4.2. *Data preparation*

First, PITS are formed, then we analyze the importance of the spatial relationship of pixels, so $N_p$ segments are extracted from the ROI. For the *scan* strategy, the number of possible $1D$ segments depends on the ROI size. For the RW strategy, we made the $N_p$ value depends on the area (i.e. number of pixels) of the ROI. In our case we consider a percentage of the number of pixels to set the $N_p$ value. The chosen percentage is 50% of the ROI area. The average number of segments for the *scan* strategy is 487 with a standard deviation equal to 110. Table 1 displays the number of instances of polygons per class and the number of segments built from these data according to the segment construction strategy.

In the following, we study the impact of the length $L$ of the segments. This enables to evaluate the impact of adding more spatial information to learn spatio-temporal features instead of considering single $0D$ pixels, as this is the case in most of the classical approaches. The used lengths $L$ are 10, 50, 100 and 224. The largest one depends on the maximum input size of the CNN SqueezeNet model. When building the $224 \times 224$ $2D$ image from the segments, if the segments are less than 224 pixel long, we center them horizontally and the rest of columns are fixed to zero value. Table 1 indicates the actual number of segments.

For the temporal dimension (vertical axis), we propose two strategies. The first one is to center the original information from the $N$ input images vertically ($N = 50$). The remaining top and bottom lines are fixed to zero value. The second one

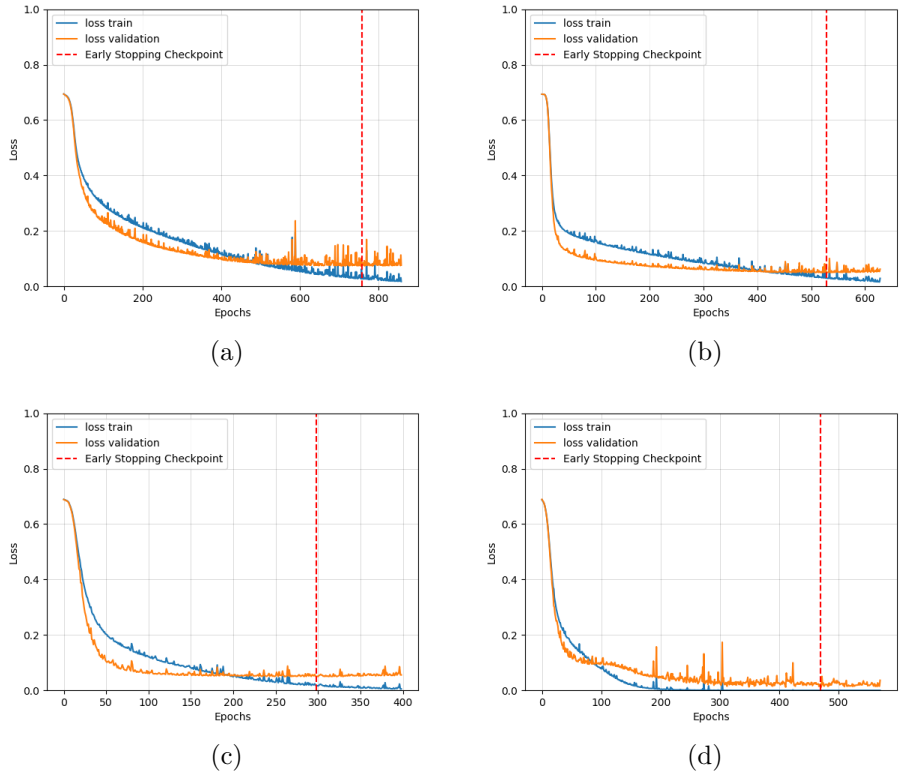



Fig. 4: Training phase loss curves, according to the length $L$ of Random Walk paths: (a) $L = 10$; (b) $L = 50$; (c) $L = 100$ and (d) $L = 224$.

is to fill the 224 values by applying a linear interpolation in the STIS on time information. We assume that the temporal information between two consecutive dates is monotonic and linear. The interpolation is then done by considering that we only have 224 days in the year so that one day is done with about 39 hours. For the initial dates (beginning of the year of 224), we affect the temporal information of the first date in the SITS. For the last dates (end of the year of the 224), we affect the last known temporal information in the SITS.

The data normalization is a linear transform based on the maximum and the minimum values of the dataset after values are limited with 2% (or 98%) percentile, as suggested in.[16]

### 4.3. *Learning and validation protocol*

The experiments are validated using a five-fold cross validation strategy. Each time, we split the dataset into three subsets at polygon level with sizes of 60%, 20% and

Table 2: Classification results (overall accuracy – OA and standard deviation – STD) obtained with our spatio-temporal representations.

| **MS-STR** | **From scratch** | | **Fine-tuning** | |
|---|---|---|---|---|
| **Lengths of the segments** | **OA** | **STD** | **OA** | **STD** |
| **with original temporal information (N=50)** | | | | |
| *scan(10)* | **64.24** | **7.42** | 74.54 | 2.60 |
| *RW(10)* | 63.33 | 1.76 | 87.27 | 1.54 |
| *RW(50)* | 63.33 | 1.76 | 94.84 | 1.54 |
| *RW(100)* | 62.72 | 1.54 | **95.75** | **1.48** |
| *RW(224)* | 63.93 | 4.22 | 95.75 | 2.42 |
| **with temporal interpolation (N=224)** | | | | |
| *scan(10)* | 61.51 | 3.53 | 76.06 | 2.60 |
| *RW(10)* | 69.69 | 1.15 | 88.78 | 2.96 |
| *RW(50)* | **92.72** | **2.42** | 95.75 | 1.13 |
| *RW(100)* | 92.12 | 3.76 | 96.66 | 0.60 |
| *RW(224)* | 87.27 | 6.68 | **97.27** | **1.13** |

20% representing respectively training, validation and test sets. The CNN model is then trained and evaluated five times at decision level. In the end, we report the average overall accuracy (OA) of the five splits and indicate the standard-deviation (STD).

The model is trained using *Adam* optimizer with a learning rate of $10^{-6}$ and default values of the other parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$) with a batch size of 8. We limit the number of epochs to 2000, following an early stopping technique with a patience number of 100. The experiments are done on a laptop machine with a Nvidia GPU model GTX 1050 Ti with Max-Q Design (4GB). We used the PYTORCH implementation of SqueezeNet[c].

According to the limited number of polygons, the training is operated with two strategies. In the first one, the model is trained from scratch and in the second one it is initialized with weights obtained with the IMAGENET database in a classification problem (ILSVRC challenge) and then fine-tuned with our data.

## 5. Results and discussion

In this section we present the results of different experimental studies, enabling to analyze the different aspects both of the concrete problem and of the system to solve it. First, we consider the length of the segments and spatio-temporal represen-

[c] `https://github.com/pytorch/vision/blob/master/torchvision/models/squeezenet.py`

Table 3: Classification results (overall accuracy – OA and standard deviation – STD) with the different architectures of TempCNN[16] and a LSTM.[10]

| | **TempCNN[16] architectures** | | | | | | | **LSTM[10]** |
|---|---|---|---|---|---|---|---|---|
| **Nb filters** | **16** | **32** | **64** | **128** | **256** | **512** | **1024** | (n.a.) |
| **with original temporal information (N=50)** | | | | | | | | |
| **OA** | 70.60 | 70.00 | **79.22** | 68.78 | 71.51 | 77.27 | 77.57 | 44.84 |
| **STD** | 6.68 | 9.45 | **7.35** | 4.24 | 1.76 | 4.97 | 7.00 | 2.26 |
| **with temporal interpolation (N=224)** | | | | | | | | |
| **OA** | 74.54 | 80.90 | 81.21 | 78.18 | **81.81** | 78.48 | 81.21 | 61.51 |
| **STD** | 8.15 | 4.94 | 7.01 | 5.63 | **6.57** | 6.16 | 4.94 | 6.74 |

tations, then we consider the use of different CNNs and we discuss their efficiency. Finally, we study the temporal or the spatial adaptation of the convolutions in the first layer of the learned CNN.

### 5.1. *Influence of segment lengths and number of dates*

The proposed $2D$ spatio-temporal representations are used to feed the chosen CNN. For the *scan* strategy, we just use the segment length $L$ of 10 since we are limited by the frontiers of the ROIs. At segment level, Figure 4 illustrates the obtained loss curves when the model is trained from scratch with the different lengths of the *RW* segments, respectively 10, 50, 100 and 224. We observe that the training is done in the best conditions with the different lengths of the Random Walk. In the loss curve of *RW(10)*, we observe strong oscillations in the curve which is not the case in others. This is potentially due to the lack of information in the images provided to the CNN (a lot of zero –black– values in the input image), and each time when increasing the length $L$, the validation loss (orange curve) decreases leading to better learning rates without losing in generalization capacity.

In the first experiment, we consider the original PITS composed of $N = 50$ images. The on-line classification results (overall accuracy) with spatio-temporal representations (with original dates) are reported in Table 2 (upper part). All the scores are in the same range. It can be noticed that in the result obtained with a from scratch learning strategy, *scan(10)* is slightly better but this has to be tempered by the very high standard deviation value. Indeed, with the different lengths of the Random Walk, we kept spatial information that allows to distinguish between the two considered classes (traditional and intensive orchards). When the lengths increase, the accuracy increases too showing the importance of spatial information. As expected, the learning achieved through a fine-tuning process enables to obtain a classification with about 20% better accuracy. The best results are obtained with

Table 4: Architecture of a small custom CNN.

| Layer (type) | Output Shape | # Params |
|---|---|---|
| input | [-1, 3, 224, 224] | 0 |
| Conv2d-1 | [-1, 64, 110, 110] | 4,864 |
| BatchNorm2d-2 | [-1, 64, 110, 110] | 128 |
| ReLU-3 | [-1, 64, 110, 110] | 0 |
| Conv2d-4 | [-1, 64, 54, 54] | 36,928 |
| BatchNorm2d-5 | [-1, 64, 54, 54] | 128 |
| ReLU-6 | [-1, 64, 54, 54] | 0 |
| MaxPool2d-7 | [-1, 64, 27, 27] | 0 |
| Conv2d-8 | [-1, 128, 25, 25] | 73,856 |
| BatchNorm2d-9 | [-1, 128, 25, 25] | 256 |
| ReLU-10 | [-1, 128, 25, 25] | 0 |
| MaxPool2d-11 | [-1, 128, 12, 12] | 0 |
| Conv2d-12 | [-1, 128, 10, 10] | 147,584 |
| BatchNorm2d-13 | [-1, 128, 10, 10] | 256 |
| ReLU-14 | [-1, 128, 10, 10] | 0 |
| Dropout-15 | [-1, 128, 10, 10] | 0 |
| Conv2d-16 | [-1, nb class, 10, 10] | 516 |
| AdaptiveAvgPool2d-17 | [-1, nb class, 1, 1] | 0 |
| Total params | – | 264 516 |

*RW(100)* and with the benefit of a fine-tuning.

In a next experiment, we consider a longer time series obtained through interpolation of the initial data, to fill the 224 columns of the data representation. Table 2 (bottom part) presents the classification results. All scores are increased compared to those with less temporal information (Table 2, upper part). This can be explained by the non-regular temporal distribution of the original images. Then the weights associated with the temporal data are applied to information that are not comparable. Whereas, with the linear interpolation, we make the temporal distribution regular to obtain 224 dates. The approach using *scan(10)* is less efficient than those based on *RW*. All the obtained scores are in the same range and the fine-tuning strategy shows the interest of a better initialization.

### 5.2. *Comparison with other state-of-the-art systems*

To evaluate the plus-value of the MS-STR approach, we compare the on-line classification scores with state-of-the-art methods. We have then selected two comparative methods. The first one is TempCNN[16] where the convolutions are applied only in the temporal domain ($1D$ convolutions). The filter sizes are fixed following the criterion given in[16] with a kernel size of 5 when considering the original dates, and

11 when considering the interpolated dates, as the size is depending on the temporal size of the series. Note that the TempCNN model is proposed with different architectures (depths of the network), leading to different numbers of filters. Then, we have experimented several architectures in order to optimize the architecture with respect to the data. We used the implementation of TempCNN provided by the authors[d]. The second method is a LSTM[10] network which is a variety of RNN. The used network is composed with 3 LSTM layers and a fully connected one as a classifier. For a fair comparison purpose, we trained and validated these methods using the same data and validation protocol than the ones used for our model.

Tables 3 reports the obtained results with TempCNN and LSTM respectively, considering the time series with $N = 50$ and 224 dates. Best scores of TempCNN are obtained considering 64 filters for short time series and 256 filters for longer time series. Here also, the interpolated time series enable to improve the efficiency of the system. LSTM provides worst score with the original temporal data ($N = 50$) and it is increased with interpolated temporal information but always in the last position. As possible reason to this disappointing result, we do not need, in our case, to manage series with different lengths, and our features are not only depending on time (spatio-temporal information). From this, we deduce that the considered LSTM system is not adapted to the considered task.

Now we can compare these results with those we have presented previously. Two contexts can be distinguished, when training from scratch and when considering a fine-tuning process. When training from scratch, our best score (92.72) is better than those obtained with TempCNN (81.81). When we use fine-tuning, we outperform them still more (97.27). We can notice that nearly all the scores obtained with our MS-STR based method, even the not so good ones are higher (with and without fine-tuning) than with TempCNN. Moreover, we can also notice our results are more stable than with TempCNN as the standard deviations computed from the five folds are divided by four. This highlights, for the context studied here, the benefice of considering a classical $2D$ CNN model for classifying $2D + t$ images combined with our spatio-temporal representations.

The obtained scores, thanks to our spatio-temporal representation, demonstrate the impact of adding the spatial information in the classification task. In the following, we study the impact of the architecture of the CNN considered in the learning process to classify the segment representation and of course we also compare with the competitive methods.

### 5.3. *Impact of the choice of the CNN*

The methodology (MS-STR) we have presented is based on the use of a $2D$ convolutional network. The network can be changed but the global architecture of the system is not modified. In this study, we are interested in the impact of the CNN

[d] `https://github.com/charlotte-pel/temporalCNN`

Table 5: Impact of the choice of the CNN for the classification task.

| | CNN | From scratch OA | From scratch STD | Fine-tuning OA | Fine-tuning STD | #Params (↓) |
|---|---|---|---|---|---|---|
| **MS-STR** | VGG16[22] | 92.42 | 2.14 | **97.87** | **7.42** | 134 268 738 |
| | AlexNet[15] | 88.78 | 3.26 | 91.51 | 3.66 | 57 020 228 |
| | SqueezeNet[9] | 92.12 | 3.76 | 96.66 | 0.60 | 723 522 |
| | Custom CNN | **96.66** | **1.13** | – | | 264 516 |
| **Others** | TempCNN[16] | 81.81 | 6.57 | – | | 3 939 586 |
| | LSTM[10] | 61.51 | 6.74 | – | | 1 434 626 |

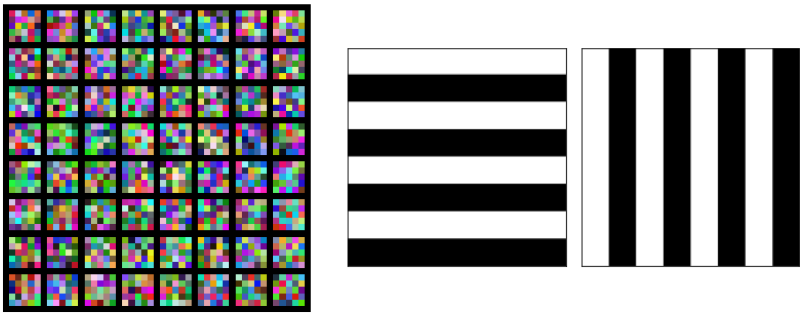
choice on the global results. One of our aim is to study the impact of the choice of the CNN on the obtained results, in particular the number of network parameters. To this end, we selected two CNN heavier than SqueezeNet and one lighter. There are described hereinafter:

(1) SqueezeNet[9] that is the network used in the previous studies whose results have been commented previously;
(2) AlexNet[15] and VGG16[22] models, as representative CNNs used in various classification tasks;
(3) a custom CNN model composed with four blocks: $2D$ convolutions, a $2D$ batch Normalization and ReLu activation function. The classifier is composed by a dropout layer, a $2D$ convolution with the number of classes as output channels number and a $2D$ adaptive average pooling layer to get the number of classes values. The architecture is detailed in Table 4.

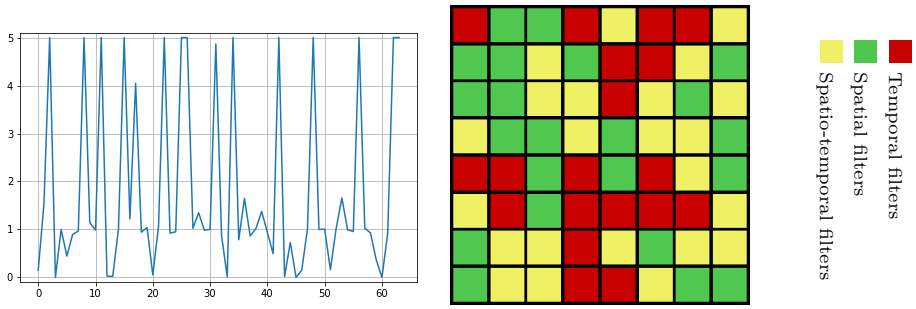
The comparison is done using the RW with length $L = 100$ and considering the decision with $N_p$ corresponding to 50% of the area of the polygon. Results are provided in Table 5, as well as the number of weights that have to be determined in the learning phase. First we can see the four MS-STR systems provide better results than when considering only time approaches (Others). In the cases where fine-tunings from ImageNet were possible, we can observe the significant improvement enabled by a fine-tuning strategy during the learning phase. But we also can notice the same result can be obtained with a less deeper network with less weights to be fixed. The results of these four MS-STR systems are quite disparate between 88% and 97% showing the efficiency of the methodology is depending both on the architecture and on the computed features. However, training from scratch CustomCNN requires a much larger number of epochs than fine-tuning VGG16 on our data (with pre-training on ImageNet), with comparable scores.

### 5.4. *Analysis of the convolutional filters learned by the CNN*

The methodology proposed in Section 3.5 will be applied on the MS-STR method



(a) Learned filters ($1^{st}$ CNN layer) (b) Synthetic images




(c) Energy ratios $R_{st}$ of the 64 filters (d) Classification of the 64 filters

Fig. 5: Importance of temporal vs. spatial features that are extracted with learned filters from the first layer of the custom CNN.

using our custom CNN. This is motivated by the size of this smaller CNN. In order to label the 64 filters of the first layer (illustrated in Figure 5(a)), the used synthetic images, $I_t$ and $I_s$, are generated by alternating the white and black color $f$ times, where $f$ is the number of color change. In our case, we chose $f = 8$. The obtained $I_t$ and $I_s$ images are illustrated respectively in Figure 5(b). When feeding the CNN with these images, we compute the energy of each of the obtained 64 feature maps where one convolution and a ReLu process are applied. We notice that the energy $E$ is computed on the output of the first activation function on the model (the output of the fourth layer in Table 4) in order to consider only the most representative features. Then, we apply the formula 1 to obtain the different ratios $R_{st}$. Figure 5(c) presents the graph figuring the ratios of energy $R_{st}(k)$ with respect to the kernel index $k$.

The identification of which filters are linked to temporal, spatial or spatio-temporal variations is done by setting $\mu = \nu = 0.1$. To illustrate the classification of the filters, a color is affected according to the conditions presented in Section 3.5, we then obtain the 64-cells grid depicted in Figure 5(d). The red color refers to

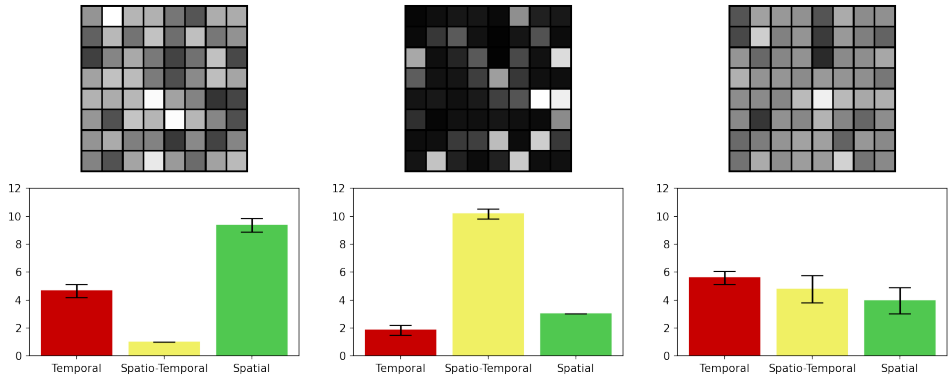
temporal filters, green refers to spatial filters and yellow to spatio-temporal filters. From this grid, we notice that all the categories of filters have been learned in the classification system built. Furthermore, the learned filters are well balanced in the three sets, temporal, spatial and spatio-temporal filters. This enforces the importance to consider both spatial and temporal aspects, not only independently, but also in a conjoint manner. Capturing such spatio-temporal information is actually not feasible when the fusion of both aspects is done at the decision level of the system.

We can then analyze which filters are more active according to the input images of the CNN, in our case real spatio-temporal representations. We illustrate the analysis with representations of three examples of crop-fields, an intensive orchard, a traditional orchard and a meadow. The question that arises is what kind of filters are the more active when analyzing the data? First the energy associated with the images are computed in order to count the most active filters. In order to produce significant results we have limited the counting to the 15 highest energies and we produce a bar graph counting the number of filters in each category indicated through the color. The study can be done at STR level or at the more semantical level of crop field by averaging the histograms associated to the $N_p$ representations of the MS-STR used in the decision making step.

The obtained results are illustrated in Figure 6(a,b,c). When comparing the histograms of intensive and traditional orchards, we notice the use of spatial filters is rather important either from the filters labeled as spatio-temporal or spatial. The regularity of the rows of trees in intensive orchards is materialized by the larger use of specific spatial filters. We have added in this experiment the results obtained from a meadow that is characterized by the homogeneity of the spatial behavior along the year and this is confirmed by the small use of spatial filters compared to the importance of the temporal filters. This analysis shows that temporal information are very important in the classification problem we consider, but spatial information are important too and justify the use of MS-STR method as well as the improvements compared to temporal approaches.

## 6. Conclusion

In this article, we present a MS-STR method to classify an image time series based on a spatio-temporal representation. This representation aims to reduce the structure of the data from $2D+t$ to $2D$ without losing too much the spatial relationship of pixels and the temporal one. Then, these new representation images are used to feed any classical 2D CNN to perform a classification. With the proposed representation, the applied $2D$ convolutions lead to a spatio-temporal feature extraction. If the methodology is convenient for any $2D$ CNN, our experiments show the architecture has to be chosen with respect to the problem, in the trend of many others study the number of weights has to be as low as possible. The aggregation of the different architectures may allow to get a decision, improving the confidence of the




(a) Intensive orchard (b) Traditional orchard (c) Meadow

Fig. 6: Most active filters based on energy computation for the classification of three examples of agricultural crops represented by MS-STR: (top) energies associated to each filter response; (bottom) natures of the most used filters.

classification and also to produce more accurate results. By considering $2D$ convolutions on this kind of images, we can also benefit of a pre-trained model, e.g. trained on the ImageNet database on a similar classification problem. Such initialization of the weights of the CNN is less tractable for $1D$ studies as no large public dataset, at the scale of ImageNet, and pre-trained networks, are available. The analysis of the trained filters shows the learned weights are linked to the temporal, spatial and spatio-temporal information. Then, the method can be applied in any other classification problem to get hint on the most important type of information.

As first perspective, we plan to generalize the filter analysis at different levels of the CNN architecture. Besides, when considering weights pre-trained on ImageNet (with purely spatial data), the way the low level filters may be transferred to deal with spatio-temporal data, can be deeper investigated. Another perspective is to involve our system in larger classification problems by considering more classes and data. We can consider for example other applicative domains like video analysis and indexation, where spatio-temporal information may be related to the speed of shifts.

## Acknowledgments

The authors thank the French ANR (Agence Nationale de la Recherche) for supporting this work under Grant ANR-17-CE23-0015.

## References

1. L. Andres, W. Salas and D. Skole, Fourier analysis of multi-temporal AVHRR data applied to a land cover classification, *International Journal of Remote Sensing* **15**(5)

(1994) 1115–1121.
2. A. Bagnall, J. Lines, A. Bostrom, J. Large and E. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, *Data Mining and Knowledge Discovery* **31**(3) (2017) 606–660.
3. L. Bruzzone and D. Prieto, Automatic analysis of the difference image for unsupervised change detection, *IEEE Transactions on Geoscience and Remote Sensing* **38**(3) (2000) 1171–1182.
4. M. Chelali, C. Kurtz, A. Puissant and N. Vincent, Urban land cover analysis from satellite image time series based on temporal stability, in *JURSE, Procs.* (2019) pp. 1–4.
5. P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys and E. Lambin, Digital change detection methods in ecosystem monitoring: A review, *International Journal of Remote Sensing* (2004) 1565–1596.
6. N. Di Mauro, A. Vergari, T. M. A. Basile, F. G. Ventola and F. Esposito, End-to-end learning of deep spatio-temporal representations for satellite image time series classification, in *DC@PKDD/ECML, Procs.* (2017) pp. 1–8.
7. L. Grady, Multilabel random walker image segmentation using prior models, in *CVPR, Procs.* (2005) pp. 763–770.
8. B. Huang, K. Lu, N. Audebert, A. Khalel, Y. Tarabalka, J. Malof and A. Boulch, Large-scale semantic classification: Outcome of the first year of inria aerial image labeling benchmark, in *IGARSS, Procs.* (2018) pp. 6947–6950.
9. F. Iandola, M. Moskewicz, K. Ashraf, S. Han, W. Dally and K. Keutzer, Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size, *Computing Research Repository* **abs/1602.07360** (2016).
10. 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* **14**(10) (2017) 1685–1689.
11. 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* **9**(1) (2017) 95–108.
12. H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar and P. Muller, Deep learning for time series classification: A review, *Data Mining and Knowledge Discovery* **33**(4) (2019) 917–963.
13. J. R. Jensen, Urban change detection mapping using Landsat digital data, *Cartography and Geographic Information Science* **8**(21) (1981) 127–147.
14. R. Johnson and E. Kasischke, Change vector analysis: A technique for the multispectral monitoring of land cover and condition, *International Journal of Remote Sensing* **19**(16) (1998) 411–426.
15. A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in *NIPS, Procs.* (2012) pp. 1106–1114.
16. C. Pelletier, G. Webb and F. Petitjean, Temporal convolutional neural network for the classification of satellite image time series, *Remote Sensing* **11**(5) (2019) 523–534.
17. F. Petitjean, J. Inglada and P. Gançarski, Satellite image time series analysis under time warping, *IEEE Transactions on Geoscience and Remote Sensing* **50**(8) (2012) 3081–3095.
18. F. Petitjean, C. Kurtz, N. Passat and P. Gançarski, Spatio-temporal reasoning for the classification of satellite image time series, *Pattern Recognition Letters* **33**(13) (2012) 1805–1815.
19. P. Ravikumar and V. S. Devi, Weighted feature-based classification of time series data, in *CIDM, Procs.* (2014) pp. 222–228.

20. M. Russwurm and M. Korner, Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images, in *EarthVision@CVPR, Procs.* (2017) pp. 1496–1504.
21. C. Senf, P. Leitao, D. Pflugmacher, S. Van der Linden and P. Hostert, Mapping land cover in complex mediterranean landscapes using landsat: Improved classification accuracies from integrating multi-seasonal and synthetic imagery, *Remote Sensing of Environment* **156** (2015) 527–536.
22. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *ICLR, Procs.* (2015)
23. D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in *ICCV, Procs.* (2015) pp. 4489–4497.
24. J. Verbesselt, R. Hyndman, G. Newnham and D. Culvenor, Detecting trend and seasonal changes in satellite image time series, *Remote Sensing of Environment* **114**(1) (2010) 106–115.

---

**Biographical Sketch**

**Mohamed Chelali** obtained the MSc in Computer Science from the Université de Paris, France, in 2018. He is currently working toward the Ph.D. degree at Université de Paris, France. His research interests include image processing and analysis applied on remote sensing.

**Camille Kurtz** obtained the MSc and PhD in Computer Science from the Université de Strasbourg, France, in 2009 and 2012. He was a post-doctoral fellow at Stanford University, CA, USA, between 2012 and 2013. He is now an Associate Professor at Université de Paris, France and a member of the SIP research group (Systèmes Intelligents de Perception) at LIPADE Lab. His scientific interests include image analysis, computer vision, medical imaging and remote sensing.

**Anne Puissant** received her Ph.D. degree in 2003 in Geography and Remote Sensing from the University Louis Pasteur, Strasbourg, France. She is currently Full Professor in the Geography Department at the University of Strasbourg. Her research topics are focused on (1) the utility of earth observation data to improve the knowledge of landscapes and to manage their state and dynamics and (2) the spatial analysis of natural or anthropic processes.

**Nicole Vincent** is Full Professor at Université de Paris, France, in the SIP research group (Systèmes Intelligents de Perception) at LIPADE Lab. As a former student from École Normale Supérieure, she received a Ph.D. in Computer Science in 1988 from INSA Lyon. Nicole Vincent has been involved in numerous projects in pattern recognition, signal and image processing and video analysis with a particular focus in the medical, remote sensing and document image analysis domains.