Image time series classification based on a planar spatio-temporal data representation

Mohamed Chelali[1], Camille Kurtz[1], Anne Puissant[2] and Nicole Vincent[1]
[1]*LIPADE, Université de Paris, Paris, France*
[2]*LIVE, Université de Strasbourg, Strasbourg, France*
firstname.lastname@{u-paris.fr, unistra.fr}

Keywords: Satellite Image Time series, spatio-temporal features, space-filling curves, Convolutional Neural Networks.

Abstract: Image time series such as MRI functional sequences or Satellite Image Time Series (SITS) provide valuable information for the automatic analysis of complex patterns through time. A major issue when analyzing such data is to consider at the same time their temporal and spatial dimensions. In this article we present a novel data representation that makes image times series compatible with classical deep learning model, such as Convolutional Neural Networks (CNN). The proposed approach is based on a novel planar representation of image time series that converts $2D+t$ data as $2D$ images without loosing too much spatial or temporal information. Doing so, CNN can learn at the same time the parameters of $2D$ filters involving temporal and spatial knowledge. Preliminary results in the remote sensing domain highlight the ability of our approach to discriminate complex agricultural land-cover classes from a SITS.

1 INTRODUCTION

Image time series are daily produced by various sensors such as MRI (functional imaging), satellites, drones or classical cameras observing particular land-cover classes leading to a large amount of images ($2D+t$). In the context of Earth observation, new constellations of satellites acquire images with a high spatial, spectral and temporal resolution around over the world. For example, Sentinel-2 produces optical Satellite Image Time Series (SITS) with a revisit time of 5 days and a spatial resolution of 10 – 20 meters.

Among relevant applications of SITS, we can mention the mapping of land cover (e.g. agricultural zones, urban areas) and the identification of land use changes (e.g. urbanization, deforestation). The growing availability of such temporal data makes it possible to produce and update accurate land-cover maps of a territory (Inglada et al., 2017). In order to efficiently handle the huge amount of data produced by these new sensors, adapted methods for SITS analysis have to be developed. Such methods should allow the end-user to obtain satisfactory results, with minimal time, and minimal effort.

A major issue when analyzing image time series is to consider simultaneously the temporal and the spatial dimensions of the $2D+t$ data-cube. Taking these two aspects into account at the same time can, for example, make it easier to discriminate between different complex agricultural land cover classes (e.g. orchards, meadows) from SITS. This article focuses on this specific problem. To deal with this issue, we define a novel spatio-temporal representation of image time series that makes it possible to consider classical deep learning framework (initially proposed for $2D$ images) for their analysis. Our main contribution is the proposal of a strategy to represent $2D+t$ data as $2D$ images without loosing too much spatial or temporal information. Doing so, deep Convolutional Neural Networks (CNN) can learn $2D$ filters involving at the same time temporal and spatial information. Here we do not aim to produce temporal land-cover maps or to study land use changes but our objective is to map complex land-cover classes prone to confusions when a single image is used.

This article is organized as follows. Section 2 recalls some existing methods for SITS analysis. Section 3 introduces our spatio-temporal representation for CNN based SITS analysis. Section 4 describes the experiments related to the classification of agricultural crops. Section 5 provides concluding remarks.

2 RELATED WORKS

SITS allow the observation and the analysis of land phenomena with a broad range of applications such as the study of land-cover or even the mapping

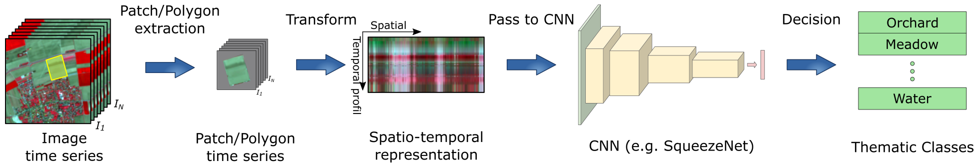


Figure 1: Flowchart of our method for SITS classification based on a planar spatio-temporal data representation.

of damage following a disaster. These changes may be of different types, origins and duration.

Pioneer methods for analyzing SITS operated on single images or stacks of images. On each image, the 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. Bi-temporal analysis, can locate and study changes occurring between two observations (Bruzzone and Prieto, 2000).

Another category of approaches were directly designed to deal with image time series. Most of them are based on multi-date classification approaches such as radiometric trajectory analysis (Verbesselt et al., 2010). Such approaches exploit the notion that land cover can vary through time (e.g. because of seasons, vegetation evolution (Senf et al., 2015)), and they take into account the order of measurements by using dedicated time series analysis methods (Bagnall et al., 2017). Every pixel is viewed as a temporally ordered (and aligned) series of measurements, and the changes of the measurements through time are analyzed to find (temporal) patterns.

Concerning the type of features, "frequency-domain" approaches include spectral analysis, wavelet analysis (Andres et al., 1994) while "time-domain" approaches involve auto-correlation and cross-correlation analysis. Concerning the classification method, the classical way is to measure similarity between any incoming sample and the training set; and assign the label of the most similar class using e.g. the Euclidean distance based on a nearest neighbor algorithm or the Dynamic Time Wrapping method (Petitjean et al., 2012a). Some methods first propose a new representation of the SITS into a new space to extract more discriminative "hand-crafted" features (Petitjean et al., 2012b; Chelali et al., 2019) and the classification is achieved in this new enriched space.

More recently deep learning approaches have also been considered to classify remote sensing images and generate land-cover maps. In many works, convolutional neural networks (CNN) are considered, generally dealing with the spatial domain of the data by applying $2D$ convolutions (Huang et al., 2018). When dealing with image time series, convolutions are often applied in the temporal domain (Pelletier et al., 2019). Another type of deep architecture that is designed for temporal data is recurrent neural network (RNN) like Long-Short Term Memory (LSTM), used successfully in (Ienco et al., 2017). In this context, deep learning approaches outperform traditional classifications algorithm like Random Forest (Ismail Fawaz et al., 2019), but as a limit, they do not directly take into account the spatial dimension of the data as they consider pixels in an independent way. Some attempts have been realized to consider both the temporal and the spatial dimensions of the $2D+t$ cube (Di Mauro et al., 2017). A common strategy is to train two models (one for spatial dimension and one for temporal dimension), and then to fuse their results at the decision level. In the domain of video analysis, spatio-temporal features are learned using deep $3D$ CNN (Tran et al., 2015) but such strategy requires the learning of an important number of parameters.

In this paper, our strategy is to classify a SITS using a classical $2D$ CNN model but we propose a new representation of image time series that embed simultaneously the temporal and the spatial dimensions of the data. We propose several representations based on various strategies, the orderings of pixels being different. The CNN learns with $2D$ convolutions temporal and spatial information at the same time.

3 PROPOSED APPROACH

This section presents our method dedicated to the classification of image time series based on a planar spatio-temporal data representation. After providing an overview of the global process, we will detail the different steps involved in the method.

3.1 Overview of the process

The proposed method is based on the use of a classical deep neural network architecture. But the input has not a $3D$ structure (Tran et al., 2015) nor a $1D$ structure (Pelletier et al., 2019) as this is often the case for the state-of-the-art methods studying the time series associated with each pixel. In our case, we propose to consider the pixels of a region of interest (e.g. an im-

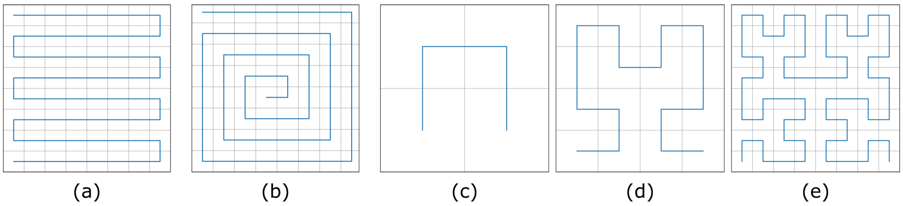


Figure 2: Illustrations of the different curves (in blue) covering a $2D$ space (in black, a grid of pixels); (a) Snake curve; (b) Spiral curve; (c,d,e) The three first orders of the Hilbert curve.

age patch or a polygon) as a whole and first to apply a transformation of this $2D+t$ data providing a (planar) $2D$ structure containing all the spatio-temporal data. This corresponds to the left part of the flowchart presented in Figure 1. Such a structure is then transferred as the input of a classical neural network to achieve classification. The network can be trained in order to learn the labels from the spatial as well as temporal information contained in the data. The right part of the flowchart depicted in Figure 1 illustrates this process.

3.2 Planar data representation: From $2D+t$ to $2D$

In order to decrease the complexity of the data structure, we propose to transform the spatial representation of the pixels in a $1D$ structure. Initially, a pixel is defined by its position (a couple of integers) in the image with height $\mathbb{H}$ and width $\mathbb{W}$. Now, it will be defined by only one integer given by an index specifying the position of the pixel in a path (i.e. a *string*) covering the region of interest. The function $\mathfrak{R}$

$$\begin{aligned} \mathfrak{R} : [1,\mathbb{W}] \times [1,\mathbb{H}] \rightarrow & \quad [1,\mathbb{W} \times \mathbb{H}] \\ (x,y) \mapsto & \quad i = \mathfrak{R}(x,y) \end{aligned}$$

associates to a pixel of coordinates (x,y) its position i in a one-dimensional space.

What is important in the plan is the notion of neighborhood. A pixel has usually 8 or 4 neighbors according to the topology that is considered. In a $1D$ string each element has only 2 nearest neighbors. Then, of course, by transforming a $2D$ space in a $1D$ space the spatial information will be diminished, but the objective is to keep the most representative information during the transform.

When a particular transform is chosen (some examples will be proposed hereinafter), it will be applied in the same way to all the N images (or for a particular region of interest) of the series. So, we get N strings that will be considered as the rows of a new image. The new image height is equal to the number N of the images in the SITS and its width is equal to the number of pixels of the region we want to represent. Such new image constitutes then a $2D$ spatio-temporal representation of a $2D+t$ image time series.

In order to keep some significant neighbors in this novel representation, the problem is then to fill a $2D$ discrete space with a discrete curve. Following the pixels along the curve, all the pixels of the region will be numbered only once and, by construction, two adjacent pixels in the curve are neighboring pixels in the plan. In the literature, many methods were proposed to achieve such a transformation but the aim is to consider statistically representative neighbors without any bias due to the path chosen in the plan. We have compared experimentally several strategies:

- the first representation is the most naive one among the others, noted $\mathfrak{R}_{snake}$. The space is filled by a simple curve which scans the image, line by line, as a snake (Figure 2 (a)). Lines are linked smartly so the spatial neighborhood information are preserved: odd lines ends are linked with heads of even ones, and *vice versa*. The pixels are finally numbered according to the curve.
- the second representation is based on Archimedean spiral, noted $\mathfrak{R}_{spiral}$. The pixel grid is associated with a spiral curve that fills a square (Figure 2 (b)). The curve starts from a center point $(0,0)$ of a square and its right neighbor then it revolves around. The construction of this curve is done by fixing two variables that indicate the next curve point, $(x+dx, y+dy)$. dx, dy are initialized to 0 and 1 respectively. The angular points are those verifying $x = y$, $x = -y$ and $y > 0$, $x-1 = -y$ and $x > 0$. The curve has to go to the right, to the left, to the bottom or to the top according to the directions of (dx,dy). The (dx,dy) values are successively $(0,1),(-1,0),(0,-1)$ and finally $(1,0)$.
- the third representation is based on space-filling curves, noted $\mathfrak{R}_{Hilbert}$. Our choice is the Hilbert curve which is a fractal space-filling curve (Butz, 1971) and it fills a square ($2D$ space). To define this curve, a recurrent process is applied starting from a square domain, the domain being divided into four equal squares. The four small squares are linked in such a way that “two parts with a common edge have two consecutive indexes”. This rule is applied recursively on squares with a

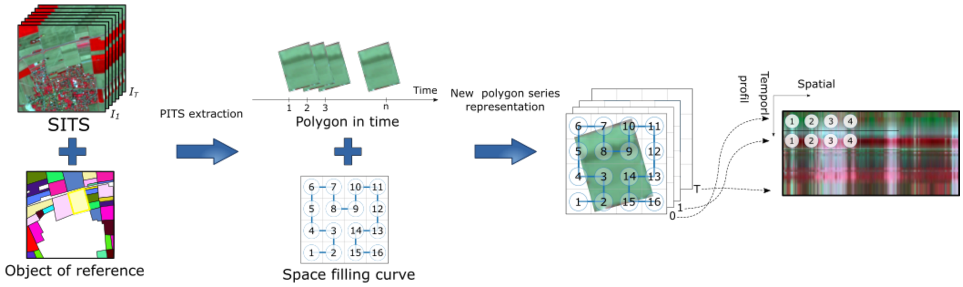


Figure 3: Polygon Image Time Series (PITS) representation based on the Hilbert curve.

width being a power of 2. The order of pixels is finally given by the Hilbert curve. The main interest of this kind of curve is the preservation of the spatial neighborhood relation of successive points on the curve. Figure 2 (c–e) illustrates the three first orders of Hilbert curves.

By applying the process to the N images (or to a specific region of interest) of the SITS, we obtain N rows of length equals to the number N_r of pixels in the region. These rows are used to fill a matrix and a new SITS image representation is obtained with N rows and N_r columns. Now this new image can also be interpreted in terms of columns. Each column is associated with a pixel and its time series in the SITS, a temporal pixel $p = \{< p_t(x,y) > | t = 1 \dots N\}$ is contained in the column of the new image. Figure 3 illustrates how the new representation is built.

3.3 CNN model (architecture)

Convolutional Neural Networks are used in most methods belonging to the family of deep learning algorithms. CNN are composed, in the left part, of layers of neurons computing convolutions of the previous layer outputs. The neurons of each layer are activated by non-linear functions which allow the extraction of high order features of the input. There is 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. In the final right part of the network, to solve classification problems we generally find a fully connected layer that provides a probability vector, coupled to a softmax function to predict a class label.

In our approach, we consider the SqueezeNet model (Iandola et al., 2016). This model is a rather small network and has few parameters to be fixed. In our case, this is an interesting model since it is adapted to our applicative context and our dataset (small size of training examples). This CNN leads to the same accuracy level as AlexNet model, when evaluated on the ImageNet dataset.

4 EXPERIMENTAL STUDY

The proposed approach has been evaluated in a remote sensing application, namely the classification of agricultural crop fields from SITS. Our objective is to separate some agricultural thematic classes (e.g. traditional vs. intensive orchards). The visual appearance of these agricultural parcels is heterogeneous because orchards are the subject of many agricultural practices, depending on the season, and their automatic identification remains a complex task. In order to differentiate these two classes, spatio-temporal features can carry useful information to better discriminate the agricultural practices.

4.1 Data presentation

The data used in experimental study are optical SITS, sensed by the Sentinel-2 satellite (East of France). The acquired data have been corrected and orthorectified by the French Theia program to be radiometrically comparable. The images are distributed with their associated cloud masks. A pre-processing was applied on the images with a linear interpolation on masked pixels to guarantee same size for all images.

We dispose of a SITS of $N = 50$ images sensed in 2017 over the same geographical area. For each image, only three bands are kept which are near-infrared (Nir), red (R) and green (G). All these bands have a spatial resolution of 10 meters.

In addition to the images, we dispose of reference data which is composed of the reference agricultural parcel delineations (in our context orchards) represented as vector polygons. These polygons are extracted from the French IGN RPG. In our case, polygons have been rasterized according to the spatial resolution of each image, leading to a new Polygon Image Time Series, noted PITS, that will be represented with the strategies presented in Section 3.

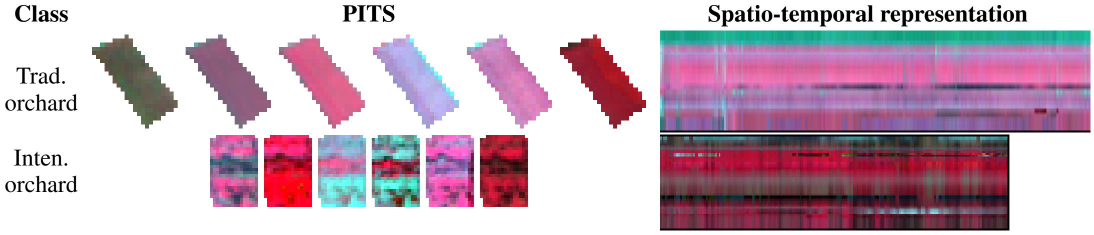


Figure 4: Example of PITS representing orchards; (left) Evolution of a traditional / intensive orchards; (right) Associated spatio-temporal representations (Hilbert strategy $\mathfrak{R}_{Hilbert}$).

The reference data used in our experiment are the semantic labels of these polygons (traditional or intensive orchards). Figure 4 presents an example of the temporal evolution of two orchards through the SITS. Finally, we dispose of 100 polygon per class. In order to get more annotated data, data augmentation (DA) technique is used by applying rotations with the angles: 45°, 90°, 135° and 180°.

4.2 Experimental protocol

We applied the proposed method to classify the two orchard classes (traditional vs. intensive). From an intuitive point of view, intensive orchards should have a more homogeneous texture in the spatial domain since the fruit trees are generally aligned which is not always the case in the traditional ones.

4.2.1 Data preparation

Firstly, the input data are prepared thanks to the proposed spatio-temporal representations of images. This is operated at the polygon level. Each PITS is processed in 3 different ways according to the $\mathfrak{R}_{snake,spiral,Hilbert}$ functions presented earlier. To highlight the interest of considering the spatial relation between pixels, we added (as a naive baseline) a random way to build the spatio-temporal representation of the PITS, noted $\mathfrak{R}_{random}$.

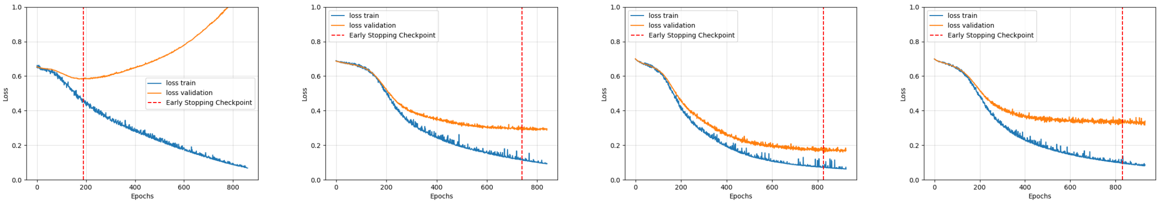
According to the CNN input size which is 224×224, we adapt our generated images to fit this size. For the temporal dimension (Y axis), we propose two strategies. The first one is to center the original information from the N input images vertically ($N = 50$). The remaining top and bottom lines are fixed to zero value. For the second, we choose to process a 224 long time series, that is to fill all the remaining vertical space. In order to do this, we have applied a linear interpolation 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, we affect the temporal information of the first date in the SITS. For the last dates, we affect the last temporal information in the SITS. For the other unknown date values, we compute them by applying a linear function that considers two consecutive available dates (taken from the set of $N = 50$ images of the SITS). Finally, we got 224 dates that complete the height of the image. These two strategies (with original dates or with temporal interpolation) will be evaluated separately.

For the spatial dimension (X axis), as the size of the polygons is rarely equal to 224, we adopted the following strategy. For polygons where pixel number is less than 224, we repeat the sequence. For those composed of more than 224 pixels, we slice the new representation into different images with 224 columns, leading potentially to a higher number of data to be classified than the number of polygons.

The data images have been normalized based on the maximum and the minimum values of the dataset. In our case, we limited the values with 2% (or 98%) percentile, as proposed in (Pelletier et al., 2019).

4.2.2 Learning and validation protocol

To validate these experiments, a 5-fold cross validation strategy is employed. In each case, the dataset is randomly split into 3 sets, at the polygon level, and we repeated 5 times the process. The size of these sets is 60%, 20% and 20% of all available data representing respectively the training, validation and test sets. In each experiment, the same folds are considered in order to make the results more comparable. The model is trained and evaluated 5 times according to each split and for one split we consider the system that gives the best result on the validation set. Note that the decision from the classifier output is taken at the polygon level. We explained before that for large polygons (which have more than 224 pixels), we build several different images in our process (see Section 4.2.1). Then several images are associated with a single polygon. To take a decision in this case, the model returns the probabilities of classes for each im-



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

Figure 5: Loss curves related to the training of our model with the different spatio-temporal representations; (a) Random strategy $\Re_{random}$; (b) Snake strategy $\Re_{snake}$; (c) Spiral strategy $\Re_{spiral}$; and (d) Hilbert strategy $\Re_{Hilbert}$.

age associated with the polygon. Then, we average these probabilities with respect to each class and we affect to the polygon the label of the class with the highest probability. We report the overall accuracy that is the average value of the results on the test sets according to the 5 splits and the standard-deviation.

We train the model using *Adam* as optimizer with a learning rate of 10^{-6} and default values of the other parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$) with batch size of 8. We limit the number of epochs to 2000, following an early stopping technique with a patience number of 100. As the size of the available dataset is limited, we train the network using two strategies: (1) from scratch and (2) with fine-tuning (the model was pre-trained on ImageNet). We have also proceeded with data augmentation.

4.3 Results and discussions

The proposed spatio-temporal representations of the PITS have been used to feed the CNN. We also used the random $\Re_{random}$ ordering of pixels in order to evaluate the importance of spatial information. Two successive pixels in the $1D$ representation are neighbors in the $2D$ space is a property of the different space filling curves we have considered. For visualization purpose, Figure 4 illustrates two PITS with their resulting spatio-temporal representations, here based on the $\Re_{Hilbert}$ strategy.

The CNN model was trained accordingly to the learning protocol, with and without fine tuning. We also evaluated the impact of considering the original temporal dates or applying an interpolation to fit the 224×224 image input size required by SqueezeNet.

Figure 5 illustrates the resulting loss curves when SqueezeNet is trained from scratch (following an early stopping technique) with images related respectively to the $\Re_{random}$, $\Re_{snake}$, $\Re_{spiral}$ and $\Re_{Hilbert}$ strategies, here with original dates. From these curves, we notice that the worst (highest) loss values are obtained with the $\Re_{random}$ strategy as expected. Also the loss curve of the $\Re_{random}$ strategy starts stabilizing near 200 epochs compared to others which start stabilizing from about 600 epochs. Intuitively, this means that the $\Re_{random}$ strategy does not provide a good representation of PITS with a good ability to generalize when training. Other representations allow to make a rather good fit. We can also see the best learning curves are obtained in (c), using the $\Re_{spiral}$, with the best results on the validation set. This ranking is not preserved at the global test set level.

Table 1 reports the classification results (overall accuracy) obtained with our spatio-temporal representations (with original dates). We notice that $\Re_{random}$ always provides the lowest scores compared to the other representations, with and without DA, or with and without fine tuning. This is quite expected as the discrimination between traditional and intensive orchards is relying on spatial information and this information is partly preserved with space filling curves providing spatial information in addition to temporal information. From Table 1, we also notice that with DA, all scores are slightly increased, and the best scores have been obtained by combining DA and the fine tuning strategy. Finally, here the best representations oscillate between $\Re_{snake}$, $\Re_{spiral}$ and $\Re_{Hilbert}$.

As comparative study, we compared our results to the ones obtained with the TempCNN method dedicated to the classification of time series, proposed in (Pelletier et al., 2019). This approach relies on the use of a CNN classifier, where convolutions are applied in the temporal domain ($1D$ convolutions). The filter sizes are fixed following the criterion given in (Pelletier et al., 2019): with a kernel size of 5 when considering the original dates, and 11 when considering the interpolated dates. For comparison purpose, we trained and validated the TempCNN model using the same validation protocol. Note that the TempCNN model is proposed with different architectures (depths), leading to different number of filters.

Table 2 reports the TempCNN results. Best scores were obtained with 256 filters. The obtained scores suggest that the results obtained with TempCNN outperform the ones obtained with our method when we train from scratch. However, when considering fine-tuning from the pre-trained model on ImageNet, we obtain better scores. This highlights, for our applicative context, the benefit of considering a classical $2D$ CNN model for classifying $2D+t$ images combined with our spatio-temporal representations.

Table 1: Classification results (overall accuracy – OA and standard deviation – STD) obtained with our spatio-temporal representations (with original date images); (first/second rows) Without/With data augmentation.

		From scratch		Fine tuning	
	Rep.	**OA**	**STD**	**OA**	**STD**
w/o DA	$\mathfrak{R}_{random}$	71.50	7.17	81.00	8.15
	$\mathfrak{R}_{snake}$	78.00	4.30	90.50	7.96
	$\mathfrak{R}_{spiral}$	76.00	8.74	**92.00**	**3.31**
	$\mathfrak{R}_{Hilbert}$	**79.00**	**5.61**	91.00	2.00
with DA	$\mathfrak{R}_{random}$	80.50	3.67	87.00	4.58
	$\mathfrak{R}_{snake}$	83.50	7.00	**93.50**	**2.54**
	$\mathfrak{R}_{spiral}$	**84.50**	**5.33**	93.00	1.87
	$\mathfrak{R}_{Hilbert}$	81.50	6.44	91.00	2.54

Table 2: Classification results (overall accuracy – OA and standard deviation – STD) with the TempCNN architectures (with original dates and kernel size of 5).

Nb filters	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
OA	78.81	77.38	81.66	78.45	**85.37**	81.73	84.80
STD	6.08	6.51	4.59	4.79	**3.44**	5.75	6.48

Table 3: Classification results (overall accuracy – OA and standard deviation – STD) obtained with our spatio-temporal representations (with temporal interpolation); (first/second rows) Without/With data augmentation.

		From scratch		Fine tuning	
	Rep.	**OA**	**STD**	**OA**	**STD**
w/o DA	$\mathfrak{R}_{random}$	84.00	9.02	87.00	4.30
	$\mathfrak{R}_{snake}$	85.00	4.18	**92.50**	**3.16**
	$\mathfrak{R}_{spiral}$	85.00	3.53	91.00	2.54
	$\mathfrak{R}_{Hilbert}$	**89.00**	**3.39**	91.00	2.54
with DA	$\mathfrak{R}_{random}$	82.00	8.71	83.50	4.35
	$\mathfrak{R}_{snake}$	86.50	5.38	90.50	1.87
	$\mathfrak{R}_{spiral}$	86.50	3.00	**91.50**	**3.74**
	$\mathfrak{R}_{Hilbert}$	**92.50**	**1.58**	89.00	3.39

Table 4: Obtained results (overall accuracy – OA and standard deviation – STD) with the TempCNN architectures (with temporal interpolation and kernel size of 11).

Nb filters	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
OA	78.96	81.40	83.96	81.86	85.93	84.23	**87.21**
STD	7.34	6.32	7.14	5.18	8.03	6.23	**8.28**

Table 3 presents the classification results when considering the temporal interpolation strategy. We notice that with more temporal information, the overall scores are increased compared to the case with less temporal information (images with original dates) reported in Table 1. This can be explained by the non-regular distribution of the original dates. Whereas, with the interpolation, we obtain a temporal information with an equal regularity to obtain 224 dates and also due to actual monotonous behavior between the consecutive dates used for the interpolation. We observe again that the $\mathfrak{R}_{random}$ strategy leads to the worst scores. This confirms that spatial information is important and not just temporal one. We see also that DA increases slightly the scores in case of from scratch learning but is not able to improve the results in case of fine tuning. In this experiment, the $\mathfrak{R}_{Hilbert}$ strategy leads to the representation that provides the best scores when we train from scratch (with and without DA). But when we fine tune, the best representations oscillate between $\mathfrak{R}_{snake}$ and $\mathfrak{R}_{spiral}$.

The obtained results with TempCNN when considering the temporal interpolation strategy are listed in Table 4. Initial scores range in the same interval as our method when we train from scratch. But with DA or/and fine-tuning, our scores are higher.

5 CONCLUSION

In this paper we present a new strategy for transforming an image time series to a planar spatio-temporal representation, reducing the complexity of an image time series structure (from $2D+t$ to $2D$) while maintaining (partially) the spatial and temporal relationships of pixels. These representations are used to feed a classical CNN in order to perform a classification. $2D$ convolutions can then lead to an extraction of spatio-temporal features. Compared to $1D$ approaches dedicated to time series, we have a lower number of annotated data, but this is compensated by data augmentation. By considering $2D$ convolutions, we can also benefit of a pre-trained model on ImageNet. Such initialization of the weights of the CNN is less tractable for $1D$ studies as no ImageNet like dataset is available.

The proposed approach has been evaluated in remote sensing for the classification of agricultural crop fields from SITS. In our experimentation, we study the impact of the spatio-temporal transformation using different space filling curves. The obtained results reflect the usefulness and the impact of considering both spatial and temporal information. From our thematical study, we observe that the classification scores are higher when considering spatio-temporal representations with more temporal information (using the temporal interpolation) than those who have less, even if built from the same initial data. It is then more important to have many data along the temporal domain than the way the 2*D* plan is filled with curves.

In our comparative study, we notice that the TempCNN method (Pelletier et al., 2019) is applied at the pixel level while our approach is applied at the polygon level. This means that for TempCNN there are more training samples compared to our method where the pixels of a polygon are all summed up in a spatio-temporal image. Despite the low number of data available, the accuracy increase made possible by our process is up to 8% based on the original data and 5% on the interpolated data.

In future works, the ordering of the pixels will be more precisely studied, till now we have ordered the pixels of a square region where the polygon is included but we have to define an order adapted to the geometry of the polygon itself. We will also increase the number of instances of orchards and also apply the same approach to problems involving a larger number of classes in order to generate land-cover maps.

ACKNOWLEDGEMENTS

The French ANR supported this work under Grant ANR-17-CE23-0015.

REFERENCES

Andres, L., Salas, W., and Skole, D. (1994). Fourier analysis of multi-temporal AVHRR data applied to a land cover classification. *Int. J. Remote Sens.*, 15(5):1115–1121.

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *DMKD*, 31(3):606–660.

Bruzzone, L. and Prieto, D. (2000). Automatic analysis of the difference image for unsupervised change detection. *IEEE Transactions on Geoscience and Remote Sens.*, 38(3):1171–1182.

Butz, A. (1971). Alternative algorithm for Hilbert's space-filling curve. *IEEE Trans. on Computers*, 20(4):424–426.

Chelali, M., Kurtz, C., Puissant, A., and Vincent, N. (2019). Urban land cover analysis from satellite image time series based on temporal stability. In *JURSE, Procs.*, pages 1–4.

Di Mauro, N., Vergari, A., Basile, T. M. A., Ventola, F. G., and Esposito, F. (2017). End-to-end learning of deep spatio-temporal representations for satellite image time series classification. In *DC@PKDD/ECML, Procs.*, pages 1–8.

Huang, B., Lu, K., Audebert, N., Khalel, A., Tarabalka, Y., Malof, J., and Boulch, A. (2018). Large-scale semantic classification: Outcome of the first year of inria aerial image labeling benchmark. In *IGARSS, Procs.*, pages 6947–6950.

Iandola, F., Moskewicz, M., Ashraf, K., Han, S., Dally, W., and Keutzer, K. (2016). Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *CoRR*, abs/1602.07360.

Ienco, D., Gaetano, R., Dupaquier, C., and Maurel, P. (2017). Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geosci. Remote Sens. Lett.*, 14(10):1685–1689.

Inglada, J., Vincent, A., Arias, M., Tardy, B., Morin, D., and Rodes, I. (2017). Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sens.*, 9(1):95–108.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. (2019). Deep learning for time series classification: A review. *DMKD*, 33(4):917–963.

Pelletier, C., Webb, G., and Petitjean, F. (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sens.*, 11(5):523–534.

Petitjean, F., Inglada, J., and Gançarski, P. (2012a). Satellite image time series analysis under time warping. *IEEE Transactions on Geoscience and Remote Sens.*, 50(8):3081–3095.

Petitjean, F., Kurtz, C., Passat, N., and Gançarski, P. (2012b). Spatio-temporal reasoning for the classification of satellite image time series. *PRL*, 33(13):1805–1815.

Senf, C., Leitao, P., Pflugmacher, D., Van der Linden, S., and Hostert, P. (2015). Mapping land cover in complex mediterranean landscapes using landsat: Improved classification accuracies from integrating multi-seasonal and synthetic imagery. *Remote Sens. Environ.*, 156:527–536.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *ICCV, Procs.*, pages 4489–4497.

Verbesselt, J., Hyndman, R., Newnham, G., and Culvenor, D. (2010). Detecting trend and seasonal changes in satellite image time series. *Remote Sens. Environ.*, 114(1):106–115.