From pixels to Random Walk based segments for image time series deep classification *

Mohamed Chelali[1 [0000−0002−0173−7368]], Camille Kurtz[1 [0000−0001−9254−7537]], Anne Puissant[2 [0000−0002−3240−9244]], and Nicole Vincent[1 [0000−0002−0151−0622]]

[1] Université de Paris, LIPADE, Paris, France.
{firstname.lastname}@u-paris.fr
[2] Université de Strasbourg, LIVE. Strasbourg, France.
{firstname.lastname}@unistra.fr

Abstract. 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 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 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 loosing 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 interest of this approach is highlighted on a remote sensing application for the classification and the mapping of complex agricultural crops.

Keywords: Satellite Image Time Series, spatio-temporal features, Random Walk, Convolutional Neural Networks.

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.

SITS 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

* The authors thank the French ANR for supporting this work under Grant ANR-17-CE23-0015.

of complex patterns related to applications related to land cover mapping (e.g. agricultural zones, urban areas) or 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, methods for SITS analysis are actually mainly based on temporal information [15] at the 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, urban vs. peri-urban areas). Note that here our objective is to map complex land-cover classes prone to confusions when a single date image is used.

This article focuses on the problem of spatio-temporal features 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 (proposed for the analysis of $2D$ images). Our main contribution is the proposal of a transformation to represent $2D + t$ data as $2D$ images without loosing 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.

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

2 Related works on SITS analysis

SITS allow the observation of the Earth surface. 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. Methods designed for bi-temporal analysis locate and study changes occurring between the two observations. These methods include image differencing [3], ratio-ing [13] or vector change analysis [14].

Another family of methods are directly dedicated to the analysis of image time series. Most of them are based on multi-date classification. Among them,

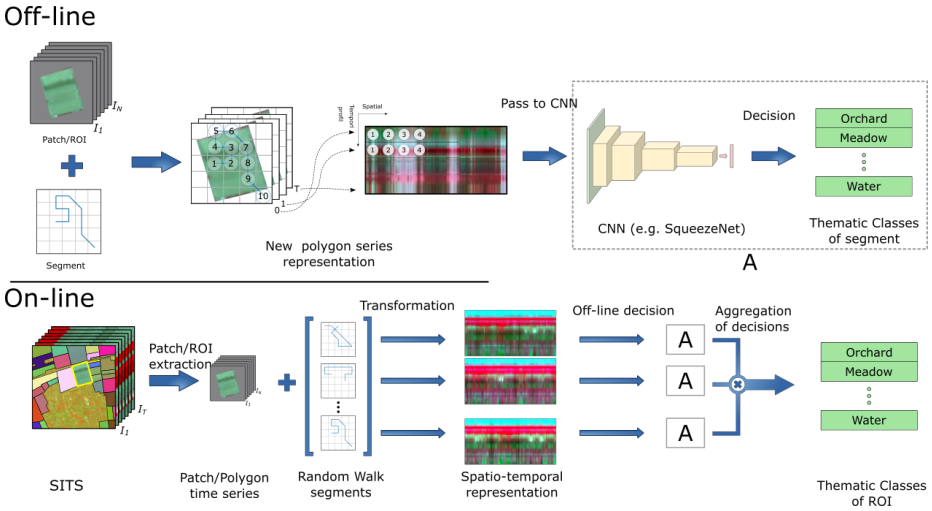

Fig. 1. Flowchart of our method for image time series deep classification based on a planar spatio-temporal data representation obtained from Random Walk based segments. (top) off-line (i.e. learning) phase and (bottom) on-line (i.e. testing) phase of the classification process.

we find radiometric trajectory analysis [22]. These methods exploit the evolution of land cover (e.g. seasons, vegetation evolution [20]), and take into account the chronology by using dedicated time series analysis methods [2]. Every pixel is considered as time ordered (and aligned) series of measurements, and the changes of the measurements 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 may 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, 17, 18]. 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 [16].

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 [15]. 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, 19]. 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 [6]. A common strategy is to train two models, one for spatial dimension and one for temporal dimension, then to fuse their results at the decision level. In video analysis, spatio-temporal features are learned directly using deep $3D$ convolutional networks [21] but such strategy requires the learning of an huge number of parameters.

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 with the use of $1D$ convolutions applied in the temporal domain [15] to classify temporal pixels (which is the current state of the art).

3 Proposed method

The proposed method aims to classify image time series from spatio-temporal features. The underlying strategy is to use a $2D$ input 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), 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 [15] or a $3D$ one [21] as we find in the state-of-the-art methods.

We start by transforming the original $2D + t$ data into planar entity containing spatio-temporal data built from $1D$ spatial segments over time. 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 used on-line. The network can be trained in order to learn the labels from the spatial, as well as 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, $0D$ 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 here 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 following these segments. In this way, the spatial representation complexity of the images is decreased, from $2D$ to $1D$ segments.

Next, segments characterized by temporal information leading to $2D$ 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.

With each of the N_p segments, we associate a $2D$ structure. In the abscissa X axis, is considered the index of the pixel in the segment (from the initial pixel) and in the ordinate Y axis is considered the evolution of the intensity of the pixels over the 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 $2D + t$ 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. These representations will be used as input of a learning process, the segments classes are the classes of the annotated input image they belong to.

3.2 Segment construction strategies

Two different strategies were considered to build segments:

- **Scanning strategy (scan).** Here we consider all the rows and columns 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 too short segments (this may correspond to segments extracted from the borders of the image. In this way, the pixels are considered only twice in the new representation, and each pixel has only 4 neighbors.

– **Random Walk (RW) based segment.** A Random Walk [7] is a mathematical process based on a random iterative system. 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.

3.3 CNN model (architecture)

Convolutional Neural Networks (CNN) refer to the family of deep learning algorithms. Systems are composed of two parts. The first one is designed to feature extraction, it has many neuron layers that compute the convolutions of 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 a fully connected layer that provides 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] but any other $2D$ CNN model can be used. SqueezeNet has interesting properties, few parameters, and 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 1×1 convolution filters followed by expand layer that contains a mix of 1×1 and 3×3 convolution filters. Also, its classifier is based on a global average pooling over feature maps, potentially decreasing the overfitting effect. We used the PYTORCH implementation of SqueezeNet[3]. The CNN model is trained with the $2D$ spatio-temporal representations 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, N_p is consequently a parameter of the method. With each segment is associated a

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

$2D$ 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. 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 a unique decision per image.

4 Experimental study

The experimental study is focused on a remote sensing application, the classification of agricultural crop fields from a SITS. The goal is to discriminate within agricultural thematic classes (e.g. traditional vs. intensive orchards). The automatic identification of these classes is a complex task since orchards are subject of 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, it contains $N = 50$ optical images sensed in 2017 over the same geographical area (East of France – tile 32ULU). Figure 2 displays the temporal distribution of the images of this SITS. The images have been corrected and orthorectified by the French Theia program[4] to be radiometrically comparable. We also dispose of the cloud, shadow and saturation masks associated with each image. A pre-processing 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.

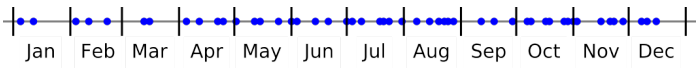

Fig. 2. Distribution of the images from the SITS (2017).

The used reference data are extracted from the (freely distributed) RPG[5], which is the agricultural parcel delineations (in our context orchards). Some examples of polygons are represented in Figure 1. These polygons have been

[4] `https://theia.cnes.fr/`
[5] `http://professionnels.ign.fr/rpg`

corrected 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).

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	100	3084	3000
Trad. orchards	100	3059	3000
Total	200	6143	6000

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. According to the ROI sizes, we set N_p to 30 for the RW strategy. For the *scan* strategy, the number of possible $1D$ segments depends on the ROI size. The average number of segments for the *scan* strategy is 487 ± 110. In the off-line part of Figure 1, we illustrate the transformation process of PITS with $RW(10)$. 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 224×224 $2D$ image from the segments, if the segments are less than 224, 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 (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. The second one 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 (begin 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 proposed in [15].

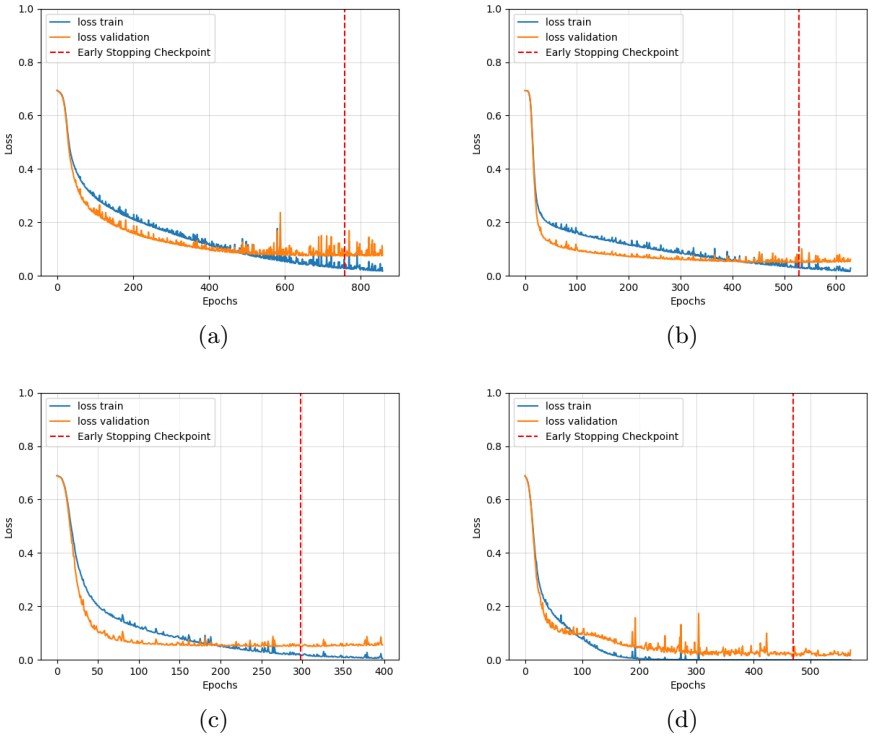

Fig. 3. Training phase loss curves, according to the length L of Random Walk paths: (a) $L = 10$; (b) $L = 50$; (c) $L = 100$ and (d) $L = 224$.

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 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).

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 and then fine-tuned with our data.

5 Results and discussion

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 dimensions of the ROIs. At segment level, Figure 3 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.

The on-line classification results (overall accuracy) with spatio-temporal representations (with original dates) are reported in Table 2. All the scores are in the same range except the *scan(10)*. 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). We notice that with fine-tuning, all the scores are increased, with *RW(224)* in the first position.

The obtained results are compared to those obtained with the TempCNN method [15][6]. TempCNN is dedicated to the classification of time series, where convolutions are applied in the temporal domain ($1D$ convolutions). The filter sizes are fixed following the criterion given in [15]: 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 data and validation protocol than the one used for our model. Note that the TempCNN model is proposed in [15] with different architectures (depths of the network), leading to different numbers of filters.

Table 3 reports the obtained results with the TempCNN method. Best scores were obtained with 256 filters. The best obtained score with our method (when we train from scratch) is slightly better than those obtained with TempCNN. However, when we use fine-tuning, we outperform them. 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 4 presents the classification results when considering the spatio-temporal images with the temporal interpolation strategy. We remark that all the scores

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

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

	From scratch		**Fine-tuning**	
Lengths of the segments	**OA**	**STD**	**OA**	**STD**
scan(10)	73.00	9.27	80.50	5.09
RW(10)	85.50	5.56	90.50	5.78
RW(50)	80.00	7.27	92.00	2.91
RW(100)	84.50	5.33	**94.00**	**2.00**
RW(224)	**86.00**	**5.61**	92.50	4.18

Table 3. Classification results (overall accuracy – OA and standard deviation – STD) with the different architectures of TempCNN [15] (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

are increased compared to those with less temporal information (Table 2). This is explained by the non-regular temporal distribution of the original images. So with the linear interpolation, we make the temporal distribution regular to obtain 224 dates. *scan(10)* is always less efficient than those that are based on *RW*. All the obtained scores are in the same rank with *RW(100)* in the first position with and without fine-tuning.

Table 5 reports scores when classifying with TempCNN [15] with more temporal data. The overall accuracy is slightly increased compared to the previous one (Table 3) and the best result is obtained with 1024 filters. The scores obtained with our method are higher (with and without fine-tuning) than with TempCNN.

Table 4. Classification results (overall accuracy – OA and standard deviation – STD) obtained with our spatio-temporal representations (with temporal interpolation).

	From scratch		**Fine-tuning**	
Lengths of the segments	**OA**	**STD**	**OA**	**STD**
scan(10)	78.50	6.44	83.00	1.87
RW(10)	90.00	4.18	**93.00**	**4.30**
RW(50)	90.50	1.87	**93.00**	**2.44**
RW(100)	**93.50**	**2.00**	**93.00**	**2.44**
RW(224)	91.50	1.22	91.00	2.00

Table 5. Obtained results (overall accuracy – OA and standard deviation – STD) with the different architectures of TempCNN [15] (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**

6 Conclusion

In this article, we present a new 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 loosing too much the spatial relationship of pixels and the temporal one. Then, these new representations images are used to feed a classical CNN to perform a classification. With the proposed representation, the applied $2D$ convolutions lead to a spatio-temporal features extraction. The trained filters have weights linked to the temporal evolution and others linked to spatial evolution, finally, the combination of both carry information on spatio-temporal evolution. 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.

References

1. Andres, L., Salas, W., Skole, D.: Fourier analysis of multi-temporal AVHRR data applied to a land cover classification. International Journal of Remote Sensing **15**(5), 1115–1121 (1994)
2. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **31**(3), 606–660 (2017)
3. Bruzzone, L., Prieto, D.: Automatic analysis of the difference image for unsupervised change detection. IEEE Transactions on Geoscience and Remote Sensing **38**(3), 1171–1182 (2000)
4. Chelali, M., Kurtz, C., Puissant, A., Vincent, N.: Urban land cover analysis from satellite image time series based on temporal stability. In: JURSE, Procs. pp. 1–4 (2019)
5. Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., Lambin, E.: Digital change detection methods in ecosystem monitoring: A review. International Journal of Remote Sensing pp. 1565–1596 (2004)
6. Di Mauro, N., Vergari, A., Basile, T.M.A., Ventola, F.G., Esposito, F.: End-to-end learning of deep spatio-temporal representations for satellite image time series classification. In: DC@PKDD/ECML, Procs. pp. 1–8 (2017)
7. Grady, L.: Multilabel random walker image segmentation using prior models. In: CVPR, Procs. pp. 763–770 (2005)

8. Huang, B., Lu, K., Audebert, N., Khalel, A., Tarabalka, Y., Malof, J., Boulch, A.: Large-scale semantic classification: Outcome of the first year of inria aerial image labeling benchmark. In: IGARSS, Procs. pp. 6947–6950 (2018)
9. Iandola, F., Moskewicz, M., Ashraf, K., Han, S., Dally, W., Keutzer, K.: Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. Computing Research Repository **abs/1602.07360** (2016)
10. Ienco, D., Gaetano, R., Dupaquier, C., Maurel, P.: Land cover classification via multitemporal spatial data by deep recurrent neural networks. IEEE Geoscience and Remote Sensing Letters **14**(10), 1685–1689 (2017)
11. Inglada, J., Vincent, A., Arias, M., Tardy, B., Morin, D., Rodes, I.: Operational high resolution land cover map production at the country scale using satellite image time series. Remote Sensing **9**(1), 95–108 (2017)
12. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.: Deep learning for time series classification: A review. Data Mining and Knowledge Discovery **33**(4), 917–963 (2019)
13. Jensen, J.R.: Urban change detection mapping using Landsat digital data. Cartography and Geographic Information Science **8**(21), 127–147 (1981)
14. Johnson, R., Kasischke, E.: Change vector analysis: A technique for the multispectral monitoring of land cover and condition. International Journal of Remote Sensing **19**(16), 411–426 (1998)
15. Pelletier, C., Webb, G., Petitjean, F.: Temporal convolutional neural network for the classification of satellite image time series. Remote Sensing **11**(5), 523–534 (2019)
16. Petitjean, F., Inglada, J., Gançarski, P.: Satellite image time series analysis under time warping. IEEE Transactions on Geoscience and Remote Sensing **50**(8), 3081–3095 (2012)
17. Petitjean, F., Kurtz, C., Passat, N., Gançarski, P.: Spatio-temporal reasoning for the classification of satellite image time series. Pattern Recognition Letters **33**(13), 1805–1815 (2012)
18. Ravikumar, P., Devi, V.S.: Weighted feature-based classification of time series data. In: CIDM, Procs. pp. 222–228 (2014)
19. Russwurm, M., Korner, M.: Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In: EarthVision@CVPR, Procs. pp. 1496–1504 (2017)
20. Senf, C., Leitao, P., Pflugmacher, D., Van der Linden, S., Hostert, P.: Mapping land cover in complex mediterranean landscapes using landsat: Improved classification accuracies from integrating multi-seasonal and synthetic imagery. Remote Sensing of Environment **156**, 527–536 (2015)
21. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV, Procs. pp. 4489–4497 (2015)
22. Verbesselt, J., Hyndman, R., Newnham, G., Culvenor, D.: Detecting trend and seasonal changes in satellite image time series. Remote Sensing of Environment **114**(1), 106–115 (2010)