Unsupervised Satellite Image Time Series Clustering using Object-Based Approaches and 3D Convolutional Autoencoder

Ekaterina Kalinicheva, Jérémie Sublime, Maria Trocan

remote sensing

MDPI

Article

Unsupervised Satellite Image Time Series Clustering Using Object-Based Approaches and 3D Convolutional Autoencoder

Ekaterina Kalinicheva [1,*], Jérémie Sublime [1,2] and Maria Trocan [1]

1 ISEP, DaSSIP Team-LISITE, 92130 Issy-Les-Moulineaux, France; jeremie.sublime@isep.fr or sublime@lipn.univ-paris13.fr (J.S.); maria.trocan@isep.fr (M.T.)
2 LIPN-CNRS UMR 7030, Université Paris 13, 93430 Villetaneuse, France
* Correspondence: ekaterina.kalinicheva@isep.fr; Tel.: +33-1-4954-5219

Received: 6 May 2020; Accepted: 31 May 2020; Published: 4 June 2020

Abstract: Nowadays, satellite image time series (SITS) analysis has become an indispensable part of many research projects as the quantity of freely available remote sensed data increases every day. However, with the growing image resolution, pixel-level SITS analysis approaches have been replaced by more efficient ones leveraging object-based data representations. Unfortunately, the segmentation of a full time series may be a complicated task as some objects undergo important variations from one image to another and can also appear and disappear. In this paper, we propose an algorithm that performs both segmentation and clustering of SITS. It is achieved by using a compressed SITS representation obtained with a multi-view 3D convolutional autoencoder. First, a unique segmentation map is computed for the whole SITS. Then, the extracted spatio-temporal objects are clustered using their encoded descriptors. The proposed approach was evaluated on two real-life datasets and outperformed the state-of-the-art methods.

Keywords: satellite image time series; unsupervised learning; clustering; segmentation; 3D convolutional network; autoencoder

1. Introduction

Satellite image time series analysis is used in many research areas these days. Contrary to the past decades, nowadays we dispose of a huge amount of time-spread geospatial data that provide us a full description of almost any area of interest in the world [1]. Exploiting satellite image time series (SITS) gives us better comprehension of a study area, its landscape, land cover, evolution and more comparing to a single image analysis [2]. While some applications demand SITS analysis in order to detect or monitor a specific event (constructions, droughts, deforestation, etc.) [3–6], others exploit SITS to perform a land cover analysis of the whole area and/or its eventual evolution [7]. For the second type of applications, the prior knowledge about temporal behavior of some classes (usually vegetation) is indispensable to make a correct classification map [8,9].

However, due to the variety of objects presented in the remote sensed images and in SITS in particular, few labeled data are available. For this reason, unsupervised approaches are becoming more and more popular for various projects. Most of the currently used unsupervised approaches for SITS clustering deploy pixel-wise analysis [10,11]. In these approaches, the pixels corresponding to the same geographical position on different images form temporal sequences that are further compared to each other and associated to different classes. Numerous studies have proven Dynamic Time Warping (DTW) algorithm [12] to be an efficient tool to compute the similarity measure between temporal sequences. The main idea of this approach is to non-lineary map one series to another by minimizing

the distance between them. Thus, the DTW distance matrix is computed for every point of the series and used as a similarity measure for a chosen clustering algorithm.

In general, DTW distance matrix has a high computational cost. To this end, the analysis of large datasets at pixel level may be extremely time-consuming and, hence, unreasonable. To deal with this issue, several object-based DTW clustering approaches have been proposed [13–15] to analyze the data both at temporal and spatial dimension. In these methods, spatio-temporal segments (in a form of a 2D map) are extracted for the whole SITS, then, the temporal sequences constructed for segment descriptors are clustered. Therefore, the object-based SITS analysis has drastically reduced computational cost and ensured more homogeneous results of clustering algorithms compared with the pixel-based approaches.

Nevertheless, not so many SITS segmentation approaches are available [16] and it can be tricky to create a proper segmentation map for the whole series as sometimes objects change from one image to another. If a series is short enough (does not cover more than a year), we can presume that objects shapes stay invariant and, in this case, we can project a single image segmentation to the whole SITS. However, this approach can not be used for a series that covers a large period of time, especially if it contains some permanent changes or important phenological variations. To capture some of these changes, segmentation may be performed on the concatenated product of two or three most representative images of the SITS [13] or even on the concatenated product of the whole time series [14]. In the first case, we may miss some objects. In the second one, the segmentation may have high computational cost and be difficult to parameterize if a SITS is long.

To overcome multi-temporal segmentation issues, in Reference [17] the authors propose a graph-based approach to analyze different spatio-temporal dynamics in SITS. In this method, each image is segmented independently and all the spatio-temporal entities that belong to the same geographical location are connected to each other and form evolution graphs. Every graph is characterized by a bounding box—an object which footprint has the intersection with all graph objects at different timestamps. Following this method, Reference [18] proposes an algorithm to cluster the extracted multi-annual graphs. Each evolution graph is firstly described by a simplified representation—synopsis. Secondly, spectral and hierarchical clustering algorithms with DTW distance measure are applied to graphs synopsis. This approach showed promising results for the clustering of natural habitat areas. However, it may be complicated to construct evolution graphs for urban areas as their segmentation is more complicated due to the non-homogeneity of the features. For this reason, the segmentation results of urban areas are usually not uniform from one image to another, contrary to the agricultural lands where a parcel is presented by one or two well-delimited segments that repeat over time if no changes happened.

To create a single segmentation map for the whole SITS, the authors of Reference [19] propose a time series segmentation approach based on DTW distance measure. In this approach, at the beginning, each pixel is characterized by its temporal sequence, each sequence firstly represents an isolated segment, then the segments with similarity measure higher that a certain threshold are iteratively merged. However, for the aforementioned reasons, we estimate that the proposed approach can be slow, even if the distances are not computed for all pixel couples.

In this paper, we propose a SITS object-based clustering algorithm based on SITS compression with 3D convolutional autoencoder (AE). 3D convolutional networks have been successfully used in remote sensing applications for supervised classification [20,21] due to its ability to deal with multi-temporal image data in addition to lower computational cost comparatively to other temporal models such as, for example, convolutional Long Short-Term Memory (LSTM) network [22]. Contrary to these methods, our 3D convolutional AE model is unsupervised and does not require any labeled data and, to our knowledge, no such models have been used in time-series remote sensing yet.

In our work, we deploy an AE neural networks structure. Traditionally, autoencoders are used for unsupervised dimensionality reduction or feature learning [23]. Different AE models have been widely used in remote sensing [24–26]. In these articles, the features are extracted from a single image

using AEs and then used for a land scene classification. However, the AE structure can be adapted for any type of data, therefore, we propose to use AEs for the feature extraction and compression of the image series.

In our method, we first encode the whole SITS into a new feature image with a multi-view 3D convolutional AE. Both encoder and decoder parts contain two branches that are concatenated together before the bottleneck. While the first branch allows to obtain deep features from the spectral bands of the whole SITS, the second one only extracts some general information from the corresponding Normalized Difference Vegetation Index (NDVI) [27] images. Second, we perform a preliminary segmentation of the SITS on its two most representative images. Then, we correct the preliminary segmentation by using the encoded feature image. Finally, we regroup the obtained objects with hierarchical clustering algorithm [28] using the encoded features as descriptors. The proposed approach showed us good results in the two real-life datasets and outperformed the concurrent methods, including the ones based on the DTW measure.

We summarize our contributions as follows:

- We propose a fully unsupervised approach of SITS clustering using deep learning approaches.
- We propose a two-branch multi-view AE that extracts more robust features comparatively to a classic convolutional AE.
- We develop a segmentation approach that produces a unique segmentation map for the whole SITS.
- The proposed architecture is new and does not rely on a pre-existing or pre-trained network.

The rest of the paper is organized as follows—Section 2 presents the proposed approach, Section 3 describes datasets we used, Section 4 gives the review of the experimental results with their qualitative and quantitative evaluation. In the last section, we resume the work done and overview the future prospective.

2. Methodology

Our proposed approach is developed for segmentation and clustering of a SITS. Let R_S be a time series of S co-registered images $Im_1, Im_2, \ldots, Im_S$ acquired at timestamps $T_1, T_2, \ldots, T_S$. The framework is composed of several steps which are the following:

1. We start by relative normalization of all the images of the SITS using an algorithm described in Reference [29] and correction of saturated pixels.
2. We deploy a two-branch multi-view 3D convolutional AE model in order to extract spatio-temporal features and compress the SITS.
3. Then, we perform a preliminary SITS segmentation using two farthest images of the dataset taken in different seasons.
4. We correct the preliminary change segmentation using the compressed SITS.
5. Finally, we perform the clustering of extracted segments using their spatio-temporal features as descriptors.

2.1. Time Series Encoding

For the compression and encoding of the SITS, we propose to use the two-branch multi-view 3D convolutional AE. While the first branch of the AE extracts deep temporal features from the initial series, the second one extracts some primary temporal features from the associated NDVI images (Figure 1). The NDVI branch improves the model capacity to distinguish different vegetation types, especially the ones with weak seasonal variance. Moreover, by allocating a separate branch to NDVI images instead of just adding a supplementary NDVI channel to the initial images, we "force" the model to extract more robust and independent vegetation features.

Contrary to traditional 2D convolutional networks where convolution filters are applied in 2D plane, 3D convolutions preserve the temporal relations between data by extending the filters

to the depth dimension [30]. Therefore, the 3D convolution network extracts both spatial and temporal features.

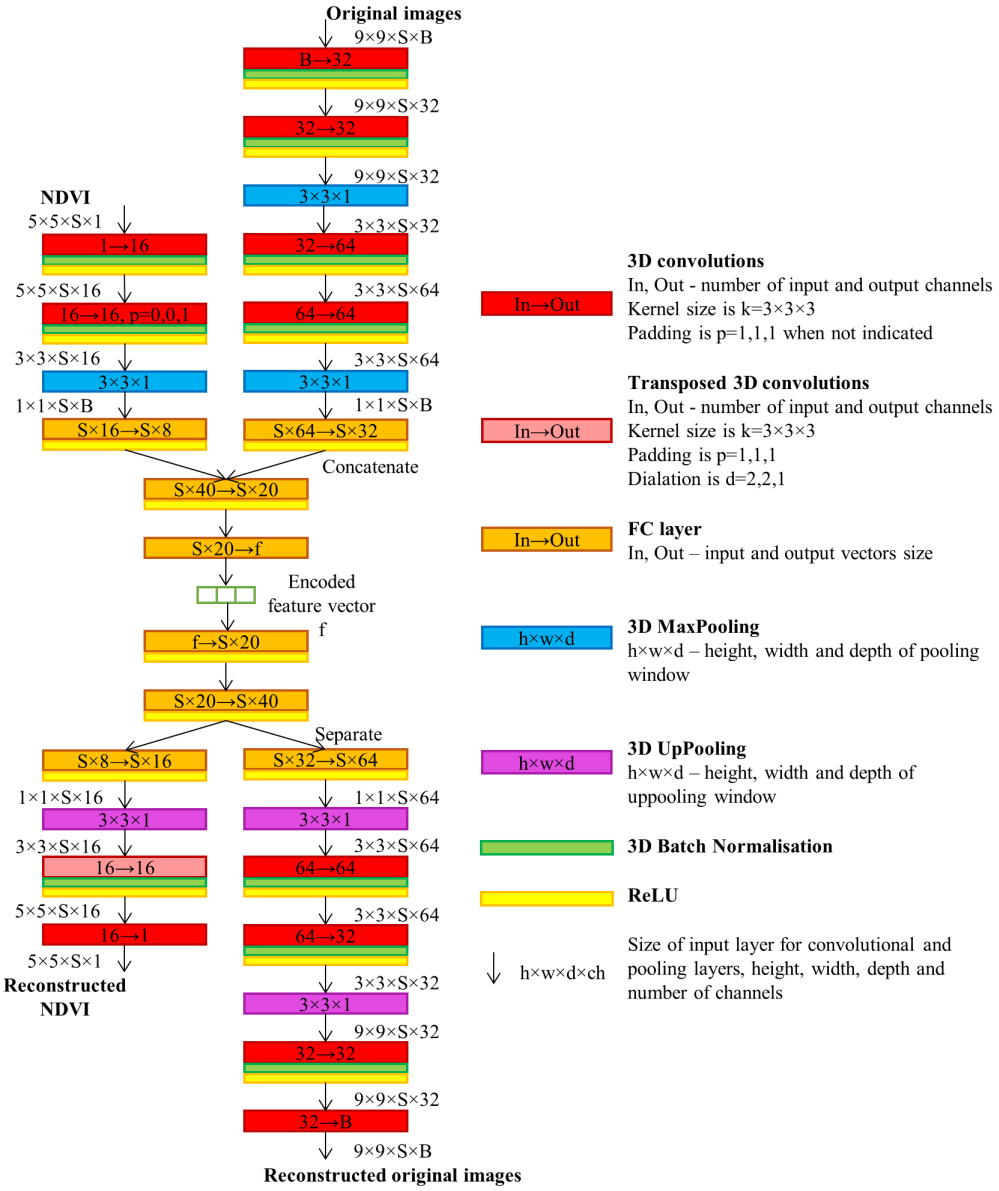

Figure 1. 3D convolutional autoencoder model.

The deployment of an AE type of model ensures the extraction of robust spatio-temporal features in an unsupervised manner without using any reference data. In classic AEs, the model firstly encodes the input data in some compressed latent representation and then decodes it back to its initial self. In image processing, the encoding pass is usually composed of convolutional and pooling layers for feature maps (FM) extraction that are followed by some fully-connected (FC) layers for feature compression. The decoding pass is often symmetrical to the encoding one. Once the model is trained, the extracted compressed representation is used to describe the data under the study. The encoder-decoder model allows us to compress the whole dataset in an uniform way. Moreover, it can compress any type of data independently of its shape and size.

In case of our multi-view AE, during the encoding step, we independently extract features from two different stack of images (original and their corresponding NDVI), the features are then merged

together to obtain a combined descriptor. During the decoding pass, the features are separated and reconstructed independently into the initial stacks of the original and NDVI images.

The training and encoding processes of the whole series are performed patch-wise for the stack of SITS images. The patches of size p are extracted for every i, j-pixel of the SITS ($i \in [1, H]$, $j \in [1, W]$, where H and W are images height and width respectively) and represent stacks of size $p \times p \times S \times B$, where B is the number of image bands. Obviously, for the first branch, B corresponds to the number of spectral bands, for the second one $B = 1$ as we deal with single channel NDVI images. To extract deep features from the original images, we propose to use patches of size $p = 9$, however, as we extract only general information from the NDVI images, the patch size of $p = 5$ is sufficient. We consider that $p = 9$ is big enough to get necessary information of the neighbor pixels as it makes a 90×90 m^2 surface footprint. In addition, it ensures smooth maxpooling with 3×3 window size and does not produce important border effect for the patches that contain two (or more) different classes (see more about it in the next subsection). For the NDVI branch, we believe that $p = 5$ is the minimum sufficient patch size to get the information about the neighborhood vegetation features ($p = 3$ covers only 1 pixel radius, so this information can not be considered relevant). Moreover, we apply no padding to the second 3D convolutional layer of the NDVI branch to reduce the size of extracted feature maps before applying the maxpooling operation. Note that we tend to decrease the network complexity and its training time by choosing a smaller NDVI patch size as all the important information about land cover textures are extracted in the main branch, while the NDVI branch is used only to detect vegetation tendencies. As one may observe from the model schema, the configuration of FC layers depends on the number of images of the SITS. It guarantees that all the layers within different models have the same input/output step ratio while compressing the features. Note than if S is elevated, one might consider to add a supplementary FC layer.

For model evaluation and optimization, we use the mean-squared error (MSE) (1):

$$MSE(o, t) = \frac{\sum_{n=1}^{N} l_n}{N}, l_n = (o_n - t_n)^2 \quad (1)$$

where o is the output patch of the model, t is the target patch and N is the number of patches per training batch.

Once the model is stable, every temporal stack of patches is encoded in a feature vector of size f that corresponds to the i, j-pixel of a new feature image of size $H \times W \times f$ that will be further used as a compressed version of the whole dataset.

2.2. Segmentation

Satellite image segmentation is a task of image processing that partitions an image into non-intersecting regions (segments) so that the ensemble of pixels of each region shares similar properties. Segmentation can therefore be seen as a first step before doing a classification or a clustering of the newly created segments for any object-based method.

As it was mentioned in the previous section, SITS segmentation can be a complicated and challenging process, especially when the number of images is elevated. The main idea of our segmentation approach is the following—to get a more robust SITS clustering that is easy to visualize, we need to obtain a unique segmentation map for the whole series. To accomplish this task, we could directly perform the segmentation on the encoded SITS image. However, as the encoding is performed in a patch-wise manner for every image pixel, one may observe a border effect. This effect is produced for pixels located close to a border of two regions. The patches extracted for these pixels contain information about two (or more) different classes, their encoded spatio-temporal features will not be "pure". For this reason, these pixels may be segmented as new objects (mostly linear) or segment borders may be shifted. Moreover, the linear objects, such as roads or rivers may not be distinguished or, on the contrary, over-segmented. Figure 2 presents two examples of the border effect and its eventual correction with our method (explained later in the text). The first row shows the shifted

borders in crop segmentation at the limits of different types of crops. The second row displays the segmentation of a road. We can observe that the road is over-segmented and its borders are shifted at the same time.

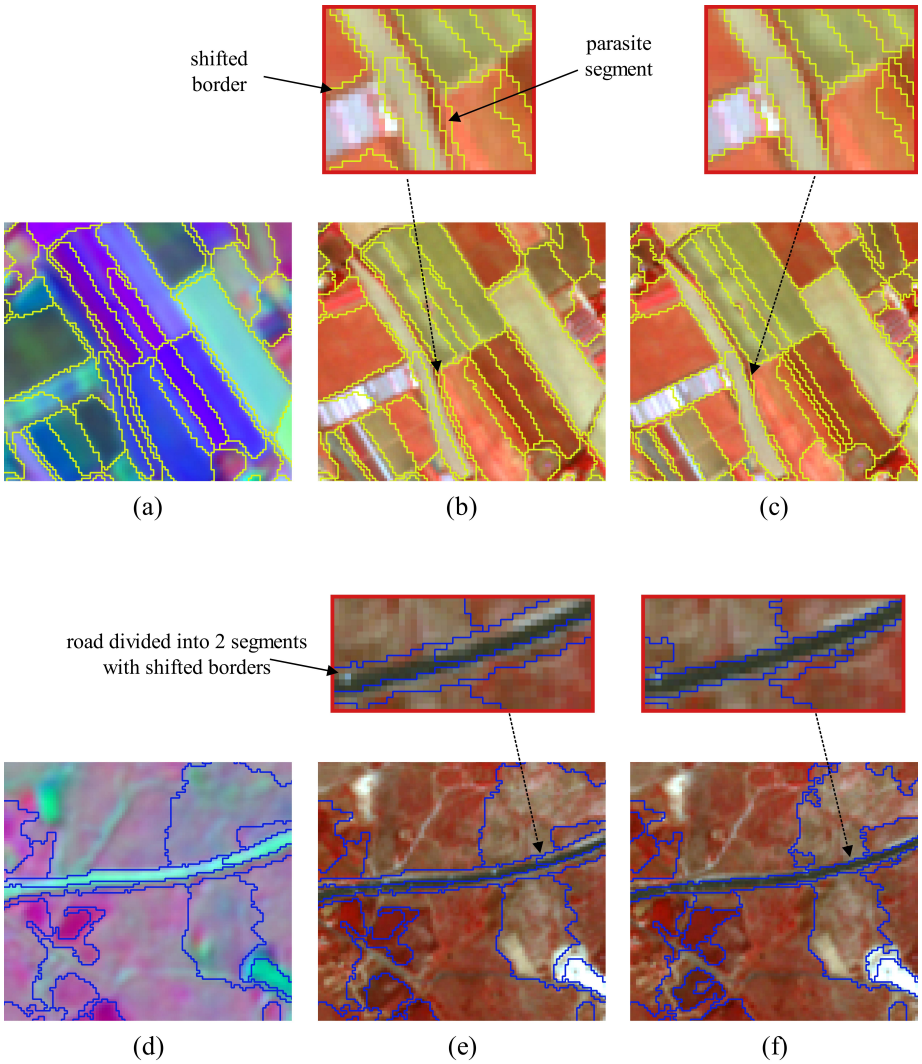

Figure 2. Influence of the border effect on the segmentation results and its eventual correction, example issued from SPOT-5 dataset (see dataset description in the following section). Top row–border effect in crops segmentation, bottom row–border effect in road segmentation. (**a**,**d**)—encoded images and corresponding segmentations, (**b**,**e**)–projection of these segmentations on the last image of the dataset, (**c**,**f**)—images corrected for the border effect. Note that 10-feature encoded image is presented in the limits of 3 channel combination, original SPOT-5 image is presented in false colors.

To tackle this problem, we propose to perform a two steps segmentation that includes the correction of the preliminary segmentation in respect to all objects borders of the time series. The preliminary segmentation is performed on two most representative concatenated images of the SITS. To obtain the maximum of coherent spatio-temporal objects in the preliminary segmentation Seg_{pr}, the chosen images should be as far apart as possible (e.g., the first and the last image) and correspond to different seasons.

For all image segmentations, the MeanShift [31] algorithm available in *Orfeo ToolBox* software (www.orfeo-toolbox.org) under *QGIS* interface was chosen. The most important parameters of the MeanShift segmentation algorithm are spatial radius R_s and range (spectral) radius R_r. The main idea of the algorithm is to firstly reproject a n-channels image into n-dimensional space and simplify its

representation by replacing each pixel with the mean of the pixels in R_r neighborhood that have values within R_s. The regions smaller than Reg_{min} are merged. Secondly, the algorithm reprojects the data back into the image plane and separates the areas with the same mean value into non-overlapping segments. At the end, the segments smaller than O_{min} are merged with their neighbors.

Despite the fact that Seg_{pr} gives us correct segment borders, it is impossible to identify all the objects presented in SITS on the base of only two images. Therefore, in the next step, we perform the segmentation Seg_{enc} of the encoded SITS that is represented as a f-channels image. As it was mentioned before, this segmentation would contain numerous irrelevant objects and shifted borders. Finally, we choose Seg_{pr} as the reference and we correct it by fitting the segments from Seg_{enc} to obtain the final segmentation map Seg_f.

The correction process is performed separately for each segment and is the following (see Figure 3):

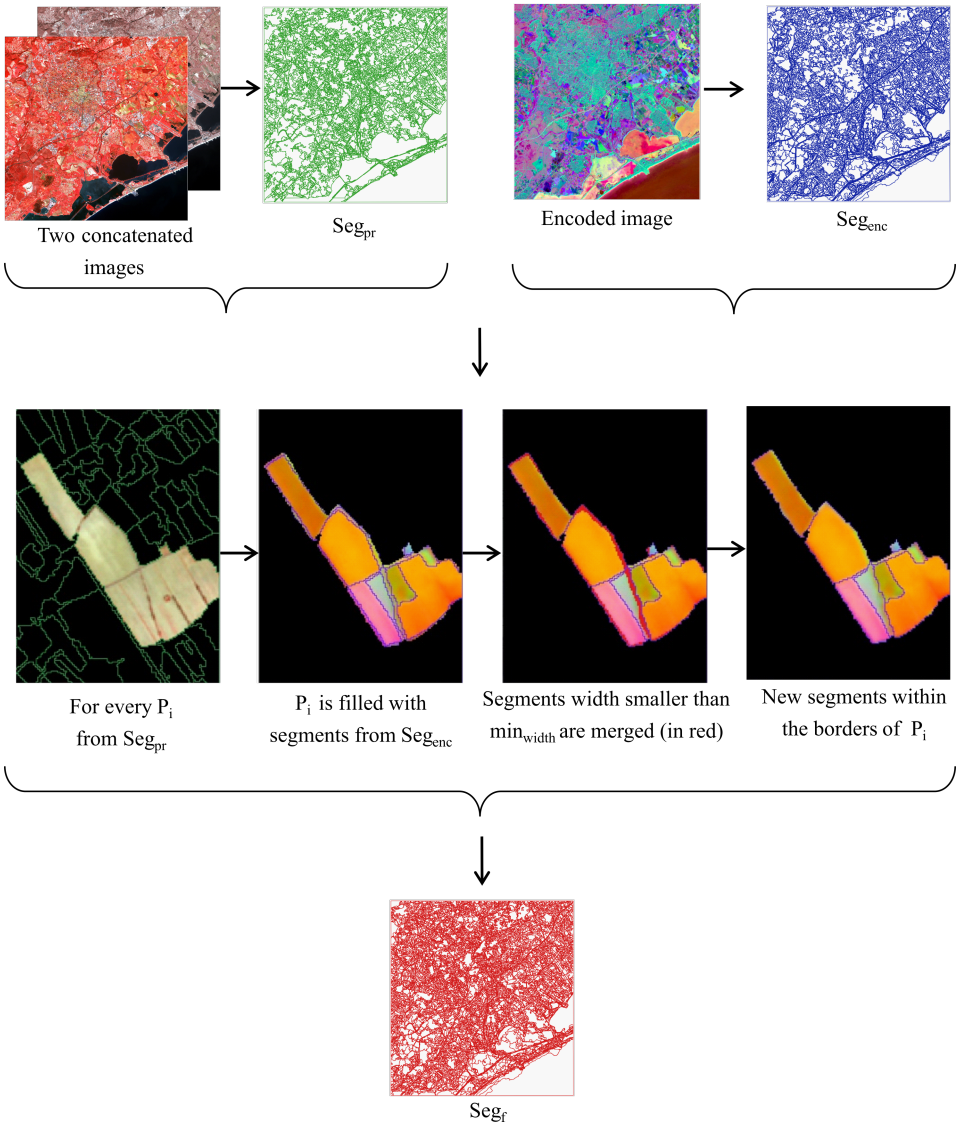

Figure 3. Segmentation correction.

- Let P_i be a segment from Seg_{pr} to correct.
- We firstly fill P_i with the segments from Seg_{enc} that have spatial intersection with it. P_i borders are preserved and used as the reference.
- Second, we check the average width of these segments in horizontal and vertical axes of the SITS coordinate system. We select objects with width smaller than min_{width} in at least one of the axes.

min_{width} size should not exceed half of the encoder patch size and be set after estimating the influence of the border effect.

- At the third place, each of these objects is merged with a neighbor with the biggest common edge if the edge is at least 3 pixels long or if the object's size does not exceed O_{min} (minimum object size that we want to distinguish in our experiments). Note that in case we have several segments to merge, we sort them by ascending size and start by merging the smallest one while other segments sizes are being iteratively updated.
- Finally, we fill a new segmentation map Seg_f with new merged segments.

Our method might still produce some shifted borders for some corrected segments, but at the same time, it allows to reduce the border effect to minimum, to preserve correct shapes of linear objects and to avoid parasite segments that correspond to border pixels.

2.3. Clustering

To regroup the obtained segments, we deploy the hierarchical clustering algorithm (HCA) [28] applied to segments descriptors.

Often, an output of a clustering algorithm does not correspond to the desired classes as some of them might be merged or, on the contrary, divided into two or more new clusters. For this reason, the user might make several tries in order to find an optimal number of clusters for the desired output partition. We choose HCA due to its ability to build a unique model to analyze cluster data at different levels. Contrary to other clustering algorithms, HCA model does not demand a researched number of clusters or a complex set of parameters that will further define the clusters number. During the algorithm execution, data points are presented as separate clusters and then, at every step, the model iteratively merges two clusters with the highest likelihood value. At the end, the user should simply choose the clustering level that corresponds the best to the desired clustering partition.

For segment descriptors, we use the median values of the encoded features of pixels within these segments. We choose the median values over the mean ones so the border pixels are not taken into account. We use Ward's linkage [28] and Euclidean distance between the segments as parameters for clustering algorithm.

3. Data

We evaluate the proposed approach on two real-life publicly available time series issued from SPOT-5 and Sentinel-2 missions. Both SITS are taken over the same geographical location (Montpellier area, France), but, however, differ in terms of spectral and temporal resolution. While the first SITS contains 12 images that are irregularly samples over 6 years, the second one contains 24 images taken over 2 years with more regular temporal resolution.

SPOT-5 dataset was taken between 2002 and 2008 and belongs to the archive Spot World Heritage (Available on https://theia.cnes.fr/). We have filtered out cloudy and flawed images from all the images acquired by the SPOT-5 mission over the considered geographic area and obtained 12 exploitable images with irregular temporal resolution (minimum temporal distance between two consecutive images is 2 months, maximum—14 months, average—6 months). Distribution of dataset images is presented in Table 1. All SPOT-5 images provide green, red, NIR and SWIR bands with 10-meters resolution.

Sentinel-2 dataset was taken between January 2017 and December 2018 (Available on https://earthexplorer.usgs.gov/). After deleting unexploitable images as well as the images that were less than 15 days apart from the previous images, we have obtained 24 images with more regular temporal resolution (minimum temporal distance between two consecutive images is 15 days, maximum—2.5 months, average—1 month). Distribution of the dataset images is presented in Table 2. Sentinel-2 images provide multiple spectral bands of different spectrum and spatial resolution, however, it was decided to keep only 10-meters resolution spectral bands—blue, green, red, NIR.

Table 1. Image acquisition dates for the SPOT-5 dataset.

	Acquisition Date, yyyy-mm-dd		
1	2002-10-05	7	2006-02-18
2	2003-09-18	8	2006-06-03
3	2004-05-14	9	2007-02-01
4	2004-08-22	10	2007-04-06
5	2005-04-27	11	2008-06-21
6	2005-12-01	12	2008-08-21

Table 2. Image acquisition dates for the Sentinel-2 dataset.

	Acquisition Date, yyyy-mm-dd						
1	2017-01-03	7	2017-08-20	13	2017-12-19	19	2018-07-17
2	2017-03-14	8	2017-09-20	14	2018-01-23	20	2018-08-06
3	2017-04-03	9	2017-10-10	15	2018-02-12	21	2018-08-26
4	2017-04-23	10	2017-10-30	16	2018-02-27	22	2018-09-20
5	2017-06-12	11	2017-11-14	17	2018-04-18	23	2018-10-05
6	2017-07-12	12	2017-11-29	18	2018-06-27	24	2018-12-29

The original images of both datasets are clipped to rectangular shapes of 1600×1700 pixels and transformed to UTM zone 31N: EPSG Projection. The clipped image extent corresponds respectively to the following latitude and longitude in WGS-84 system:

- bottom left corner: 43°30′6.0444″N, 3°47′30.066″E
- top right corner: 43°39′22.4856″N, 3°59′31.596″E

The pre-processing level of both datasets is 1C (orthorectified images, reflectance at the top of atmosphere). For this reason, both SITS were radiometrically normalized with the aim to obtain homogeneous and comparable spectral values over each dataset. For the image normalization, we have used an algorithm introduced in Reference [29] that is based on histogram analysis of pixel distributions.

The ground truth (GT) for both datasets was taken from an open data website of Montpellier agglomeration (http://data.montpellier3m.fr/) and correspond to landcover maps which we have manually modified to keep only distinguishable classes and merged the look-alike classes. While for the SPOT-5 dataset we have used Corina Land Cover (CLC) map of the 2008 year, for the Sentinel-2 dataset CLC of the 2017 year was taken. We have defined 9 well-distinguished GT classes:

1. urban and artificial area,
2. wooded area (include forests, parks, family gardens etc.),
3. natural area (not wooded),
4. water surface,
5. annual crops,
6. prairies,
7. vineyards,
8. orchards,
9. olive plantation.

For both datasets, the olive plantation class is very small, so we choose 8 reference classes for our clustering algorithm. The GT olive plantation class will be ignored during the evaluation.

Note that it is difficult to create a GT for a multi-annual SITS analysis as some objects may go through changes and it is impossible to detect all these changes manually. For this reason, for the SPOT-5 dataset, we use the GT that corresponds to the last year of the SITS. The SPOT-5 dataset was taken over 6 years and contains many change processes, mostly such as different constructions and permanent crop rotations. The study for change detection in the SPOT-5 dataset is presented

in Reference [6]. As these changes are less numerous, they will be considered by most of clustering algorithms as outliers, hence, they will be mixed with "stable" classes. However, some of these changes are only several timestamps long, so we still perform the clustering of the whole SITS instead of only free-change areas. Thus, the change areas will be regrouped with no change areas with the most similar temporal behavior or even make their proper clusters. At the same time, we consider that the Sentinel-2 dataset does not have any or has very few change areas as it is spread over only two years.

4. Experiments

All the algorithms were tested on 6 cores Intel(R) Core(TM) i7-6850K CPU 3.60 GHz with 32 GB of RAM computer with a NVIDIA Titan X GPU with 12 GB of RAM and developed in *Python* programming language using *PyTorch 1.3* library on Ubuntu 16.4. For segmentation, we used *Orfeo ToolBox 6.6.1 under QGIS 2.18*. *tslearn* library [32] was used to calculate DTW distance matrices for the concurrent approaches.

4.1. Experimental Settings

4.1.1. Time Series Encoding

As it was mentioned in Section 2, we have the same AE model for both datasets with the parameters that depend on the time series length. Thereafter, we set different sizes of the encoded feature vector f for our datasets that is proportional to the SITS length. f values were obtained empirically and correspond to $f = 10$ and $f = 20$ for the SPOT-5 and Sentinel-2 datasets respectively (for 12 and 24 images in the datasets).

During the model training, we equally add Gaussian noise with 0.25 factor to the input patches of the original images to extract more robust and generalized features. We do not add any noise to the NDVI patches as it reduces model capacities to differentiate some minor variations in the vegetation.

After several trials to assess the best parameters values, we set learning rate to 0.0001 and batch size to 150 to ensure the most optimal model converging during the training. We use all SITS patches during the training of the model. We train the model for both datasets for 2 epochs until it is stable. The number of epochs was obtained after the analysis of loss trend and the visual analysis of the encoded images. However, if one does not dispose of a sufficient graphic memory for 2 epochs model training, it can be trained for only one epoch without significant loss in accuracy.

4.1.2. Preliminary Segmentation

As it was mentioned earlier, we perform the preliminary segmentation of two concatenated images of the dataset that should be as far apart as possible and belong to two different seasons. For the SPOT-5 dataset, these images were the first and the last images of the dataset (taken on 2002-10-05 and 2008-08-21), for the Sentinel-2—the fifth and the last image (taken on 2017-06-12 and 2018-12-29). The chosen segmentation parameters for the MeanShift algorithm are presented in Table 3, other parameters are used by default in the *Orfeo ToolBox* software, including $Reg_{min} = 100$. For the choice of range and spatial radius, note that pixel values of SPOT-5 images do not exceed 475, while the Sentinel-2 images have 4096 maximum pixel value (after the elimination of saturated pixels). To simplify the choice of the parameters and reduce the computation time, the pixel values of the Sentinel-2 images were divided by 10 only for segmentation to bring these values closer to the ones of the SPOT-5 images.

The segmentation parameters were chosen to obtain the most relevant results for the reference objects.

Table 3. Preliminary segmentation parameters.

Dataset	Parameters		
	R_s	R_r	O_{min}
SPOT-5	45	40	10
Sentinel-2	40	35	10

4.1.3. Correction of Segmentation Results

As for the preliminary segmentation, we use MeanShift algorithm to segment the encoded images. Initially, the pixel values of encoded images are contained between -1 and 1 and then they have been re-scaled between 0 and 255 for the segmentation. The choice of both radiuses fully depend on the number of encoded features f and is the following: $R_s = 3 \times f$, $R_r = 2 \times f$. $O_{min} = 10$ as for preliminary segmentation.

For the correction of segmentation results we set $min_{width} < 4$ as it corresponds to the radius of neighborhood for the patches extracted for the original images.

4.1.4. Clustering

As in any unsupervised algorithm, the obtained clusters often do not correspond to the ones defined by the human. In most cases, one real-life class is often divided into two or even more clusters, so the manual inspection of the results is needed to choose the right number of clusters. For this reason, we perform clustering for different number of classes (from 8 to 15). We evaluate the obtained results with Normalized Mutual Information (NMI) index [33].

4.2. Results

To evaluate the proposed clustering framework, we compare it with different time series clustering approaches—object-based DTW, graph-based DTW [18], 3D convolutional AE without NDVI branch and a variation of our pipeline without segmentation correction.The following parameters are set for the concurrent approaches:

- **Object-based (OB) DTW**—as the segmentation reference we use the preliminary segmentation results for our method, we exploit hierarchical clustering algorithm with DTW distance matrix to regroup the obtained segments.
- **Graph-based (GB) DTW**—we use MeanShift algorithm with the following parameters to segment every image of the SITS $R_s = 20$, $R_r = 20$, $O_{min} = 10$ for SPOT-5 dataset and $R_s = 20$, $R_r = 15$, $O_{min} = 10$ for Sentinel-2 dataset. For both datasets we use the following parameters for graph construction, see Reference [18] for details: $\alpha = 0.3$, $\tau_1 = 0.5$. We omit τ_2 as it lowers the quality of the results. The hierarchical clustering method with DTW distance matrix is applied as in the original article.
- **3D convolutional AE without NDVI branch**—we use the same pipeline and segmentation parameters as in our proposed method, however, the feature extraction model is different. For feature maps extraction, we have the same 4 convolutional and 2 maxpooling layers as in the original images branch that are followed by 2 FC layers with the following input and output sizes for the encoding part: $S \times 64 \rightarrow S \times 12$, $S \times 12 \rightarrow f$. The decoder FC layers are symmetrical to the encoder.
- **3D convolutional AE without segmentation correction**—we use the same pipeline and segmentation parameters as in our proposed method, however, the clustering is performed for the preliminary segmentation made for two concatenated images.

For all methods with the DTW matrix computation we apply Sakoe-Chiba constraint [34] with bandwidth = 2 to speed up the algorithm and improve its results. This constraint restricts the alignment of the time series and prevents the shifting greater that one timestamp. We use hierarchical algorithm

with Ward's linkage [28] and Euclidean distance between the segments to cluster the SITS as in our methods. Mean segment descriptors are used as in the initial articles.

Moreover, for our method and for the object-based DTW, we equally perform clustering for the ground truth segmentation to analyze the robustness of the proposed object descriptors with an "ideal" segmentation map (referred as "GT seg." in the resulting table). All DTW algorithms are computed for the original images and for the ones enriched with the NDVI band.

We do not compare our method to any pixel-based approaches due to their high computation cost for the chosen datasets related to the distance matrix size and the memory allocation.

4.2.1. Quantitative Evaluation

The evaluation of the obtained results with the NMI index is presented in Table 4. For the aforementioned reasons, we present NMI for the reference number of classes (8) as well as the best NMI score within the selected range of classes ("NMI best" in the table). As the results obtained with the AE methods may vary between several attempts, each AE method was launched 3 times. For these methods, we present the mean NMI value with the error margins.

As it was mentioned in Section 3, the SPOT-5 dataset was taken over 6 years and contains several changes. However, the GT corresponds to 2008 year (the end of the series). For this reason, we compute NMI for the whole SITS as well as for the SITS without outliers (which are mostly made of these changes and therefore do not match the 2008 end of the series GT). We exclude the airport area when computing NMI as it is mostly represented by a grass field. Its temporal behavior is unknown to us, so we can not associate it to any of existing clusters.

Table 4. Accuracy of different methods.

	SPOT-5		**SPOT-5 w/o Outliers**		**Sentinel-2**	
	NMI	**NMI Best**	**NMI**	**NMI Best**	**NMI**	**NMI Best**
Our method	**0.45 ± 0.01**	**0.45 ± 0.01**	**0.5 ± 0.01**	**0.5 ± 0.01**	0.44 ± 0.01	**0.45 ± 0.01**
OB DTW	0.4	0.4	0.43	0.46	0.36	0.38
OB DTW + NDVI	0.41	0.41	0.44	0.44	0.36	0.37
GB DTW	0.36	0.38	0.37	0.39	0.36	0.36
GB DTW + NDVI	0.41	0.42	0.42	0.42	0.37	0.37
AE w/o NDVI	**0.45 ± 0.01**	**0.46 ± 0**	0.48 ± 0.02	0.48 ± 0.02	0.43 ± 0.01	0.43 ± 0.01
AE w/o segm. corr.	0.42 ± 0.01	0.42 ± 0.01	0.47 ± 0	0.47 ± 0	0.43 ± 0.01	0.43 ± 0.01
Our method + GT seg.	0.52 ± 0.01	**0.52 ± 0.01**	0.58 ± 0.01	**0.59 ± 0**	0.46 ± 0.05	**0.5 ± 0.01**
OB DTW + GT seg.	0.51	**0.53**	0.52	0.54	0.48	0.48
OB DTW + GT seg. + NDVI	0.52	**0.53**	0.54	0.54	0.4	0.45

Table 5 presents the computation times for the most essential and time consuming steps of the presented approaches. For the DTW methods, we also present the number of segments or graphs to give an idea about the size of the distance matrix as it is the most defining parameter of the computation time. Note that we present the results for the DTW matrix computation made with *tslearn* library based on CPU computations with parallelisation. For a fairer comparison, we also customized parts of *tslearn* source code from the original library to run it on GPU with *numba* library. Doing so has greatly improved the computation time (maximum computation time for the longest sequence was less than 2 min).

It is still worth mentionning that GPU transfer is not adapted to all types of data, for instance even with a GPU, the computation time increases greatly when the sequence lengths grows. Moreover, regardless of whether it is paralellized on a GPU or a CPU, DTW computations are limited by the number of sequences: the size of the distance matrix grows with the number of sequences, and at some point can not be stored in memory anymore.

Table 5. Computation time.

	Algorithm Step	Computation Time, Min	
		SPOT-5	**Sentinel-2**
Our method	Model training [a] + encoding	15 + 5	25 + 6
AE w/o NDVI	Model training [a] + encoding	11 + 3	15 + 4
OB DTW	DTW calc. [b]	35 (4433 objects)	30 (3751 objects)
OB DTW + NDVI	DTW calc. [b]	40 (4433 objects)	33 (3751 objects)
GB DTW	Graph constr. + DTW calc. [b]	26 + 33 (3550 graphs)	55 + 51 (3739 graphs)
GB DTW + NDVI	Graph constr. + DTW calc. [b]	26 + 36 (3550 graphs)	55 + 56 (3739 graphs)
OB DTW + GT seg.	DTW calc. [b]	202 (10,725 objects)	299 (12,236 objects)
OB DTW + GT seg. + NDVI	DTW calc. [b]	232 (10,725 objects)	335 (12,236 objects)

[a] Time given for one epoch. [b] Time given for the full computation using *tslearn* library.

Please, note that the computation time of our algorithm can be improved by around 30% with the parallelisation of branches computations.

4.2.2. Qualitative Evaluation. Segmentation

Firstly, we analyze the results obtained for the "ideal" GT segmentation for our proposed method and for the object-based DTW. We can observe from the Table 4 that both methods gave higher scores comparatively to the user-made segmentation. At the same time, in average, our method has slightly outperformed its concurrent approach. Moreover, when using the existing *Python* libraries, our method has less computational cost, hence, is more adapted for big datasets.

For the clustering results obtained for user-made segmentation, we observe that the results of our method are significantly better than the concurrent approaches. It can be explained by more precise segmentation results that were achieved by our proposed methodology. We remind that for our method, segmentation is realized in several steps: preliminary segmentation, segmentation of the encoded SITS image and the final corrected segmentation that combines the previous ones. At the same time, the object-based clustering method is performed on the preliminary segmentation results. The preliminary segmentation results are always under-segmented as they can not caption all the seasonal vegetation variations that are reflected in the encoded SITS image.

The obtained results highlight the necessity of the segmentation correction as one can notice the decreasing of the accuracy when none is performed. Figure 4 presents the advantage of our proposed approach with higher detalization level over the segmentation performed on the two most representative images of the dataset.

Moreover, our proposed approach correctly produces segments free of border effect that perfectly correspond to true object borders (see Figure 2) that facilitate the interpretation of the obtained objects.

The graph-based DTW approaches gave the results similar to the object-based DTW methods in addition to the slowest computation time related to the graph construction. This poor performance can be explained by the difficulty to segment the whole SITS, especially in the urban area. However, this method might give better results for a smaller dataset with prevailing agricultural areas.

4.2.3. Qualitative Evaluation. Feature Extraction and Clustering

Figure 5 presents clustering results for our method for both datasets for the best of 3 runs. After visual analysis and the analysis of the NMI values, it was established that 14 clusters provided the best data partitioning for both datasets. The obtained clusters were associated to the ground truth clusters and colored in the same manner.

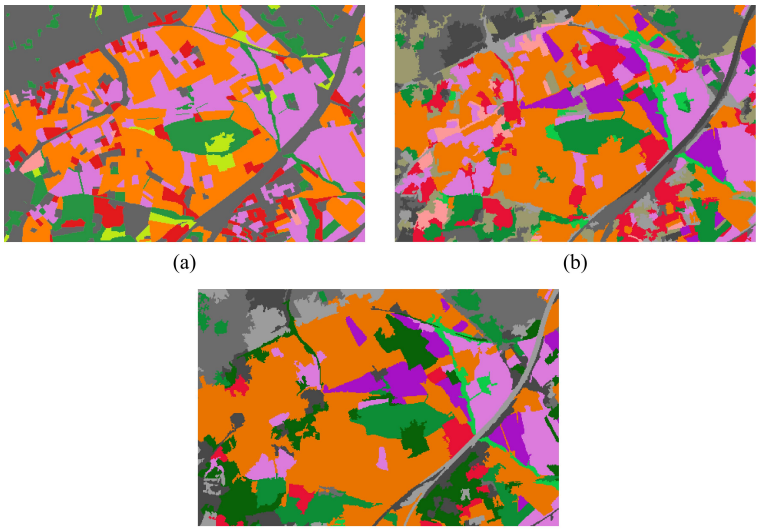

Figure 4. Example of clustering results highlighting the advantage of our proposed segmentation correction. (**a**) ground truth (GT) for 2017 year, (**b**) results for our method for the Sentinel-2 dataset, (**c**) results for our pipeline without segmentation correction for the Sentinel-2 dataset (clustering performed on objects extracted from the segmentation of two most representative images). The clusters are colored accordingly to the GT map. The reference map legend can be consulted on Figure 5.

As it was mentioned in the data section, the olive plantation class was ignored during the analysis due to its small size in addition to the fact that none of the proposed algorithms has identified it in a separated cluster.

Clusters corresponding to water surfaces and urban areas were correctly identified for both datasets. Three vegetation classes corresponding to annual crops, vineyards and prairies were mostly correctly detected as well due to the specificities of their temporal behavior. We do not dispose of finer classification level for annual crops, however, the hierarchical algorithm has divided it in several clusters which we can easily distinguish by analyzing crops temporal behavior. At the other hand, the orchard class is mixed with other vegetation classes for both datasets and the natural areas with small vegetation are mixed with wooded areas. It can be explained by the fact that both classes have similar growing cycles and spatial textures.

After comparing the obtained clusters with change maps from Reference [6], we associate one of the clusters obtained for SPOT-5 dataset with changes corresponding to the construction of a new area.

Unfortunately, some linear objects are not correctly clustered despite the fact that they are correctly segmented. Linear objects presented by rivers and narrow vegetation areas are mostly misclassified. We find several explanations for it: the main reason is that the segments are too narrow and contain many encoded feature pixels affected by the border effect. For the clustering, we use the median feature values of the segments as their descriptors which are biased with border pixels. On the other hand, the algorithm does not detect roads as a separate class, but it still classifies them as different variations of the urban cluster most of the time. We explain it by the fact that the detected roads are in average a little bit larger than other linear objects and their feature response is better defined, hence, their median segment values are less biased.

As it can be seen, the variation of our model without NDVI branch gives the NMI score similar to our method, but after the visual examination of the results (see Figure 6), we can state that our method with the NDVI branch gives better clustering results of vegetation areas. For example, for the number of clusters higher than 8, it distinguishes different types of annual crops that do not figure in GT maps, but can be easily spotted on the SITSs. Furthermore, when increasing the number of clusters, our model with the NDVI branch gives the results that are more intuitive to interpret as the model iteratively divides each cluster into two clusters of more or less proportional sizes. At the other

hand, the model without the NDVI branch produces many small clusters: for example, for 14 clusters segmentation results, we obtain 6 clusters of size less that 1000 pixels each that correspond to some minor variations in water bodies. At the same time, some vegetation types are not detected at all: classes corresponding to prairies and orchards are missing.

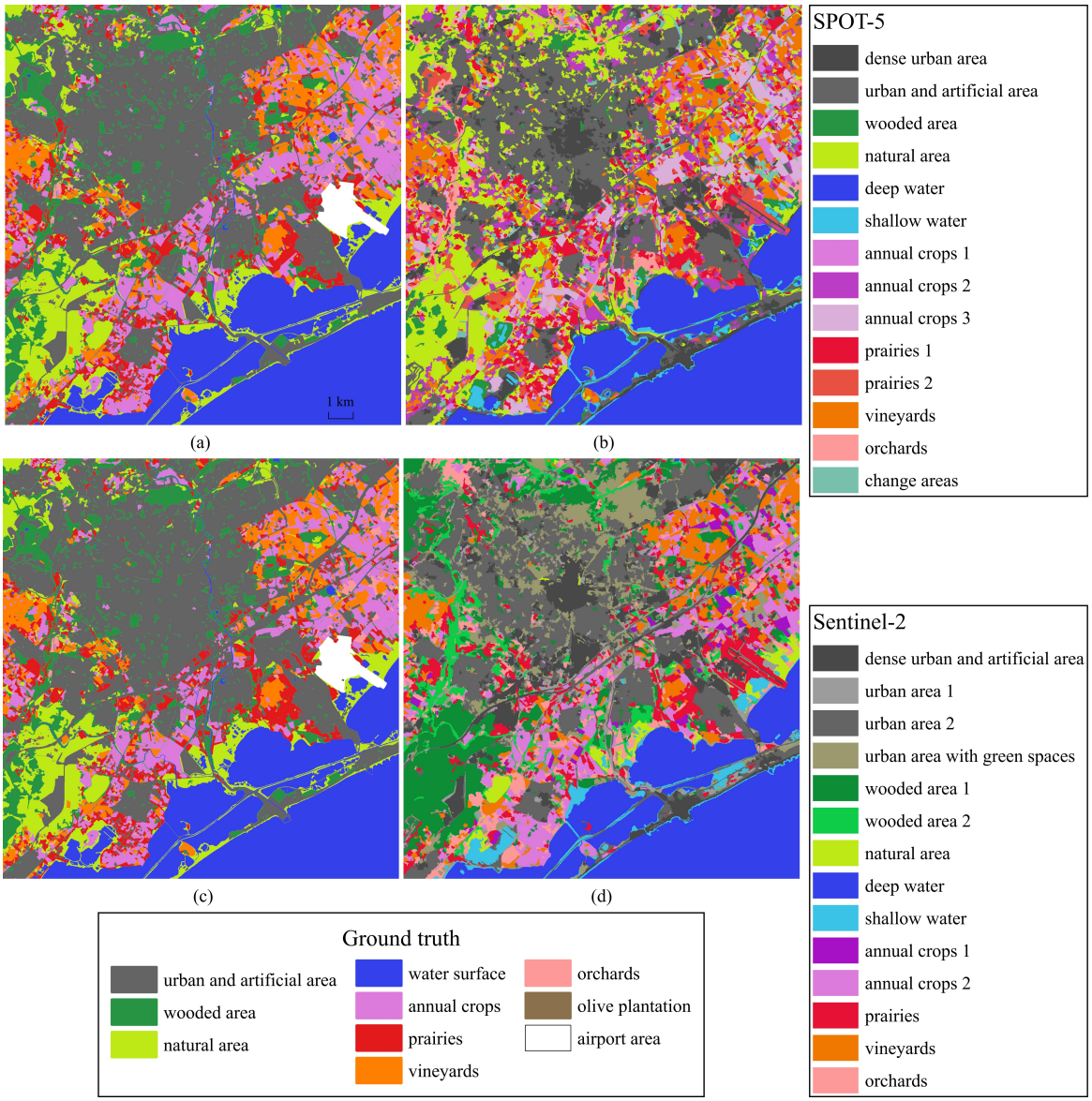

Figure 5. Clustering results for our proposed pipeline. (**a**) GT for 2008 year, (**b**) results for the SPOT-5 dataset, (**c**) GT for 2017 year, (**d**) results for the Sentinel-2 dataset.

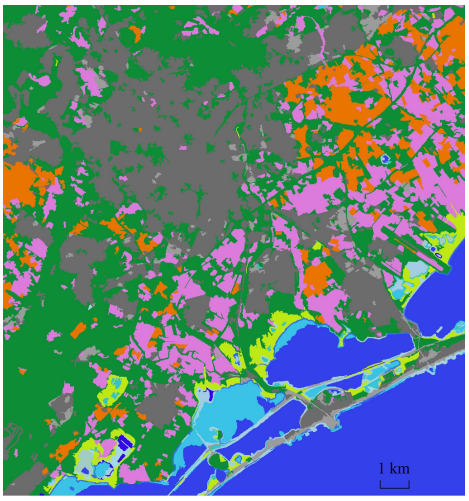

Figure 6. Clustering results for the proposed pipeline without Normalized Difference Vegetation Index (NDVI) branch. The clusters are colored according to the clustering map (b) in Figure 5. 6 clusters of water surface are colored in different shades of blue.

For the concurrent approaches, we can spot that enriching the original images with NDVI band has only slightly increased the accuracy of the DTW methods, though no significant improvements were done.

Figure 7 presents the classification results for the concurrent approaches for the best number of clusters with the added NDVI band. We notice that clusters that correspond to the residential area and the forest are often mixed. It can be explained by the fact that no spatial features were extracted and the mean segment value can not always discriminate these classes. Moreover, the prairies and orchards clusters are mixed with other vegetation clusters, that is probably related to the imperfection of the segmentation. It made it more difficult to associate the obtained results to the real-life classes than for our method as the "computer clustering logic" is less obvious.

We equally notice that the graph-based method does not ensure the whole coverage of the study area due to the specificity of the algorithm (non-covered areas are shown in white) that leads to some missing information. Moreover, this method gives the most uneven shapes of clusters, especially, in the city area that might complicate the interpretation of the results.

We believe that the chosen area is a relatively complex landscape because it contains many different land cover types, so potentially it should work for any similar area or areas with less classes (including areas that are predominantly agricultural, as about half of our study area corresponds to vegetation). However, for a larger number of clusters, the readers should understand that no ideal unsupervised clustering algorithm exists. When the number of ground truth classes is much higher, we usually have to deal with look-alike classes. For example, two classes have the same growing cycle, but slightly different textures. These classes will be difficult to distinguish when no supervised constraint is applied.

Furthermore, besides the weakness that we mentioned for the clustering of linear objects, and the cases of a large number of lookalike land cover classes, we did not detect any limitations specific to our algorithm. Although, we insists that our algorithm still possesses all the limitations common to unsupervised neural networks and clustering. For example, for a dataset with unbalanced classes, it may be difficult to extract the features for small size classes; the final model varies from one experiment to another; setting the right parameters for the network may sometimes be complex and depends on the data. Moreover, if one is aware that the study area contains various changes in land cover types, we recommend to first perform a change detection analysis and interpret it together with the clustering results. Finally, as for any unsupervised methods, the obtained clusters do not always correspond to the expected classes, as the computer's pattern analysis is different from the human perception.

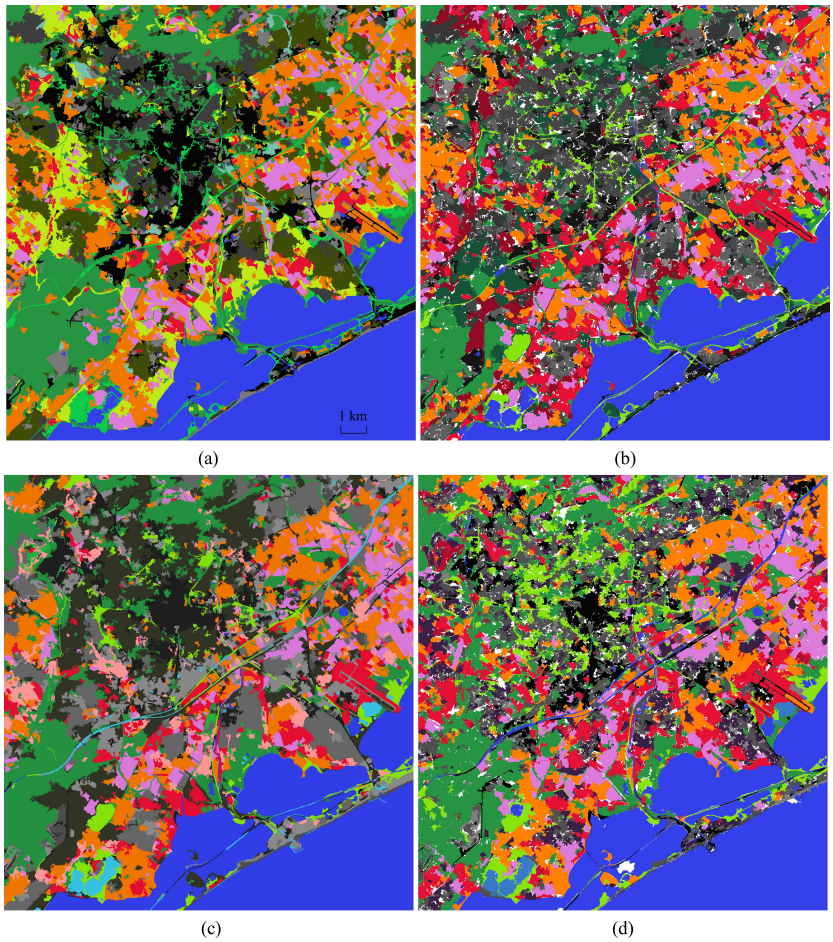

Figure 7. Clustering results for the concurrent approaches. (**a**) Object-based Dynamic Time Warping (DTW) for the SPOT-5 dataset, (**b**) Graph-based DTW for the SPOT-5 dataset, (**c**) Object-based DTW for the Sentinel-2 dataset, (**d**) Graph-based DTW for the Sentinel-2 dataset. For the legend, please, refer to Figure 5. The clusters are colored accordingly to the GT maps, if one class is presented by several clusters, these clusters are colored in different shades of the referent class color. For graph-based DTW white pixels correspond to the areas that are not covered by graphs.

5. Conclusions

In this article, we have presented a fully unsupervised approach for SITS clustering based on a two branch multi-view 3D convolutional AE that does not demand any labeled data. The proposed approach exploits the AE model to compress a time series into an encoded image by extracting its spatio-temporal features. Then it performs the segmentation of the encoded image with the eventual correction of shifted segment borders related to the specificity of the encoding. The proposed approach was tested on two real-life datasets and showed its efficiency comparatively to the concurrent approaches.

The main advantages of the proposed algorithm include the improvement of traditional segmentation methods that are not initially adapted for the SITS that leads to higher NMI score. In addition, we have shown that we can improve clustering results by simply introducing a temporal NDVI branch in the AE model. The presented approach is a good alternative to traditional DTW-based methods as deep learning techniques are able to extract more robust and complex features compared with traditional Machine Learning methods.

In our future plans, we want to adapt our algorithm to all spectral bands of Sentinel-2 datasets as well as improve its accuracy for the clustering of linear objects. Other spectral indices can be also integrated in the proposed model, however, a closer study is needed to estimate the influence of each

index on the clustering results. Moreover, the contextual constraint may be introduced to distinguish more classes (e.g., artificial areas such as beaches can be discriminated from urban areas as they are close to the water).

Author Contributions: This research article was written as part of E.K. ongoing Ph.D. Thesis under the joint supervision of M.T. and J.S., E.K. did most of the heavy work regarding the data preparation, programming, methodology, draft preparation, design of the experimental protocol and experimentations. The validation of the results was a joint work between E.K. and J.S. The review and revision of the paper, as well as the project supervisions were done by M.T. and J.S., who also provided the original subject and handled the project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by an EDITE Ph.D. scholarship.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SITS	Satellite Image Time Series
AE	AutoEncoder
NN	Neural Network
NMI	Normalized Mutual Information
DTW	Dynamic Time Warping
GT	Ground Truth
FC	Fully-Connected
NDVI	Normalized Difference Vegetation Index
FM	Feature Maps
OB	Object-based
GB	Graph-based

References

1. Nativi, S.; Mazzetti, P.; Santoro, M.; Papeschi, F.; Craglia, M.; Ochiai, O. Big Data challenges in building the Global Earth Observation System of Systems. *Environ. Model. Softw.* **2015**, *68*, 1–26. doi:10.1016/j.envsoft.2015.01.017. [CrossRef]
2. Kuenzer, C.; Dech, S.; Wagner, W. *Remote Sensing Time Series—Revealing Land Surface Dynamics*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 22. doi:10.1007/978-3-319-15967-6. [CrossRef]
3. Alqurashi, A.; Kumar, L.; Sinha, P. Urban Land Cover Change Modelling Using Time-Series Satellite Images: A Case Study of Urban Growth in Five Cities of Saudi Arabia. *Remote. Sens.* **2016**, *8*, 838. doi:10.3390/rs8100838. [CrossRef]
4. Van Hoek, M.; Jia, L.; Zhou, J.; Zheng, C.; Menenti, M. Early Drought Detection by Spectral Analysis of Satellite Time Series of Precipitation and Normalized Difference Vegetation Index (NDVI). *Remote. Sens.* **2016**, *8*, 422. [CrossRef]
5. Song, X.P.; Huang, C.; Sexton, J.O.; Channan, S.; Townshend, J.R. Annual Detection of Forest Cover Loss Using Time Series Satellite Measurements of Percent Tree Cover. *Remote. Sens.* **2014**, *6*, 8878–8903. [CrossRef]
6. Kalinicheva, E.; Ienco, D.; Sublime, J.; Trocan, M. Unsupervised Change Detection Analysis in Satellite Image Time Series using Deep Learning Combined with Graph-Based Approaches. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2020**. doi:10.1109/JSTARS.2020.2982631. [CrossRef]
7. Gómez, C.; White, J.C.; Wulder, M.A. Optical remotely sensed time series data for land cover classification: A review. *ISPRS J. Photogramm. Remote. Sens.* **2016**, *116*, 55–72. doi:10.1016/j.isprsjprs.2016.03.008. [CrossRef]
8. Pelletier, C.; Webb, G.I.; Petitjean, F. Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series. *Remote. Sens.* **2019**, *11*, 523. doi:10.3390/rs11050523. [CrossRef]
9. Interdonato, R.; Ienco, D.; Gaetano, R.; Ose, K. DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn. *ISPRS J. Photogramm. Remote. Sens.* **2019**, *149*, 91–104. doi:10.1016/j.isprsjprs.2019.01.011. [CrossRef]

10. Petitjean, F.; Inglada, J.; Gancarski, P. Clustering of satellite image time series under Time Warping. In Proceedings of the 2011 6th International Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp), Trento, Italy, 12–14 July 2011; pp. 69–72. doi:10.1109/Multi-Temp.2011.6005050. [CrossRef]
11. Zhang, Z.; Tang, P.; Huo, L.; Zhou, Z. MODIS NDVI time series clustering under dynamic time warping. *Int. J. Wavelets MultireSolut. Inf. Process.* **2014**, *12*, 1461011. doi:10.1142/S0219691314610116. [CrossRef]
12. Rakthanmanon, T.; Campana, B.; Mueen, A.; Batista, G.; Westover, M.B.; Zhu, Q.; Zakaria, J.; Keogh, E. Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. *Acm Trans. Knowl. Discov. Data* **2013**, *7*. doi:10.1145/2500489. [CrossRef]
13. Belgiu, M.; Csillik, O. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote. Sens. Environ.* **2017**, *204*. doi:10.1016/j.rse.2017.10.005. [CrossRef]
14. Csillik, O.; Belgiu, M.; Asner, G.; Kelly, M. Object-Based Time-Constrained Dynamic Time Warping Classification of Crops Using Sentinel-2. *Remote. Sens.* **2019**, *11*, 1257. doi:10.3390/rs11101257. [CrossRef]
15. Petitjean, F.; Weber, J. Efficient Satellite Image Time Series Analysis Under Time Warping. *IEEE Geosci. Remote. Sens. Lett.* **2014**, *11*, 1143–1147. doi:10.1109/LGRS.2013.2288358. [CrossRef]
16. Petitjean, F.; Kurtz, C.; Passat, N.; Gancarski, P. Spatio-Temporal Reasoning for the Classification of Satellite Image Time Series. *Pattern Recognit. Lett.* **2013**, *33*, 1805. doi:10.1016/j.patrec.2012.06.009. [CrossRef]
17. Guttler, F.; Ienco, D.; Nin, J.; Teisseire, M.; Poncelet, P. A graph-based approach to detect spatiotemporal dynamics in satellite image time series. *ISPRS J. Photogramm. Remote. Sens.* **2017**, *130*, 92–107. doi:10.1016/j.isprsjprs.2017.05.013. [CrossRef]
18. Khiali, L.; Ndiath, M.; Alleaume, S.; Ienco, D.; Ose, K.; Teisseire, M. Detection of spatio-temporal evolutions on multi-annual satellite image time series: A clustering based approach. *Int. J. Appl. Earth Obs. Geoinf.* **2019**, *74*, 103–119. [CrossRef]
19. Costa, W.; Fonseca, L.; Körting, T.; Simoes, M.; Kuchler, P. A Case Study for a Multitemporal Segmentation Approach in Optical Remote Sensing Images. In *Proceedings of 10th International Conference on Advanced Geographic Information Systems, Applications, and Services*; Israel Institute of Technology: Haifa, Israel, 2018; pp. 66–70.
20. Ji, S.; Chi, Z.; Xu, A.; Duan, Y. 3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images. *Remote. Sens.* **2018**, *10*, 75. doi:10.3390/rs10010075. [CrossRef]
21. Li, Y.; Zhang, H.; Shen, Q. Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote. Sens.* **2017**, *9*, 67. doi:10.3390/rs9010067. [CrossRef]
22. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *arXiv* **2015**, arXiv:1506.04214.
23. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; Adaptive Computation and Machine Learning Series; Chapter Autoencoders; MIT Press: Cambridge, MA, USA, 2016.
24. Xing, C.; Ma, L.; Yang, X. Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images. *J. Sens.* **2016**, *2016*, 1–10. doi:10.1155/2016/3632943. [CrossRef]
25. Cui, W.; Zhou, Q. Application of a Hybrid Model Based on a Convolutional Auto-Encoder and Convolutional Neural Network in Object-Oriented Remote Sensing Classification. *Algorithms* **2018**, *11*, 9. doi:10.3390/a11010009. [CrossRef]
26. Liang, P.; Shi, W.; Zhang, X. Remote Sensing Image Classification Based on Stacked Denoising Autoencoder. *Remote. Sens.* **2017**, *10*, 16. doi:10.3390/rs10010016. [CrossRef]
27. Rouse, J.W.; Haas, R.H.; Schell, J.A.; Deering, D.W. *Monitoring Vegetation Systems in the Great Plains with Erts*; NASA Special Publication: Washington, DC, USA, 1974; Volume 351, p. 309.
28. Ward, J.H., Jr. Hierarchical Grouping to Optimize an Objective Function. *J. Am. Stat. Assoc.* **1963**, *58*, 236–244. [CrossRef]
29. El Hajj, M.; Bégué, A.; Lafrance, B.; Hagolle, O.; Dedieu, G.; Rumeau, M. Relative Radiometric Normalization and Atmospheric Correction of a SPOT 5 Time Series. *Sensors* **2008**, *8*, 2774–2791. doi:10.3390/s8042774. [CrossRef]
30. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497. doi:10.1109/ICCV.2015.510. [CrossRef]

31. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40. doi:10.1109/TIT.1975.1055330. [CrossRef]
32. Tavenard, R.; Faouzi, J.; Vandewiele, G.; Divo, F.; Androz, G.; Holtz, C.; Payne, M.; Yurchak, R.; Rußwurm, M.; Kolar, K.; et al. Tslearn: A Machine Learning Toolkit Dedicated to Time-Series Data. 2017. Available online: https://github.com/rtavenar/tslearn (accessed on 1 May 2020).
33. Cover, T.M.; Thomas, J.A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*; Wiley-Interscience: Hoboken, NJ, USA, 2006.
34. Sakoe, H.; Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [CrossRef]

Sample Availability: The source code and image results are available from the authors and in the *GitHub* repository https://github.com/ekalinicheva/3D_SITS_Clustering/.