Satellite Image Time Series Classification
with Pixel-Set Encoders and Temporal Self-Attention

V. Sainte Fare Garnot¹, L. Landrieu¹, S. Giordano¹, N. Chehata¹²
¹ LASTIG, Univ Gustave Eiffel, ENSG, IGN, F-94160 Saint-Mande, France
² EA G&E Bordeaux INP, Université Bordeaux Montaigne, France

Abstract

Satellite image time series, bolstered by their growing availability, are at the forefront of an extensive effort towards automated Earth monitoring by international institutions. In particular, large-scale control of agricultural parcels is an issue of major political and economic importance. In this regard, hybrid convolutional-recurrent neural architectures have shown promising results for the automated classification of satellite image time series.

We propose an alternative approach in which the convolutional layers are advantageously replaced with encoders operating on unordered sets of pixels to exploit the typically coarse resolution of publicly available satellite images. We also propose to extract temporal features using a bespoke neural architecture based on self-attention instead of recurrent networks. We demonstrate experimentally that our method not only outperforms previous state-of-the-art approaches in terms of precision, but also significantly decreases processing time and memory requirements. Lastly, we release a large open-access annotated dataset as a benchmark for future work on satellite image time series.

1. Introduction

The rising availability of high quality satellite data by both state [43, 10] and private actors [37] opens up numerous high-impact applications for machine learning methods. Among these, crop type classification is a major challenge for agricultural and environmental policy makers. In the European Union (EU), yearly crop maps are needed to grant the Common Agricultural Policy subsidies, an endowment of over 50 billion euros each year [1]. Currently, European farmers declare the cultivated species manually on a yearly basis. The EU’s Joint Research Center has thus called for the development of efficient tools to achieve automated monitoring [2]. This push to automation is motivated in part by the launch of the Sentinel-2 satellite—which became fully operational in mid-2017—by the European Space Agency [10], and whose settings are particularly valuable for crop classification. Indeed, its high spectral resolution (13 bands) and short revisit time of 5 days are well-suited to analysing crop phenology, i.e. the cyclical evolution of vegetation [40]. Additionally, the farmers’ yearly manual declarations provide a considerable amount of annotated data (10 million parcels labelled each year in France alone) to train learning algorithms. Such models would have a wide array of applications beyond crop monitoring, for both public and private entities.

Practitioners mainly rely on traditional methods such as Random Forest (RF) and Support Vector Machine (SVM), which operate on handcrafted features for automated crop classification [15, 47]. Recently, the gradual adoption of deep learning methods such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for learning spatial and temporal attributes has brought significant improvements in classification performance. More specifically, hybrid neural architectures combining convolutions and recurrent units in a single architecture are the current state-of-the-art for crop type classification [31, 34].

In this paper, we argue that such hybrid recurrent-convolutional architectures fail to adapt to some key characteristics of the problem under consideration.

Spatial Encoding of Parcels: Sensors typically used for crop classification, such as the Sentinel-2 satellites, have a coarser spatial resolution (10m per pixel) than the typical agricultural textural information such as furrows or tree rows. However, CNNs rely heavily on texture to extract spatial features [12]. Given this limitation, we propose to view medium-resolution images of agricultural parcels as unordered sets of pixels. Indeed, recent advances in 3D point cloud processing have spurred the development of powerful encoders for data comprised of sets of unordered elements [29, 45]. We show in this paper that set-based encoders can successfully extract learned statistics of the distribution of spectra across the spatial extent of the parcels. Furthermore, we show that this approach handles the highly-variable size of parcels in a more efficient way than CNNs.
Temporal Encoding of Satellite Time Series: Earlier work in crop classification has shown the importance of the temporal dimension when classifying crop types [34]. While RNNs have been widely used to analyse temporal sequences, recent work in Natural Language Processing (NLP) has introduced a promising new approach based on attention mechanisms [39]. The improved parallelism brought by this approach is particularly valuable for automated crop monitoring, as its typical scale spans entire continents: one year of Sentinel-2 observations amounts to 25 Tb of data for agricultural areas in the EU. Therefore, we propose an adapted attention-based approach for the classification of time series.

The key contributions of this paper are as follows:

- Inspired by Qi et al. [29], we introduce the pixel-set encoder as an efficient alternative to convolutional neural networks for medium-resolution satellite images.
- We adapted the work of Vaswani et al. [39] to an end-to-end, sequence-to-embedding setting for time series.
- We establish a new state-of-the-art for the task of large-scale agricultural parcel classification. Moreover, our method not only improves the classification precision by a significant margin, but simultaneously boosts a acceleration of over 4 times and a memory imprint reduced by over 70% compared to the best-performing approaches in the literature.
- We release the first open-access dataset of Sentinel-2 images for crop classification with ground truth labels.

2. Related Work

The problem of satellite image time series classification can be addressed at pixel level or object level. Pixel-based approaches do not require a priori knowledge of the borders of parcels, but cannot leverage the spatial homogeneity of class labels within the object’s extent. Conversely, in the case of crop classification, object-based approaches can leverage the parcels’ shape to extract helpful spatial information for achieving better classifications [9].

Traditional Machine Learning: Until recently, the common approach for crop classification has been to use traditional discriminative models with handcrafted features [41, 15, 42]. For instance, the Normalized Difference Vegetation Index (NDVI) combining the red and near-infrared spectral bands has been widely used as it relates to crop photosynthetic activity [38]. Certain work also includes phenological features derived from the study of the NDVI as well as external meteorological information [48]. Although robust and easily interpretable, such handcrafted indices do not compare favorably to end-to-end learned features.

In such work, the prevalent approach to represent temporal evolution is to concatenate each date’s spatial and spectral features. This is not well-suited to application over large geographical areas, in which the acquisition dates vary depending on the satellite orbit, and in which cloud cover and meteorological condition can be heterogeneous, resulting in sequences of variable length and temporal sampling. Consequently, other work oriented their efforts towards a better modeling of time using Hidden Markov Models [36], Conditional Random Fields [3], or Dynamic Time Warping [4].

Convolutional and Recurrent Approaches: More recently, the successful advances in the deep learning literature have provided efficient tools for both spatial and temporal feature extraction. Although some work only uses these tools as feature extractors [26], or combine them with feature engineering [46], most current work follows the deep learning paradigm of end-to-end trainable architec-
tutes. More specifically, Kussul et al. [20] proposed to use a Multi Layer Perceptron (MLP) on raw observation data instead of traditional RF of SVM. Further work sets out to leverage the spatial and temporal structures of time series of satellite images. CNNs [21] appeared to be a natural choice to address the spatial dimensions of the data [19, 32]. Similarly, Long-Short Term Memory (LSTM) networks [13] were successfully applied to model the temporal dimension of the data [30, 25], outperforming RF and SVM [14].

Furthermore, Rußwurm et al. [31] first proposed to use hybrid recurrent convolutional approach by applying the ConvLSTM architecture [44] to parcel classification. This work yielded state-of-the-art results and also showed that ConvLSTMs are able to learn to detect and ignore cloud obstruction. A similar approach was successfully used for automated change detection from Sentinel-2 data as well [27]. Finally, Garnot et al. showed in [34] that higher classification performance can be obtained by implementing such a hybrid model but with two dedicated modules for spatial and temporal feature extraction respectively: the series of images is first embedded by a shared CNN and the resulting embedding sequence is fed to a Gate Recurrent Unit (GRU) [8]. The use of a GRU is motivated by the smaller number of parameters required to achieve similar performance as LSTM, as corroborated in [32]. Additionally, Garnot et al. show that the relatively low spatial resolution of multi-temporal satellite images may question the relevance of CNNs since handcrafted descriptors of spectral distribution performed nearly as well as trainable spatial encoders when used in combination to the recurrent units. This is one of the issues we propose to address in the present study.

Attention-Based Approach: Following the adoption of self-attention in the NLP literature as an efficient alternative to RNNs, Rußwurm et al. proposed in [33] to apply the Transformer architecture [39]—a self-attention based network—to pixel-based classification. Their extensive experiments show that the Transformer yields classification performance that is on par with RNN-based models and present the same robustness to cloud-obstructed observations. Likewise, we propose to extend self-attention mechanisms to end-to-end sequence-to-embedding learning on images for object-level classification.

Purely Convolutional Approach: Multiple papers propose to address the temporal dimension with convolutions. Ji et al. present in [17] a spatio-temporal 3D-CNN for parcel-based classification, and spectro-temporal convolutions are found to outperform LSTMs for pixel-based segmentation on temporal profiles in [28], and outperform an MLP in [19]. Similar results are found in [49], where temporal convolutions yield better results than an LSTM network for classification based on NDVI temporal profiles. In addition, temporal convolutions have significantly lower processing times than RNNs. Yet, the ability to account for long-term dependencies requires deeper architectures. Furthermore, the fixed architecture of temporal CNN prevents the same network from being used on sequences of different lengths or with different acquisition dates.

Lastly, 2D and 3D convolutions have been extensively used in video analysis for object segmentation [5, 35] or action recognition [6, 11]. However, specificities of satellite time series such as their different time-scale and resolution prevents the direct application of such networks.

3. Methods

In this section, we present the different components of our proposed architecture for encoding time series of medium-resolution multi-spectral images. We denote the observations of a given parcel by a spatio-spectro-temporal tensor $[x^{(0)}, \ldots, x^{(T)}]_{T=1}^T$ of size $T \times C \times H \times W$, with $T$ the number of temporal observations, $C$ the number of spectral channels, and $H$ and $W$ the dimension in pixels of a tight bounding box containing the spatial extent of the parcel. All values are set to 0 outside the parcel’s borders, as shown in Figure 1.

3.1. Spatial Encoder

In recent years, CNNs have become the established approach to extract spatial features from images. However, our analysis suggests that convolutions may not be well-suited for the analysis of medium-resolution satellite images of agricultural parcels. Indeed, as mentioned above, the typical spatial resolution of satellites with high revisit frequency struggles to capture textural information. Second, efficiently training CNNs requires organizing the data into batches of images of identical dimensions. The irregular size of the parcels makes this process very memory intensive. Indeed, to limit textural information loss for large parcels, this amounts to oversampling most smaller parcels several times over.

To circumvent both these issues, we propose an alternative architecture called Pixel-Set Encoder (PSE) and inspired by the point-set encoder PointNet [29] and the DeepSet architecture [45] commonly used for 3D point cloud processing. The motivation behind this design is that, instead of textural information, the network computes learned statistical descriptors of the spectral distribution of the parcel’s observations.

The network proceeds as follows to embed an input observation $x^{(i)}$:

i) A set $\mathcal{S} \subset [1, \ldots, N]$ of $S$ pixels is randomly drawn from the $N$ pixels within the parcel, as described in Equation 1. When the total number of pixels in the image is smaller than $S$, an arbitrary pixel is repeated.
Figure 2: Schematic view of our spatio-temporal encoder. Variables in bold are tensors concatenated along the temporal dimension, e.g. \( e = [e^{(0)}, \ldots, e^{(T)}] \).

ii) Each sampled pixel \( s \) is processed by a shared multi-layer perceptron MLP\(_1\), as seen in Equation 2, composed of a succession of fully-connected layers, batch-norms [16], and Rectified Linear Units [24].

iii) The resulting set of values is pooled along the pixel axis—to obtain a vector capturing the statistics of the whole parcel and which is invariant by permutation of the pixels’ indices. We concatenate to this learned feature a vector of pre-computed geometric features \( f \): perimeter, pixel count \( N \), cover ratio (\( N \) divided by the number of pixels in the bounding box) and the ratio between perimeter and surface of the parcel.

iv) This vector is processed by another perceptron MLP\(_2\), as shown in Equation 3, to yield \( e^{(t)} \) the parcel’s spatio-spectral embedding at time \( t \).

The PSE architecture is represented in Figure 2, and can be summarized by the following equations:

\[
\begin{align*}
\mathcal{S} &= \text{sample} (S, N) \\
\hat{e}_s^{(t)} &= \text{MLP}_1 \left( x_s^{(t)} \right), \ \forall s \in \mathcal{S} \\
e^{(t)} &= \text{MLP}_2 \left( \text{pooling} \left( \{ \hat{e}_s^{(t)} \}_{s \in \mathcal{S}}, f \right) \right). \tag{3}
\end{align*}
\]

Among possible pooling operations, we had the best results for the concatenation of the mean and the standard deviation across the sampled pixel dimension \( S \). For parcels smaller than \( S \), repeated pixels should be removed before pooling to obtain unbiased estimates.

Although only a limited amount of information per parcel is used by this encoder, the sampling being different at each training step ensures the learning of robust embeddings exploiting all the available information.

### 3.2. Temporal Attention Encoder

RNNs have proven efficient for encoding sequential information [23]. However, since RNNs process the elements of the sequence successively, they prevent parallelization and incur long training times. In [39], Vaswani et al. introduce the Transformer architecture, an attention-based network achieving equal or better performance than RNNs on text translation tasks, while being completely parallelizable and thus faster. We propose to adapt their ideas to the the encoding of satellite image time series.

**Transformer Network:** In the original Transformer model a query-key-value triplet \( (q^{(t)}, k^{(t)}, v^{(t)}) \) is computed simultaneously for each element of the input sequence by three fully-connected layers. For a given element of a sequence, the key \( k^{(t)} \) conveys information about the nature of its content, while the value \( v^{(t)} \) encodes the content itself. The output of a given element is defined as the sum of the values of previous elements weighted by an attention mask. This mask is defined as the compatibility (dot product) of the keys of the previous elements with the query \( q^{(t)} \), re-scaled through a modified softmax layer. In other words, each element indicates which kind of information it needs through its query, and what sort of information it contains through its key.

Since the computation of the triplets \( (q^{(t)}, k^{(t)}, v^{(t)}) \) and their multiplications can be performed in parallel, the Transformer takes full advantage of modern GPU architecture and yields a significant speed increase compared to recurrent architectures. This procedure can be computed several times.
times in parallel with different set of independent parameters, or heads. This approach, called multi-head attention, allows for the specialization of different set of query-key compatibility.

**Positional Encoding:** In their paper on text translation, Vaswani et al. add order information to elements of the input sequence by adding a positional encoding tensor to each element. Equation 4 describes this positional encoding of the observation $t$, with $d_e$ the dimension of the input, and $i$ the coordinates of the positional encoding. Since our considered sequences are typically shorter than the ones considered in NLP, we chose $\tau = 1000$—instead of 10000. Additionally, $\text{day}(t)$ is the number of days since the first observation for observation $t$ instead of its index. This helps to account for inconsistent temporal sampling (see Figure 1).

\[
[p(t)]_{i=1}^{d_e} = \sin \left( \text{day}(t) \sqrt{\frac{2}{\pi}} + \frac{\pi}{2} \mod(i, 2) \right) \tag{4}
\]

**End-to-End Encoding:** The original Transformer network takes pretrained word embeddings as inputs. In our setting however, the parameters of the network producing the inputs are learnt simultaneously to the attention parameters. Therefore, we propose that each head only computes key-query pairs from the spatial embeddings (5) since these embeddings can directly serve as values: $e^{(h)} = e^{(t)} + p^{(t)}$. This removes needless computations, and avoids a potential information bottleneck when computing the values.

**Sequence-to-Embedding Attention:** While the original Transformer produces an output for each element of a sequence, our goal is to encode an entire time series into a single embedding. Consequently, we only retain the encoder part of the Transformer and define a single master query $q_h$ for each head $h$. Such a query, in combination with the keys of the elements of the sequence, determines which dates contain the most useful information. A first approach would be to select the query of a given date, such as the last one. However, the selected element of the sequence may not contain enough information to produce a meaningful query. Instead, we propose to construct the master query as a temporal average of the queries of all dates and processed by a single fully-connected layer (6). As shown in Equation 7, this query is then multiplied with the keys of all elements of the sequence to determine a single attention mask $a^{(h)} \in [0, 1]^T$, in turn weighting the input sequence of embeddings (8).

\[
\text{Multi-Head Self-Attention: } \text{We concatenate the output } o_h \text{ of each head } h \text{ for the } H \text{ different heads and process the resulting tensor with } \text{MLP}_3, \text{ to obtain the final output } \hat{o} \text{ of the Temporal Attention Encoder (TAE), as shown in Equation 9. Note that unlike the Transformer network, we directly use } \hat{o} \text{ as the spatio-temporal embedding instead of using residual connections. We show an illustration of head specialization on Figure 3 plotting the average attention masks for two classes of cereal parcels, after training a TAE on the dataset described in Section 4. We can see that each of the four heads specializes on a specific portion of the time series. Additionally, the dependency of the attention mask to the input is especially clear for head 4 which focuses on late spring for Spring Cereal samples and late summer for Summer Cereal samples.}

**Temporal Attention Encoder:** For each head $h$, we denote by $\text{FC}_1^{(h)}$ the fully-connected layer generating the key-query pairs, $\text{FC}_2^{(h)}$ the fully-connected layer yielding the master query, and $d_k$ the shared dimensions of keys and queries. Our attention mechanism can be summarized by the following equations for all $t \in [1, \ldots, T]$ and $h \in [1, \ldots, H]$:

\[
k_h^{(t)} \cdot q_h^{(t)} = \text{FC}_1^{(h)} \left( e^{(t)} + p^{(t)} \right) \tag{5}
\]

\[
\hat{q}_h = \text{FC}_2^{(h)} \left( \text{mean}_{T} \{ q_h^{(t)} \}_{t=1}^{T} \right) \tag{6}
\]

\[
a_h = \text{softmax} \left( \frac{1}{\sqrt{d_k}} \left[ \hat{q}_h \cdot k_h^{(t)} \right]_t \right) \tag{7}
\]

\[
o_h = \sum_{t=1}^{T} a_h[t] \left( e^{(t)} + p^{(t)} \right) \tag{8}
\]

\[
\hat{o} = \text{MLP}_3 ([o_1, \ldots, o_H]) \tag{9}
\]

**3.3. Spatio-Temporal Classifier**

Our spatio-temporal classifier architecture combines the two components presented in the previous sections: all input images of the time series are embedded in parallel by...
a shared PSE, and the resulting sequence of embeddings is processed by the temporal encoder, as illustrated in Figure 2. Finally, the resulting embedding is processed by an MLP decoder to produce class logits: $y = MLP_4(\hat{a})$.

### 3.4. Implementation details

All the architectures presented here are implemented in PyTorch, and released on GitHub upon publication\(^1\). We trained all models on a machine with a single GPU (Nvidia 1080Ti) and an 8-core Intel i7 CPU for data loading from an SSD hard drive. The exact configuration of our network is displayed in Table 1. We chose the hyperparameters of each architecture presented in the numerical experiments such that they all have approximately 160k trainable parameters (see appendix). We use the Adam optimizer [18] with its default values ($lr = 10^{-3}$, $\beta = (0.9, 0.999)$) and a batch size of 128 parcels. We train the models with focal loss [22] ($\gamma = 1$) and implement a 5-fold cross-validation scheme: for each fold the dataset is split into train, validation, and test set with a 3:1:1 ratio. The networks are trained for 100 epochs, which is sufficient for all models to achieve convergence. We use the validation step to select the best-performing epoch, and evaluate it on the test set. For augmentation purpose, we add a random Gaussian noise to $x^{(t)}$ with standard deviation $10^{-2}$ and clipped to $5 \cdot 10^{-2}$ on the values of the pixels. Lastly, the image time series are normalized channel-wise and for each date individually.

### 4. Numerical Experiments

#### 4.1. Dataset

We evaluate our models using Sentinel-2 multi-spectral image sequences in top-of-canopy reflectance. We leave out the atmospheric bands (bands 1, 9, and 10), keeping $C = 10$ spectral bands. The six 20m-resolution bands are resampled to the maximum spatial resolution of 10m.

The area of interest (AOI) corresponds to a single tile of the Sentinel-2 tiling grid (T31TFM) in southern France. This tile provides a challenging use case with a high diversity of crop types and different terrain conditions. The AOI spans a surface of 12 100 km\(^2\) and contains 191 703 individual parcels, all observed on 24 dates from January to October 2017. The values of cloudy pixels are linearly interpolated from the first previous and next available pixel using Orfeo Toolbox [7].

We retrieve the geo-referenced polygon and class label of each parcel from the French Land Parcel Identification System records\(^2\). We crop the satellite images using this polygon to constitute the image time series.

**Data Preparation:** In order to evaluate both ours and convolution-based methods, we organize the parcels into two different formats: patches and pixel sets.

In the **patch** format, we resize each parcel into a tensor of size $T \times 32 \times 32$ by interpolating each spectral channel and temporal acquisition independently into patches of fixed size $32 \times 32$. We use nearest neighbor interpolation, and both the horizontal and vertical axes are rescaled so that the overall shape of the parcel may be altered. We use zero-padding outside the extent of the parcel (see Figure 1). This same size of 32 pixels was used in [34], while a larger $48 \times 48$ patch size was used in [31], albeit for a pixel-wise classification task.

For the **pixel-set** format, the pixels of each parcels are stored in arbitrary order into a tensor of size $T \times C \times N$, with $N$ the total number of pixels in a given parcel. Note that this format neither lose nor create information, regardless of parcel size. Hence, this setup saves up to 70% disk space compared to the patch format (28.6Gb vs. 98.1Gb). Note that the geometric features $f$ must be computed and saved before preparing the dataset, as all spatial structure is henceforth lost.

The classification labels are defined with respect to a 20 class nomenclature designed by the subsidy allocation authority of France. The dataset is highly imbalanced: four classes cover 90% of the samples (see Figure 3 in the appendix). This motivated the use of the focal loss to train our models.

Both datasets will be released upon publication\(^1\). To the best of our knowledge, no benchmark dataset currently exists for object-based agricultural parcel classification. Our datasets are a first step towards more reproducible and comparable methodological work in this field.

<table>
<thead>
<tr>
<th>Modules</th>
<th>Hyperparameters</th>
<th>Number of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSE</td>
<td></td>
<td>19 936</td>
</tr>
<tr>
<td>S</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>MLP(_1)</td>
<td>10 $\rightarrow$ 32 $\rightarrow$ 64</td>
<td></td>
</tr>
<tr>
<td>MLP(_2)</td>
<td>132 $\rightarrow$ 128</td>
<td></td>
</tr>
<tr>
<td>TAE</td>
<td></td>
<td>136 192</td>
</tr>
<tr>
<td>$d_x$, $d_k$, $H$</td>
<td>128, 32, 4</td>
<td></td>
</tr>
<tr>
<td>FC(_1)</td>
<td>128 $\rightarrow$ (32 $\times$ 2)</td>
<td></td>
</tr>
<tr>
<td>FC(_2)</td>
<td>32 $\rightarrow$ 32</td>
<td></td>
</tr>
<tr>
<td>MLP(_3)</td>
<td>512 $\rightarrow$ 128 $\rightarrow$ 128</td>
<td></td>
</tr>
<tr>
<td>Decoder</td>
<td></td>
<td>11 188</td>
</tr>
<tr>
<td>MLP(_4)</td>
<td>128 $\rightarrow$ 64 $\rightarrow$ 32 $\rightarrow$ 20</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>164 116</td>
</tr>
</tbody>
</table>

Table 1: Configuration of our model chosen for the numerical experiments. The dimension of each successive feature space is given for MLPs and fully connected layers.
4.2. Comparison with State-of-the-Art Methods

Competing Methods: We compare our approach to recent algorithms operating on similar datasets, which we have re-implemented. The different hyperparameters chosen for each model are shown in the appendix. All share the same decoding layer configuration MLP₄.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>OA</th>
<th>mIoU</th>
<th>Training (s/epoch)</th>
<th>Inference (s/dataset)</th>
<th>Disk Size Gb</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSE+TAE (ours)</td>
<td>94.2 ±0.1</td>
<td>50.9 ±0.8</td>
<td>158</td>
<td>149</td>
<td>28.6 / 12.3¹</td>
</tr>
<tr>
<td>CNN+GRU [34]</td>
<td>93.8 ±0.3</td>
<td>48.1 ±0.6</td>
<td>656</td>
<td>633</td>
<td>98.1</td>
</tr>
<tr>
<td>CNN+TempCNN [28]</td>
<td>93.3 ±0.2</td>
<td>47.5 ±1.0</td>
<td>635</td>
<td>608</td>
<td>98.1</td>
</tr>
<tr>
<td>Transformer [33]</td>
<td>92.2 ±0.3</td>
<td>42.8 ±1.1</td>
<td>13</td>
<td>420 + 4³</td>
<td>28.6 / 0.22⁴</td>
</tr>
<tr>
<td>ConvLSTM [31]</td>
<td>92.5 ±0.5</td>
<td>42.1 ±1.2</td>
<td>1283</td>
<td>666</td>
<td>98.1</td>
</tr>
<tr>
<td>Random Forest [3]</td>
<td>91.6 ±1.7</td>
<td>32.5 ±1.4</td>
<td>293²</td>
<td>420 + 4³</td>
<td>28.6 / 0.44⁴</td>
</tr>
</tbody>
</table>

Table 2: Classification metrics and time benchmark of the different architectures. The inter-fold standard deviation of the OA and mIoU is given in smaller font. Additionally, the total time for one epoch of training, and for inference on the complete dataset are given on the third and fourth columns. ¹ disk space required for training and pure inference, ² time for the entire training step, ³ preprocessing and inference time, ⁴ dataset before and after preprocessing.

Random Forest Lastly, we use a Random Forest classifier with 100 trees as a non-deep learning baseline. The classifier operates on handcrafted features comprised of the mean and standard deviation of each band within the parcel, and concatenated along the temporal axis, as described by [3].

We present the results of our experiments in Table 2. Our proposed architecture outperforms the other deep learning models in Overall Accuracy (OA) by 0.4 points, and mean per-class Intersection over Union (mIoU) by 3 to 9 points. It also provides a four-fold speed up over convolution-based methods, and a decrease in disk usage of over 70% for training, and close to 90% when considering the inference task alone (i.e. when only S pixels per parcels are kept). This speed-up is due both to improved loading time as the pixel set dataset is smaller, but also inference and backpropagation time, as detailed in Table 2 of the appendix. While the temporal convolutions of TempCNN are faster to train, they yield worse performance and suffer from the limitations discussed in section 2. The Transformer method, which processes pre-computed parcel means, is also faster to train, but only achieves a 42.8 mIoU score.

Beyond its poor precision, the RF classifier has a significant speed and memory advantage. This can explain its persisting popularity among practitioners. However, our approach bridges in part this performance gap and provides much higher classification rates, making it a compelling strategy for large-scale object-based crop type mapping.

4.3. Ablation Studies

In order to independently assess the contribution of the spatial and temporal components of our proposed architecture, we present in Table 3 the results obtained when alternatively replacing the PSE by a CNN (CNN+TAE) or the TAE by a GRU (PSE+GRU).

Contribution of the PSE: As seen in Table 3, the PSE accounts for an increase of 1.7 points of mIoU compared to the CNN-based model (CNN+TAE). This supports both the
hypothesis that CNNs are only partly relevant on medium-resolution images, and that considering the image as an unordered set of pixels is a valid alternative. Not only does this approach yield better classification performance, but it also circumvents the problem of image batching, which leads to faster data loading. Additionally, we train a TAE on pre-computed means and standard deviations of the spectral channels over the parcels (MS+TAE), which achieves a 48.9 mIoU score. We can thus conclude that the PSE learns statistical descriptors of the acquisitions’ spectra which are more meaningful than simple means and variances or convolutional features.

**Design of the PSE:** We show in Table 3, the performance of our architecture without geometric features \( f \). The resulting 0.9 point decrease in mIoU confirms that geometric information plays a role in the classification process. We note that, even without such features, our proposed approach outperforms the convolution-based model (CNN+TAE).

We have tried replacing the handcrafted geometric features \( f \) with a CNN operating on the binary mask of the parcel. However, the gains were minimal, and we removed this extra step for simplicity’s sake.

Lastly, we tried training our architecture with a reduced number of sampled pixels \( (S = 16, S = 32) \). The model maintains a good performance with an mIoU over 50 points. This indicates that the decrease in processing time and memory could be further improved at the cost of a minor drop in precision.

**Contribution of the TAE:** Replacing the temporal attention encoder with a GRU (PSE+GRU) decreases the performance by 3.6 points of mIoU (Table 3). The TAE not only produces a better classification but also trains faster thanks to parallelization. Unlike the comparison between Transformer and RNNs architectures in [33], our modified self-attention mechanism extracts more expressive features than the RNN-based approach.

We also evaluate the influence of the positional encoding \( p \) of the Transformer by adding \( p \) to the input tensors of the GRU unit (PSE+GRU+p). This reduces the gap with our method to 2.2 points of mIoU. This shows that the improvement brought by the TAE is due to both its structure and the use of a positional encoding.

**Design of the TAE:** In order to evaluate the benefits of our different contributions over the Transformer, we adapted the architecture presented in [33] to use a PSE network instead of spectral means for embedding parcels (PSE+Transformer), for a performance 4.3 points below our TAE. By replacing the proposed temporal max-pooling by our master query forming scheme \( (\text{PSE+Transformer}+\hat{q}) \), we observed an increase of 2.9 points of mIoU. The remaining 1.4 mIoU points between this implementation and ours can thus be attributed to our direct use of inputs to compute the TAE’s output instead of a smaller intermediary value tensor.

Finally, we compare our mean-pooling strategy with max-pooling \((\hat{q} = \max q^{(t)}\) and computing the master query from the last element of the sequence \((\hat{q} = q^{(T)})\). While the mean query approach yields the best performance, the last element of the sequence in our dataset produces a meaningful query as well. However, this may not be the case for other regions or acquisition years.

### 5. Conclusion

In this paper, we considered the problem of object-based classification from time series of satellite images. We proposed to view such images as unordered sets of pixels to reflect the typical coarseness of their spatial resolution, and introduced a fitting encoder. To exploit the temporal dimension of such series, we adapted the Transformer architecture [39] for embedding time-sequences. We introduced a master query forming strategy, and exploited that our network learns end-to-end to simplify some operations.

Evaluated on our new open-access annotated benchmark of agricultural parcels, our method produces a better classification than all other re-implemented methods. Furthermore, our network is several times faster and more parsimonious in memory than other state-of-the-art methods such as convolutional-recurrent hybrid networks. We hope that by mitigating some of the limitations of deep learning methods such as processing time and memory requirement, our approach would accelerate their adoption in real-life, large-scale Earth observation applications.

Our results suggest that attention-based models are an interesting venue to explore for analysing the temporal profiles of satellite time series, as well as other analogous vision tasks such as action recognition in videos. Likewise, set-based encoders are a promising and overlooked paradigm for working with the coarser resolutions of remote sensing applications.

<table>
<thead>
<tr>
<th>Design of the PSE:</th>
<th>O.A.</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSE+Transformer+(q)</td>
<td>94.1 ±0.2</td>
<td>49.5 ±0.7</td>
</tr>
<tr>
<td>CNN+TAE</td>
<td>94.0 ±0.1</td>
<td>49.2 ±1.1</td>
</tr>
<tr>
<td>MS+TAE</td>
<td>93.7 ±0.1</td>
<td>48.9 ±0.9</td>
</tr>
<tr>
<td>PSE+GRU+p</td>
<td>93.6 ±0.2</td>
<td>48.7 ±0.3</td>
</tr>
<tr>
<td>PSE+GRU</td>
<td>93.6 ±0.2</td>
<td>47.3 ±0.3</td>
</tr>
<tr>
<td>PSE+Transformer</td>
<td>93.4 ±0.2</td>
<td>46.6 ±0.9</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of our different design choices, sorted by decreasing mIoU.
References


