Learning from few labeled time series with segment-based self-supervised learning: application to remote-sensing

Antoine Saget*[1], Baptiste Lafabregue[1], Antoine Cornuéjols[2], Pierre Gançarski[1]

[1] *ICube, Strasbourg, France*
[2] *UMR MIA-Paris, AgroParisTech, INRAE - Université Paris-Saclay, Palaiseau, France*

Taking advantage of unlabeled data remains a major challenge with current classification methods in most domains, including remote sensing. To address this issue, we introduce a novel method for selecting positive pairs in contrastive self-supervised learning (SSL) and apply it to remote sensing time series classification. Using preexisting groups (i.e., segments) within the data, our approach eliminates the need for strong data augmentations required in contrastive SSL. The learned representations can be used on downstream classification tasks with simple linear classifiers. We show that it achieves comparable performance to state-of-the-art models while requiring nearly half as much labeled data. We achieve 80% accuracy on a 20-class classification task with 50 labeled samples per class, while the best compared method requires 100. We experimentally validate our method on a new large-scale Sentinel-2 satellite image time series dataset for cropland classification.

1 Introduction

Satellite Images Time Series (SITS) are now widely available worldwide with high temporal and spatial resolutions [1]. Such data is valuable for applications like monitoring natural disasters and urban planning [2]. In this article, we focus on cropland classification. Although current state-of-the-art machine learning methods yield reliable results, they typically require large amounts of labeled data [3] while the daily volume of unlabeled data generated by satellites far exceeds what can be manually labeled, leaving most of these data unexploited by these methods.

Self-Supervised Learning (SSL) learn new representation from unlabeled data as it only uses information generated from the data itself. These representations are applicable to various downstream tasks and offer greater label efficiency than standard supervised methods [4]. Thus, SSL emerges as a natural candidate for SITS classification, where there is an abundance of unlabeled data but a shortage of labeled data.

*Corresponding author. E-Mail: antoine.saget@unistra.fr

SSL methods, primarily based on contrastive or generative reconstruction losses (see Section 2.2), have achieved great success in computer vision [5–9]. Building on this success, specific SSL methods for time series have been proposed [10–15]. Although both generative reconstruction-based SSL and contrastive SSL methods are promising for SITS, in this paper we focus exclusively on contrastive SSL.

In contrastive SSL, positive examples are similar datapoints, typically created through data augmentation. However, generating positive examples is challenging for SITS data because strong data augmentations commonly used in image processing cannot be directly applied to time series. Instead, we propose using preexisting groupings of the data (i.e., segments) as positive examples to adapt contrastive SSL for SITS.

The remainder of this paper is organized as follows:

- In Section 2, we introduce time series classification and self-supervised learning.
- In Section 3, we present a new large-scale SITS crop classification dataset.
- In Section 4, we detail our proposed method and motivate that using preexisting data groupings, in our case the bounds of agricultural parcels, is an effective alternative to augmentations for selecting positive examples in contrastive SSL.
- In Section 5, we show experimental results. The learned representation achieves a similar performance to the best compared method using almost half the number of training labels.

2 Related works

2.1 Time Series Classification

In [16], Ruiz et al. benchmark the performance of 16 multivariate time series classification algorithms on 26 datasets from the UEA archive [17]. These algorithms include a variety of approaches including shapelet-based method, dynamic time warping, frequent pattern mining, and convolutional techniques.

In ROCKET [18], thousands of random convolutional kernels are applied to the input time series to generate a new representation on which the classification is performed. HIVE-COTE [19], is an ensemble of four different classifiers, one of them being an ensemble of ROCKET classifiers. Despite reaching state-of-the-art results, especially with HIVE-COTE 2 [20], it is orders of magnitude slower than other methods, making it almost unusable for large datasets. In InceptionTime [21, 22], Fawaz et al. adapt the Inception-v4 architecture for timeseries data and reach results similar to HIVE-COTE while being significantly faster.

2.2 Self-Supervised Learning

Self-supervised learning (SSL) extracts useful features from unlabeled data by learning from autogenerated labels through a dedicated task (often called a pretext task). The learned representation can then be used on various downstream tasks. Currently, two pretext task designs are prevalent in the field:

- **Generative** SSL involves masking parts of the input data and training a model to reconstruct these masked portions. An example is Masked Auto-Encoders (MAE) [23].
- **Contrastive** SSL trains a model to minimize the distance between pairs of similar examples (positive) while maximizing the distance between pairs of dissimilar examples (negative).

Contrastive SSL achieved notable success in image data [5–8]. However, these methods typically rely on heavy data augmentation [5] like cropping and color jitter, which are not easily transferable to other data types, such as time series. Moreover, designing universal augmentations for time series data is challenging due to their diversity. For example, SITS, electrocardiograms, and stock prices each have distinct characteristics, sampling frequencies, and lengths.

In SimCLR [5], positive examples are generated by applying different augmentations to the same sample, while negative examples are other samples in the batch. SimCLR advocates for strong augmentations, which go beyond those typically used in supervised learning. It also emphasizes the use of large batch sizes to provide enough negative examples to the model and prevent collapse. We refer the reader to [9] for a comprehensive review of contrastive representation learning.

In TimeMAE [15], a transformer model tailored for time series is jointly trained on a reconstruction task and a classification task and shows strong results compared to various other SSL time series methods, including TNC [12], TS-TCC [14], and TS2Vec [11].

3 Dataset

We created a new large-scale SITS cropland classification dataset inspired by [3], featuring over 5.8 million labeled parcels in Metropolitan France. This fully labeled dataset allows us to evaluate the benefits of SSL in a controlled environment. By artificially withholding most labels during training, we can simulate datasets with genuinely scarce labels while still evaluating on a large quantity of labeled data. Details on dataset construction are provided next.

3.1 Labels: France Agricultural Land Parcel Information System

The Agricultural Land Parcel Information System or RPG (Registre Parcellaire Graphique) provided annually by the French National Institute of Forest and Geography Information (IGN), lists all cultivated parcels in France along their geometries and crop types. For our dataset, we used the RPG crop types of 2022 in metropolitan France as class labels. This results in a highly imbalanced dataset with 232 classes. As shown in Figure 1, this imbalance already spans an order of magnitude within the 10 most common crops.

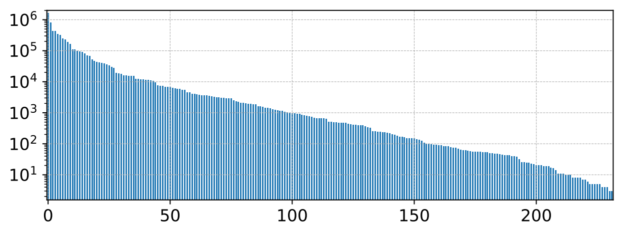

Figure 1: Repartition of crop types in the 2022 France RPG. Logarithmic scale on the y-axis.

3.2 Features: Sentinel-2 time series

For each parcel, 100 pixel time series are randomly sampled within its bounds. Parcels with fewer than 100 pixels (less than one hectare) are discarded. Each time series spans from February 1, 2022, to November 30, 2022. A simple preprocessing is performed on each time series. First, any time step identified as cloudy by the MSK_CLDPRB and SCL Sentinel-2 L2A channels is discarded. Second, each time series is reindexed and interpolated, resulting in mostly cloudless, temporally aligned time series of equal lengths. Finally, we apply the 2-98% min/max normalization of [1] followed by a $2x-1$ transformation, which scales the time series to a range of mostly $[-1,1]$.

In summary, our dataset comprises approximately 5.8 million parcels, each labeled as one of 232 possible classes. Each parcel consists of 100 distinct time series of 60 time steps across the 12 Sentinel-2 L2A radiometric bands.

4 Leveraging preexisting groups within the data

In domains where data augmentation remains a challenge, we think that identifying preexisting groups within the data to select positive pairs is a viable alternative. We refer to this approach as "Groups as Positive Pairs" or GaPP. We also use these groups by passing several samples from the same group through the encoder and aggregating their resulting representations with an average pooling layer. We simply refer to this as "Average Pooling" or AvgP.

In this article, we use pixels time series sampled from the same agricultural parcel as preexisting groups. Such groups are particularly suitable because, while each pixel time series from the same parcel belongs to the same class, they are still unique and different from each other, as illustrated in Figure 2. Replicating this diversity through classical augmentations is not trivial and would require careful parameter tuning and expert knowledge.

Our proposed approach is applicable to any contrastive SSL method. In Section 5, we present experiments using SimCLR [5].

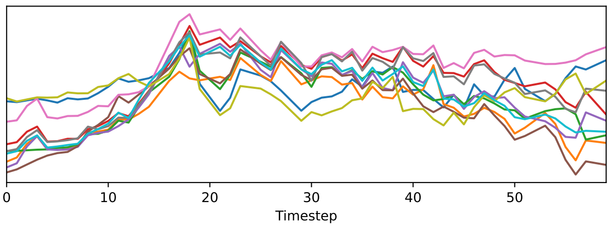

Figure 2: 10 Sentinel-2 pixels time series (B08 band) sampled within the same parcel (pastured woodlands class).

5 Experiments and results

5.1 Splits

We selected a subset of 20 classes of interest to simulate an expert providing hand-labeled examples. We divided the dataset into four splits:

- ***contrast***: approximately 4 million unbalanced and unfiltered samples. This split is used during SSL and does not include any labels.
- ***train***: a labeled subset of the *contrast* split, balanced and filtered to include only samples from the 20 classes. The number of samples in this set varies across experiments.
- ***validation***: labeled, balanced, and filtered to the 20 classes, with 10 samples per class.
- ***test***: 20k samples, filtered to the 20 classes but unbalanced as in the true data distribution.

The *contrast*, *validation*, and *test* sets are mutually exclusive.

5.2 Training and evaluation protocols

Models are trained and evaluated as follows:

- **Standard methods** are trained on the *train* split and evaluated on the *test* split. The *validation* split is used for early stopping and hyperparameter tuning when applicable.
- **SSL methods** are trained on the *contrast* split with the *validation* split for early stopping and hyperparameter tuning. A logistic regression is then trained using the new representation inferred by the SSL model on the *train* split and evaluated on the *test* split.

Model predictions are evaluated on the parcel's class, obtained by a majority vote on the 100 pixel time series within each parcel.

Using a labeled validation set during SSL training deviates from a fully unsupervised setting. However, during experimentations, we observed instances of model collapse that were undetectable through the training loss alone, but identifiable with a small labeled validation set. Thus, we incorporated this validation set to stop the model before collapse. Occurrences of collapse suggest that our augmentation alternatives are not robust enough and require further improvements to work in a fully unsupervised setting.

5.3 Compared methods

We list all compared methods and parameters. Logistic Regression uses the cuml [24] implementation with default parameters. Both MiniRocket [25] and InceptionTime [21] use the aeon [26] implementation. We report only MiniRocket results, as it outperformed ROCKET [18], MultiRocket [27], and Hydra-MultiRocket [28] on our dataset. InceptionTime is configured with a batch size of 512 and 100 epochs. These changes sped up training without affecting the validation accuracy. TimeMAE [15] uses default parameters except vocab size 64, wave length 6, alpha 1, depth model 512, and batch size 512. SimCLR [5] employs a time series ResNet encoder [29] from tsai [30] with width ×4 and Lightly [31] implementation, optimized with SGD, learning rate 0.02, momentum 0.9, weight decay 5×10^{-4}, and batch size 256.

5.4 Results

All results are average classification accuracies over 20 repetitions with different subsets of the train set. Standard deviations are below 4 (resp. 2) on training sets with 5 or 10 (resp. 50 or 100) samples per class. We omitted them, as they are very similar across methods and only different across training set sizes.

5.4.1 Label efficiency

In Table 1, we show test accuracies for different training set sizes. The main benefit of our approach is label efficiency. For example, our method with 50 labeled parcels per class achieves 80% accuracy, the same as InceptionTime with 100.

Method	N sample/class in train set 5	10	50	100
Logistic Regression	42	49	68	74
MiniRocket	47	58	75	79
InceptionTime	56	65	77	80
TimeMAE	53	61	75	79
SimCLR + GaPP + AvgP	62	70	80	82

Table 1: Models performances with various train set sizes (See Section 4 for GaPP/AvgP).

5.4.2 Ablation

Table 2 shows that best accuracy is achieved when combining both average pooling (AvgP) and using samples from the same parcel as positive pairs (GaPP).

5.4.3 Impact of number of averaged timeseries

Table 3 shows the average pooling layer performs best when four time series are averaged. This hyperparameter might be dataset dependent and requires further experiments to assess its sensitivity.

Scenario	N samples/class in train set 5	10	50	100
Noise only	45	53	70	75
Noise + AvgP	45	53	70	75
Noise + GaPP	50	60	76	80
Noise + GaPP + AvgP	62	70	80	82

Table 2: Performance of our method when ablating its components. Noise correspond to Gaussian noise ($\mu = 0$, $\sigma = 0.1$) added to the input time series (See Section 4 for GaPP/AvgP).

N samples averaged	N samples/class in train set 5	10	50	100
2	56	66	78	82
4	62	70	80	82
8	61	69	80	82

Table 3: Performance of our method with varying number of timeseries aggregated in the average pooling layer.

6 Discussion

In this paper, we proposed an alternative to augmentations in contrastive SSL when preexisting groups within the data are available. Experiments on a new large-scale dataset show nearly double the label efficiency on a downstream classification task compared to other methods. Nonetheless, several questions remain open for future work:

- A small validation dataset is currently required to prevent model collapse. Further works investigating how small this validation dataset can be or if it can be completely discarded are needed.
- Although SSL is known for producing general representations, we only evaluated our method on a single downstream classification task. Further research should test the versatility of the learned representations in various tasks.
- We used parcels bounds as groups in our experiments. It would be valuable to test our approach on other datasets with different groupings or to create synthetic groups through segmentation or clustering techniques.
- We only focused on contrastive SSL, but the use of group information in generative SSL remains to be explored.

7 Acknowledgments

The authors thank the French National Research Agency (ANR) for funding the Ph.D. work through the ANR ArtIC and supporting this research as part of the ANR HERELLES. The authors also express their gratitude to the European Space Agency (ESA) and the Copernicus program for making Sentinel-2 data freely accessible to everyone.

References

1. Pelletier, C., Webb, G. I. & Petitjean, F. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing* **11,** 523 (2019).
2. Casagli, N., Intrieri, E., Tofani, V., Gigli, G. & Raspini, F. Landslide detection, monitoring and prediction with remote-sensing techniques. *Nature Reviews Earth & Environment* **4,** 51–64 (2023).
3. Rußwurm, M., Pelletier, C., Zollner, M., Lefèvre, S. & Körner, M. Breizhcrops: A time series dataset for crop type mapping. *arXiv preprint arXiv:1905.11893* (2019).
4. Henaff, O. *Data-efficient image recognition with contrastive predictive coding* in *International conference on machine learning* (2020), 4182–4192.
5. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. *A simple framework for contrastive learning of visual representations* in *International conference on machine learning* (2020), 1597–1607.
6. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. *Momentum contrast for unsupervised visual representation learning* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), 9729–9738.
7. Grill, J.-B. *et al.* Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems* **33,** 21271–21284 (2020).
8. Chen, X. & He, K. *Exploring simple siamese representation learning* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), 15750–15758.
9. Le-Khac, P. H., Healy, G. & Smeaton, A. F. Contrastive representation learning: A framework and review. *Ieee Access* **8,** 193907–193934 (2020).
10. Kiyasseh, D., Zhu, T. & Clifton, D. A. *Clocs: Contrastive learning of cardiac signals across space, time, and patients* in *International Conference on Machine Learning* (2021), 5606–5615.
11. Yue, Z. *et al. Ts2vec: Towards universal representation of time series* in *Proceedings of the AAAI Conference on Artificial Intelligence* **36** (2022), 8980–8987.
12. Tonekaboni, S., Eytan, D. & Goldenberg, A. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750* (2021).
13. Lan, X., Ng, D., Hong, S. & Feng, M. *Intra-inter subject self-supervised learning for multivariate cardiac signals* in *Proceedings of the AAAI Conference on Artificial Intelligence* **36** (2022), 4532–4540.
14. Eldele, E. *et al.* Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112* (2021).
15. Cheng, M. *et al.* Timemae: Self-supervised representations of time series with decoupled masked autoencoders. *arXiv preprint arXiv:2303.00320* (2023).
16. Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M. & Bagnall, A. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* **35,** 401–449 (2021).
17. Bagnall, A. *et al.* The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).
18. Dempster, A., Petitjean, F. & Webb, G. I. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* **34,** 1454–1495 (2020).
19. Lines, J., Taylor, S. & Bagnall, A. *Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification* in *2016 IEEE 16th international conference on data mining (ICDM)* (2016), 1041–1046.
20. Middlehurst, M. *et al.* HIVE-COTE 2.0: a new meta ensemble for time series classification. *Machine Learning* **110,** 3211–3243 (2021).
21. Ismail Fawaz, H. *et al.* Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* **34,** 1936–1962 (2020).
22. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J. & Forestier, G. *Lite: Light inception with boosting techniques for time series classification* in *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)* (2023), 1–10.
23. He, K. *et al. Masked autoencoders are scalable vision learners* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), 16000–16009.
24. Raschka, S., Patterson, J. & Nolet, C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information* **11,** 193 (2020).
25. Dempster, A., Schmidt, D. F. & Webb, G. I. *Minirocket: A very fast (almost) deterministic transform for time series classification* in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (2021), 248–257.
26. *aeon - A scikit-learn compatible toolkit for time series* 2023. https://github.com/aeon-toolkit/aeon.
27. Tan, C. W., Dempster, A., Bergmeir, C. & Webb, G. I. MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery* **36,** 1623–1646 (2022).
28. Dempster, A., Schmidt, D. F. & Webb, G. I. Hydra: Competing convolutional kernels for fast and accurate time series classification. *Data Mining and Knowledge Discovery* **37,** 1779–1805 (2023).
29. Wang, Z., Yan, W. & Oates, T. *Time series classification from scratch with deep neural networks: A strong baseline* in *2017 International joint conference on neural networks (IJCNN)* (2017), 1578–1585.
30. Oguiza, I. *tsai - A state-of-the-art deep learning library for time series and sequential data* Github. 2023. https://github.com/timeseriesAI/tsai.
31. Susmelj, I., Heller, M., Wirth, P., Prescott, J. & et al., M. E. *Lightly* 2020. https://github.com/lightly-ai/lightly.