HAL open science

Learning spatio-temporal representations of satellite time series for large-scale crop mapping

Vivien Sainte Fare Garnot

To cite this version:

Vivien Sainte Fare Garnot. Learning spatio-temporal representations of satellite time series for large-scale crop mapping. Signal and Image processing. Université Gustave Eiffel, 2022. English. NNT : 2022UEFL2006 . tel-03783589

HAL Id: tel-03783589

https://theses.hal.science/tel-03783589v1

Submitted on 22 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning Spatio-temporal Representations of Satellite Time Series for Large-scale Crop Mapping.

Thèse de doctorat de l'Université Gustave Eiffel

École doctorale n° 532, Mathématiques, Science, et Technologie de l'Information et de la Communication (MSTIC)
Spécialité de doctorat : Signal, Image, et Automatique
Unité de recherche : Laboratoire des Sciences et Technologies de l'Information Geographique (LASTIG), IGN.

Thèse présentée et soutenue à l'Université Gustave Eiffel, le 07/01/2022, par

Vivien SAINTE FARE GARNOT

Composition du Jury

Florence TUPIN Professeur, Telecom ParisTech, Department of Image and Signal Processing, France	Présidente
Jocelyn CHANUSSOT Professeur, Univ. Grenoble Alpes, Inria, France	Rapporteur
Konrad SCHINDLER Professeur, Institut de Géodesie et Photogrammetrie, ETH Zurich, Suisse	Rapporteur
Patrick PEREZ Directeur Scientifique, Valeo.ai, France	Examinateur
Nesrine CHEHATA Maitre de Conference, Bordeaux INP, France	Directrice de thèse
Clément MALLET Directeur de recherche, IGN-ENSG, LaSTIG, France	Directeur de thèse

Encadrement de la thèse

Loïc LANDRIEU Chargé de recherche, IGN-ENSG, LaSTIG, France	Co-encadrant

©2021 – Vivien Sainte Fare Garnot
all rights reserved.

Thesis advisors: Nesrine Chehata, Clement Mallet
Co-advisor: Loic Landrieu

Vivien Sainte Fare Garnot

Learning Spatio-temporal Representations of Satellite Time Series for Large-scale Crop Mapping.

Abstract

Understanding and monitoring the agricultural activity of a territory requires the production of accurate crop type maps. Such maps identify the boundaries of each agricultural parcel along with the cultivated crop type. This information is valuable for a variety of stakeholders and has applications ranging from food supply prediction to subsidy allocation and environmental monitoring.

While early crop type maps required tedious in situ data collection, the advent of automated analysis of remote sensing data enabled large-scale mapping efforts. In this dissertation, we consider the problem of crop type mapping from multispectral satellite image time series. In most of the literature of the past decade, this problem is typically addressed with traditional machine learning models trained on hand-engineered descriptors. Meanwhile, in Computer Vision (CV) and Natural Language Processing (NLP), the ability to learn representations directly from raw data provoked a paradigm shift leading to unprecedented levels of performance on a variety of problems. Similarly, the application of deep learning models to remote sensing data significantly improved the state-of-the-art for crop type mapping as well as other tasks.

In this thesis, we hold that the direct application of CV and NLP methods to remote sensing tasks tends to ignore crucial particularities of the data at hand. Instead, we argue for the design of bespoke methods leveraging the complex spatial, spectral, and temporal structures of satellite time series. We successively formulate crop type mapping as parcel-based classification, semantic segmentation, and panoptic segmentation, three increasingly difficult tasks. For each of these tasks, we propose a novel deep learning architecture adapted to the task's specificities and inspired by recent advances in the deep learning literature. Our methods set a new state-of-the-art for each task while being more computationally efficient than competing approaches. Specifically, we introduce (i) the Pixel-Set Encoder, an efficient spatial parcel-based encoder, (ii) the Temporal Attention Encoder (TAE), a self-attention temporal encoder, (iii) U-net with TAE, a variation of the TAE for segmentation problems, and (iv) Parcel-as-Point, a lightweight instance segmentation module for the panoptic segmentation of parcels.

We also explore how these architectures can be adapted to multimodal image time series combining optical and radar information through well-chosen fusion schemes. Multimodality improves the mapping performance as well as the robustness to cloud obstruction. Lastly, we focus on the hierarchical tree that encapsulates the semantic relationships between crop classes. We introduce a method to include such structure in the learning process. For crop classification as well as other classification problems, we show that our method reduces the rate of errors between semantically distant classes.

Along with these methods, we introduce PASTIS, the first large-scale open-access dataset of multimodal satellite image time series with panoptic annotations of agricultural parcels. We hope that this dataset, along with the promising results presented in this dissertation, will encourage further research in this direction and help produce ever more accurate agricultural maps.

Resumé

L'analyse et le suivi de l'activité agricole d'un territoire nécessitent la production de cartes agricoles précises. Ces cartes identifient les bordures de chaque parcelle ainsi que le type de culture. Ces informations sont précieuses pour une variété d'acteurs et ont des applications allant de la prévision de la production alimentaire à l'allocation de subventions ou à la gestion environnementale.

Alors que les premières cartes agricoles nécessitaient un travail de terrain fastidieux, l'essors de l'analyse automatisée des données de télédétection a ouvert la voie à des cartographies à grande échelle. Dans cette thèse nous nous intéressons à la cartographie agricole à partir de séries temporelles d'images satellites multispectrales. Dans la plupart des travaux de la dernière décennie ce problème est abordé à l'aide de modèles d'apprentissage automatique entraînés sur des descripteurs conçus par des experts. Cependant, dans la littérature de vision par ordinateur (VO) et du traitement automatique de la langue (TAL), l'entrainement de modèles d'apprentissage *profond* à apprendre des représentations à partir des données brutes a constitué un changement de paradigme menant à des performances sans précédent sur une variété de problèmes. De même, l'application de ces modèles d'apprentissage profond aux données de télédétection a considérablement amélioré l'état de l'art pour la cartographie agricole ainsi que d'autres tâches de télédetection.

Dans cette thèse nous soutenons que les méthodes actuelles issues des littérature VO et TAL ignorent certaines des spécificités des données de télédétection et ne devraient pas être appliquées directement. Au contraire, nous pronons le developpement de méthodes adaptées, exploitant les structures spatiales, spectrales et temporelles spécifiques des séries temporelles d'images satellites. Nous caractérisons la cartographie agricole successivement comme une classification à la parcelle, une segmentation sémantique et une segmentation panoptique. Pour chacune de ces tâches, nous développons une nouvelle architecture d'apprentissage profond adaptée aux particularités de la tâche et inspirée des avancées récentes de l'apprentissage profond. Nous montrons que nos méthodes établissent un nouvel état de l'art tout en étant plus efficaces que les approches concurrentes. Plus précisément, nous présentons (i) le Pixel-Set Encoder, un encodeur spatial efficace, (ii) le Temporal Attention Encoder (TAE), un encodeur temporel utilisant la self-attention, (iii) le U-net avec TAE, une variation du TAE pour les problèmes de segmentation, et (iv) Parcel-as-Point, un module de segmentation d'instance conçu pour la segmentation panoptique des parcelles.

Nous étudions également comment exploiter des séries temporelles multimodales combinant des informations optiques et radar. Nous améliorons ainsi les performances de nos modèles ainsi que leur robustesse aux nuages. Enfin, nous considérons l'arbre hiérarchique qui décrit les relations sémantiques entre les types de culture. Nous présentons une méthode pour inclure cette structure dans le processus d'apprentissage. Sur la classification des cultures ainsi que d'autres problèmes de classification, notre méthode réduit le taux d'erreurs entre les classes sémantiquement éloignées.

En plus de ces méthodes, nous introduisons PASTIS, le premier jeu de données en accès libre de séries temporelles d'images satellites multimodales avec des annotations panoptiques de parcelles agricoles. Nous espérons que ce jeu de données, ainsi que les résultats prometteurs présentés dans cette thèse encourageront d'autres travaux de recherche et aideront à produire des cartes agricoles toujours plus précises.

Contents

0 Introduction 1
- 0.1 A brief history of crop type mapping . . . 3
- 0.2 Problem statement . . . 10
- 0.3 Contributions . . . 14

1 Spatial and Temporal encoding for parcel-based classification 21
- 1.1 Time-space tradeoff for parcel-based classification . . . 23
- 1.2 Pixel-Set Encoder (PSE) . . . 33
- 1.3 Temporal Attention Encoder (TAE) . . . 37
- 1.4 Numerical Experiments: PSE+TAE . . . 42
- 1.5 Lightweight Temporal Attention Encoder (L-TAE) . . . 54
- 1.6 Numerical experiments: PSE+L-TAE . . . 57
- 1.7 Conclusion . . . 65

2 Pixel-based segmentation methods 67
- 2.1 U-Net with Temporal Attention Encoder (U-TAE) . . . 69
- 2.2 Numerical experiments: semantic segmentation . . . 73
- 2.3 Panoptic segmentation: Parcels-as-Points (PaPs) . . . 86
- 2.4 Numerical Experiments: panoptic segmentation . . . 94
- 2.5 Conclusion . . . 100

3 Leveraging multiple modalities 101
- 3.1 Multitemporal fusion . . . 102
- 3.2 Numerical Experiments: Fusion . . . 112
- 3.3 Conclusion . . . 127

4 Leveraging the class hierarchy 130
- 4.1 Metric-guided prototype learning . . . 131
- 4.2 Numerical experiments . . . 140
- 4.3 Conclusion . . . 159

5 Conclusion 160
5.1 Summary 161
5.2 Towards large-scale automated crop mapping. 163
5.3 Epilogue 165

References 167

Acronyms

AHC Average Hierarchical Cost.
AOI Area Of Interest.
BCE Binary Cross Entropy.
CNN Convolutional Neural Network.
CV Computer Vision.
LPIS Land Parcel Identification System.
LSTM Long Short Term Memory.
mIoU mean Intersect over Union.
ML Machine Learning.
NDVI Normalized Difference Vegetation Index.
NLP Natural Language Processing.
OA Overall Accuracy.
PQ Panoptic Quality.
PSE Pixel Set Encoder.
RNN Recurrent Neural Network.
RQ Recognition Quality.
RS Remote Sensing.
SITS Satellite Image Time Series.
SQ Segmentation Quality.
SVM Support Vector Machine.
TAE Temporal Attention Encoder.

To whom it may concern.

Acknowledgments

I would like to thank the team of researchers who supervised this thesis, Nesrine Chehata, Clement Mallet, Sebastien Giordano, and especially Loic Landrieu. Their trust and support, as well as their constructive criticism were all necessary ingredients to accomplish this work.

I am thankful to the LASTIG lab for hosting me during these three years, and for IGN and ASP to have made this PhD possible. I am also thankful to the AUC cafeteria of IGN for providing insane salads, at least until COVID-19 struck.

I am glad to have shared these doctorate years with the other PhD students of LASTIG: Raphael and Damien with whom going climbing was a much appreciated escape, Anatol who was kind enough to give me full usufruct of the office we shared, and all the others with whom I shared fragments of daily life: Luc, Emile, Katerina, Nathan, Romain, Tang, Ewelina, Paul, and all dwellers of the third floor of building K.

Spending three years thinking about how to help a computer distinguish potatoes and carrots from space is no easy business. I could not have made it and remained sane without the presence of my close ones. I thank my family, Iris, Zoï, and Vincent for their support. Special mention to Danijela with whom I shared most closely the good and bad times and whose energy helped me carry on, hvala draga. The next one goes to my friends, those people who live on a similar planet to yours and make life more jovial: Charles, Marco, Theau, Victor, Ronan, Malo, Robin, Raphael, Laetitia, Theo, Vee, Remi, Pauline, Yasmine, Diego, Thomas, Pierre, Dimi, Romain, thank you for being around.

Lastly, I would like to thank *Arkose* for setting up a climbing gym 5 minutes away from the lab, and *Entre midi et 2* for their visionary sandwiches. My experiments showed that these two combined constitute state-of-the-art lunch break.

One of the symptoms of an approaching nervous breakdown is the belief that one's work is terribly important.

Bertrand Russel

0 Introduction

Crop type mapping provides spatially structured information on cultivated crops across all agricultural land of a given territory during a given period. This information is used for a variety of applications: extracting crop production statistics, predicting food supplies, or monitoring crop rotation practices to estimate soil nutrient availability. In some regions, crop type maps are also used for the yearly allocation of agricultural subsidies to farmers. In the European Union or in the United States of America, such subsidies amount to 50 billion euros and 22 billion dollars, respectively. Designing

methods to produce crop type maps at a large scale thus entails major economical and environmental interests. In the following sections, we first show how crop type mapping evolved over the last century. In particular, we show that the challenges shifted from data collection to data analysis. In this perspective, the present manuscript presents novel data analysis methods for automated crop type mapping from satellite imagery.

In this thesis, we explore the potential of modern deep learning methods to automatically analyse large volumes of satellite images to predict crop type maps. Deep learning is a subfield of Machine Learning (ML), and is at the core of great advances during the past decade through impressive applicative achievements on Computer Vision (CV) and Natural Language Processing (NLP) problems [84,171,16,5]. The field of machine learning can be defined in general terms as the study of algorithms that can use experience in the form of observational data, to improve their performance on a given task, *i.e.*, learn [163]. Learning problems can often be formulated as the search for a function that maps a given set of input *features* to a desired outcome [166]. One of the specificity of deep learning methods is the nature of the features this function is applied on. Traditional ML methods typically operate on hand-engineered features and only learn to predict the output based on these features. In contrast, deep learning methods operate directly on the raw observed data and simultaneously learn to extract features and return predictions [50]. This paradigm led to a dramatic improvement in performance on a variety of CV or NLP tasks. In this dissertation, we leverage these advances for remote sensing tasks and data. To this aim, we first outline the precise framing of crop type mapping as a learning problem. In particular, we describe the remote sensing data that we use as input, and the annotations used to train our methods. We also formalize the problem of crop type mapping into three increasingly difficult problems. Lastly, we present an overview of the contributions of the present dissertation.

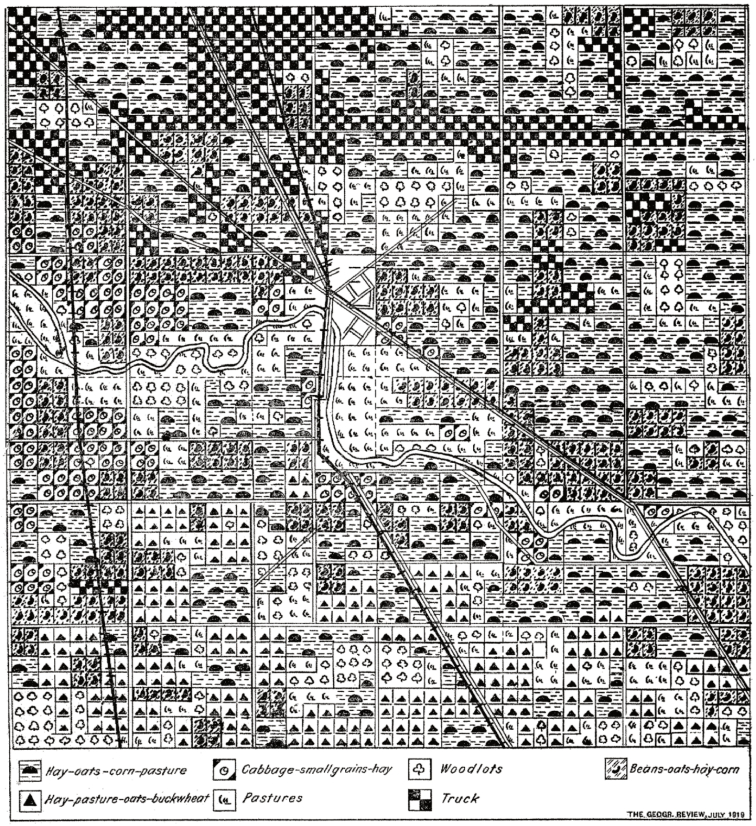


FIG. 1—Map showing, according to the method suggested in the paper, the utilization of the land in Bridgeport Township, Saginaw County, Michigan. Scale, 1:82,000.

Figure 1: Early crop type map. A crop type map of 1919, taken from Sauer [138]. This crop type map covers a square patch of land of a size of approximately 1km, and was produced based on field work. In this map the exact shape of each parcel is not depicted. Instead, the area is subdivided in 20m squares. For each square the Sauer reports the typical rotation of crop cultivated.

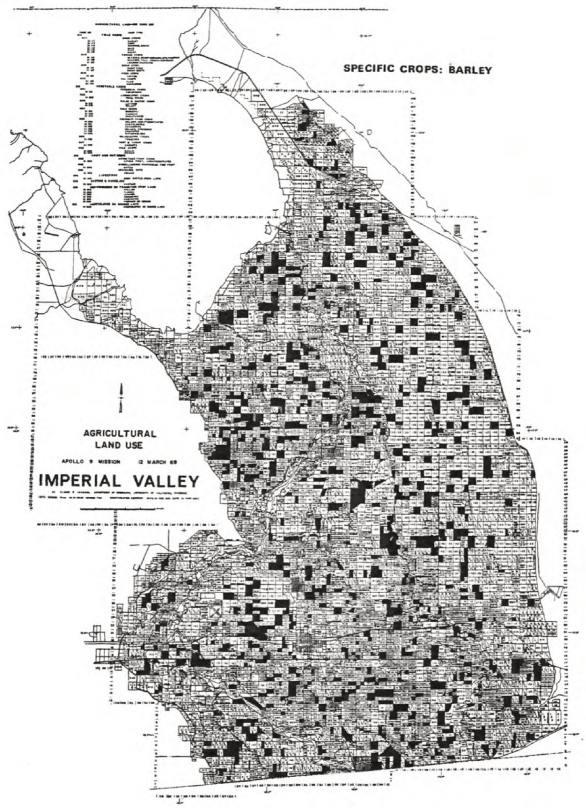
0.1 A brief history of crop type mapping

Early crop type maps. Early implementations of a modern crop type mapping system can be traced back at least to the begining of the 20-th century. We show on Figure 1 a crop type map recorded by Sauer [138] for the Bridgeport Township, Michigan in 1919. Sauer argued that such maps can be a valuable tool for planning both urban and agricultural development. He, indeed, foresaw many of the applications for which they are used nowadays: "[giving] information about the economic condi-

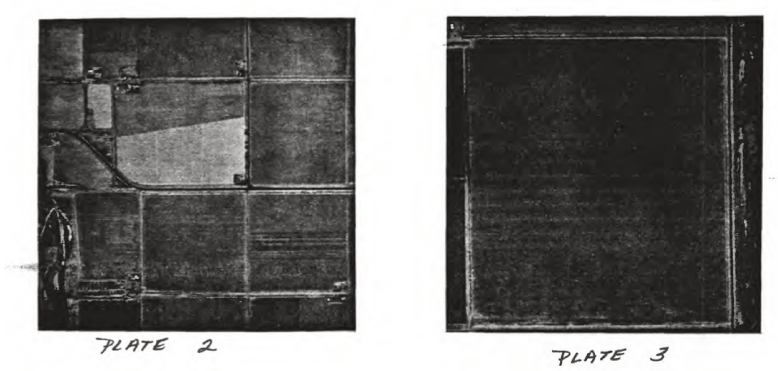
tions", or "[laying] a foundation for a scientific system for tax assessment". Yet, producing such maps required field data collection and tedious work by geographers [138,34], which significantly limited the opportunity of using them at a large scale.

Remote sensing for large scale mapping. The development of Remote Sensing (RS) technologies around the turn of World War II opened the opportunity for efficient crop type mapping efforts at a larger scale in a short time. Airplanes equipped with optical sensors could indeed acquire observations spanning larger areas. A large corpus of work emerged to define photointerpretation techniques to produce forest and crop type maps from such observations [180,6,13]. Photointerpretation consisted in formulating decision rules that expert interpreters could use to assess crop types based on the observed colour, texture, pattern, shape, size, and topographic site [13]. Later, the scope of such mapping efforts widened dramatically with the advent of space-borne remote sensing [70,72,60]. In 1969, one of the first crop type maps based on satellite imagery [70] was produced using photointerpreted images taken by the Apollo 9 mission (see Figure 2) . This landmark study already identified some of the key challenges of crop type mapping from satellite imagery. For example, it discussed the difficulty of having a consistent nomenclature of crop types across different geographical zones due to diet habits and cultural history. Moreover, the Apollo 9 study identified that many of the features used for photointerpretation of aerial images, such as texture, are lost with space-borne sensing. Indeed, the higher altitude of satellite sensors compared to aerial sensors entails a decrease in spatial resolution. The authors concluded that spectral information (*i.e.*, colour) was the only remaining reliable information. As we will see in the first chapter of this dissertation, this observation remains valid and insightful for some present day satellite data.

Evolution of sensors. After the early successes of space-borne remote sensing for crop type mapping, sustained technological efforts have been devoted to improve the quality and availability of re-



(a) Crop type map of Barley in the Imperial Valley.



(b) Sample Apollo 9 images, used to produce the crop type map.

Figure 2: Apollo 9 crop type map. Illustrations of one of the first crop type maps produced using satellite imagery, taken from Johnson *et al.*[70]. The images taken by the Apollo 9 satellite (b), were photointerpreted to predict crop types (a). This map of the Imperial Valley covers a patch of land of approximately 20 by 50km. The thousand-fold increase in the surface covered compared the the map of Sauer (Figure 1) illustrates how remotely sensed observation opened new possibilities for large scale crop type mapping.

mote sensing data. Along the last decades of the 20-th century and up until today, the spatial and spectral resolution as well as the geographical coverage of Earth Observation missions kept on improving. The spectral resolution was first increased with the adoption of multispectral instruments that measure the reflectance of the Earth's surface not only in the visible but also the infrared spectrum. With the launch of the Landsat mission in the 1972 the spatial resolution of multi-spectral images was improved to 30 meters per pixel, proving valuable for a variety of land cover mapping applications. Long-term satellite missions facilitated the production of multitemporal data: successive observations of the Earth's surface could show its evolution over time. The addition of the temporal dimension, as we will see in Chapter 1, was especially useful for crop type mapping as it enabled to observe the temporal dynamics of crop growth. Another line of development has been the diversification of sensors sent to orbit. Notably, in the last couple of decades, satellites equipped with Synthetic Aperture Radar (SAR) sensors were developed, providing complementary information with multispectral observations, as we will see in Chapter 3.

Automated analysis. Along with the development of more sophisticated space-borne sensors, much effort has been devoted to design automated prediction methods and move away from labour-intensive photointerpretation. Early works proposed simple discriminant analysis of Gaussian mixture models [35,166] to classify crop types based on the observed reflectance spectra [79,28], a method refered to *Maximum Likelihood Classification* in the remote sensing literature. Later on, the common approach for crop classification geared to training discriminative ML models such as Random Forest (RF) or Support Vector Machines (SVM) on handcrafted features [174,66,179]. For instance, the Normalised Difference Vegetation Index (NDVI) combining the red and near-infrared spectral bands has been widely used as it relates to crop photosynthetic activity [167]. Certain work also includes phenological features derived from the study of the NDVI as well as external meteorological information [192]. This kind of approaches combining hand-engineered expert features with discriminative model, remained the

state-of-the-art up to recently.

Deep learning. In the past decade, successful advances in the CV and NLP literature[84,171] have provided efficient tools for both spatial and temporal feature extraction. In the context of crop type mapping from Satellite Image Time Series (SITS), the combination of large volumes of open-access data and of publicly available ground truth data (see next section) makes for a natural playground for deep learning approaches. In practice, the remote sensing community followed suit and is gradually adopting deep learning models for automated crop type mapping[87,130,111,63]. Although some work only uses these tools as feature extractors[114], or combine them with feature engineering[190], most current work follows the deep learning paradigm of end-to-end trainable architectures operating on raw data. More specifically, first studies[87,86] proposed to use a Multi Layer Perceptron (MLP) on raw observations instead of traditional RF of SVM. Further work sets out to leverage the spatial and temporal structures of time series of satellite images. Convolutional Neural Nets (CNNs)[88] appeared to be a natural choice to address the spatial dimensions of the data[85,132]. Similarly, Recurrent Neural Networks (RNN)[59] and self-attention[171] networks were successfully adopted from the NLP literature to model the temporal dimension of the data[130,111,131], outperforming RF and SVM[63]. As a results, the work presented here started in a context where the state-of-the-art for crop type mapping tasks was based on deep learning models.

Specificities of Remote Sensing. In this dissertation, we argue that remote sensing tasks such as crop type mapping should not only be seen as special cases of CV or NLP problems. Remote sensing tasks present some key differences in the observed data and phenomena. We hold that this represents a unique opportunity for tailored methods that go beyond applying generic approaches to specific problems.

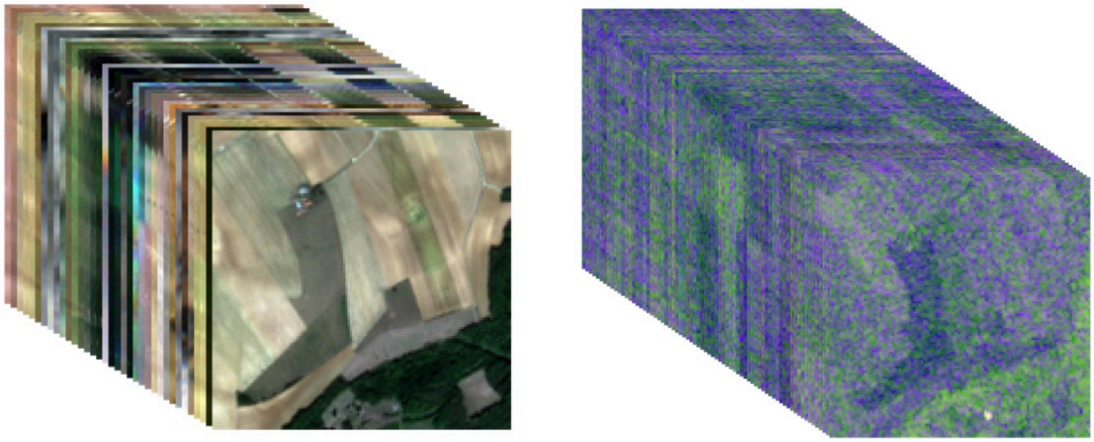
We identify in Table 1 several of the specificities of remote sensing data compared to typical CV

Table 1: Specificities of remote sensing data. Summary of some specificities of remote sensing data as compared to typical data encountered in computer vision: natural images (left) and video (right).

	Sentinel-2 Image	Imagenet Image
Pixel position	Absolute	Relative
Channels	13	3
Sensor	MSI	Camera
Occlusion	Clouds	Between objects

	SITS	Video
Frame of reference	Absolute	Relative
Acquisition time	Crucial information	Arbitrary
Sampling rate	Uneven	Regular
Objects	Fixed	Mobile

data. We consider satellite image time series of Sentinel-2, the main source of satellite data used in this dissertation. Sentinel-2 images are multispectral, providing information beyond the visible part of the spectrum as opposed to the red, green, and blue channels of ImageNet images[84]. In other RS settings images can even be obtained with other types of sensors such as hyperspectral or SAR sensors, see Figure 3b. The objects appearing on Sentinel-2 images do not occlude one another as the line of sight between the satellite and the Earth's surface is clear, except for the occurrence of clouds. Each pixel of a satellite-acquired image is geo-referenced and corresponds to an absolute position on the Earth's surface. This implies that satellite image time series can be constructed with an absolute and fixed frame of reference. In comparison, the sensor position and viewing angle are often moving in video data. Similarly, the observed objects over the Earth's surface do not move in space: the position of an agricultural parcel is fixed, eliminating the need for motion tracking that video analysis models typically need to address. On the contrary, the sensing dates of satellite image time series require more careful processing than the arbitrary acquisition times of videos. A satellite image captured in November does not convey the same insights into the status of a crop's growth cycle as an image captured in March. Furthermore, images in a satellite sequence are often irregularly sampled in time compared to the steady sampling rate of modern cameras. Additionally, in the context of crop type mapping, the relation between the spatial resolution of Sentinel-2 images and the typical dimension of textural information on the parcel's surface is significantly different than in the CV setting. While CNNs have been proven to rely heavily on the texture of objects in natural images[45], the 10m per pixel resolution



(a) Sentinel-2 time series.

(b) Sentinel-1 time series

Figure 3: Sentinel image time series. Optical (left) and SAR (right) image time series of the same patch of agricultural land.

of Sentinel-2 does not allow to resolve most of the texture of agricultural fields such as rows of crops, or ploughing traces, see Figure 3a. Similarly, the relation between the temporal resolution of Sentinel-2, a 5-day revisit time, and the characteristic time of evolution of crops of around a week, is specific to the RS setting. In comparison, video data analysis often addresses the movement of objects with a characteristic time of evolution significantly larger than the sampling period of cameras.

Consequently, we argue that a crucial step in the design of deep learning methods for RS problems is a careful analysis of the specificities of the data and the problem at hand, for the design of an adapted method. Throughout this dissertation, we will try to show how this analysis can lead to the creation of original methods, or adaptations of existing deep learning methods that yield better performance than direct application. We will also see how this can bring significant improvement in terms of speed, precision, and memory usage.

0.2 Problem statement

In this section, we succinctly present the precise framing in which we address crop type mapping. Specifically, we outline the specificities of the data and annotations and show the different ways in which crop type mapping can be formulated.

Figure 4: Multispectral image time series. Sentinel-2 image time series of the same zone across the year 2019 (RGB bands shown only). Note the evolution of the agricultural parcels' appearance. This sample time series also highlights the problem of clouds that can partially (fourth and seventh observation) or totally (third observation) occlude the acquisition.

Satellite image time series. In 2021, the census of the comity on Earth Observation satellites identified more than 250 active satellite missions [2] carried out by private and governmental organisations. This figure is comparable to the total number of Earth Observation satellite missions completed since 1960, and highlights the unprecedented pace at which satellite observations are now produced. In this dissertation, we will focus on multispectral satellite imagery, *i.e.*, with more spectral channels than typical RGB natural images. Near-infrared reflectance is, indeed, a valuable modality for the observation of crops as it allows to monitor the photosynthetic activity of plants [167]. We consider the data produced by the Sentinel-2 mission as the main source of satellite imagery. Sentinel-2 is, as of today, the publicly available source of satellite multispectral imagery providing the best spatial and temporal resolution. Sentinel-2 produces multispectral optical images, with 13 spectral bands ranging from the visible to the short wave infrared range. The spatial resolution of Sentinel-2 varies from 10m to 60m

per pixel depending on the spectral band, and the maximal revisit time is of 5 days. Lastly, Sentinel-2 provides a global coverage of the Earth's land surface. Hence, for any given geographical region, the Sentinel-2 observations during a specific period can be aggregated into a four dimensional multispectral Satellite Image Time Series (SITS): $T \times C \times H \times W$, with a temporal ($T$), spectral ($C$), and spatial ($H \times W$) structure. In this dissertation, we explore learning methods that exploit these dimensions to predict crop type maps. Along the completion of the presented work, we prepared and publicly released two datasets of SITS for crop type mapping. We introduce them in Chapters 1 and 2.

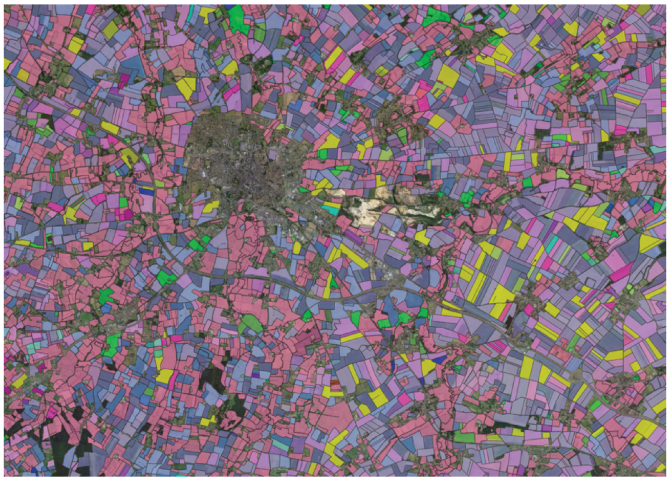


Figure 5: Agricultural maps based on farmers declarations. Portion of the French LPIS for year 2019. The boundaries of each agricultural parcel is depicted by the polygons the color of which depends on the crop type. This dataset can be readily used as annotations for learning-based crop type mapping methods.

Land Parcel Identification System (LPIS). In several countries such as France, crop type maps covering the complete territory are produced on a yearly basis. These crop type maps take the form of a Land Parcel Identification System (LPIS) that is necessary for subsidy allocation. The production of such LPIS relies on the manual declaration by the farmers themselves, who delineate on a geographical map the border of each of their parcels and declare the species of the cultivated crop. In France, this information is publicly released since 2015 amounting to around 10 million annotated parcels every

year. The French Payment Agency estimates the accuracy of crop annotations via in situ control over 98% and the relative error in terms of surfaces under 0.3%. In this dissertation, we use the French LPIS as annotation to train models for predicting parcel crop types and boundaries. In the French LPIS, the crop type reported for a given year n, corresponds to the main cultivated crop between the beginning of fall of year $n-1$ and the end of summer of year n[1]. This temporal frame is denoted as an *agricultural year* in the rest of the manuscript.

Framing the problem. In this context, crop type mapping amounts to retrieving the information contained in the LPIS (extent and crop type of each agricultural parcel) from satellite observations of the corresponding agricultural year. More specifically, as summarised in Table 2 the problem can be framed in the following ways:

- **Parcel-based classification:** In this setting, the borders of each parcel are known and only the crop type needs to be determined by the classification method. Parcel-based classification methods thus focus on discriminating the different types of crops, and take advantage of the information that is available about the extent of the parcels.

- **Pixel-based classification / Semantic segmentation:** In these settings, the borders of the parcels are not known. The classification methods need to make a semantic prediction on the crop type for each pixel of a given area of interest. In pixel-based classification, the prediction for a given pixel is made only using the information of this pixel, whereas in semantic segmentations classification methods have also access to the information of the surrounding pixels of the pixel under consideration. Both approaches allow to quantify the total surface allocated to each crop type on a given territory and retrieve production statistics. However, the borders of each parcel are not predicted, which is limiting in contexts where parcels need to be attributed to their owner, such as for subsidy allocation.

Table 2: Crop type mapping tasks. Summary of the different crop type mapping tasks, in terms of input data shape, a priori knowledge required and prediction level. T: number of observations, C: number of spectral channels, HxW: spatial size of the area of interest, hxw: spatial size of a parcel.

		Input shape	A priori knowledge	Prediction level
Classification	Parcel-based	TxCxhxw	Parcel boundaries	Parcel
	Pixel-based	TxC	-	Pixel
Segmentation	Semantic	TxCxHxW	-	Pixel
	Panoptic	TxCxHxW	-	Parcel & Pixel

- **Instance segmentation / Panoptic segmentation:** The third setting is the most general one, as it aims at retrieving the borders of each parcel as well as their crop type from the SITS. A method that achieves a sufficient performance on this task can thus be used to recover the full LPIS from the satellite observations.

Challenges. The problem of crop type mapping from SITS with deep learning methods presents a variety of challenges. As mapping efforts are nowadays carried out at the scale of an entire country or continent, the employed methods should adapt to variations in the data that can be caused by changes in the meteorological context, farming practices, or growing seasons. The features learnt should also be robust to the occurrence of clouds in the observations. As seen in Figure 4, clouds can obstruct the observations and corrupt the pixel values. Moreover, satellite data providers such as THEIA* do not process satellite acquisitions with a fraction of cloudy pixels exceeding a certain threshold. As a result, the number of available observations changes from year to year and with the region under consideration. Crop type mapping methods should thus be able to process satellite image time series of varying lengths and be robust to their irregular sampling frequency. Crop type mapping also presents the problem of long-tailed distributed classes that can be challenging for learning methods. The number of *Meadow* parcels, the most common crop type in the French territory, is around 250

*https://catalogue.theia-land.fr/

times the count of less common classes such as *Rice*[97]. Those rare classes are often harder to correctly predict as they are less frequent in the training data. For many downstream applications, it is nonetheless crucial that the classification performance is equally high on rare and frequent classes. Lastly, as crop type mapping methods are applied to large volumes of data, computational efficiency needs to be taken into account. In particular, the interest of novel methods should be assessed based on their performance/complexity trade-off.

0.3 Contributions

In the following section, we describe the content of the present manuscript. We assume that the reader is familiar with deep learning. More specifically, we assume that the following concepts are familiar to the reader: training a machine learning model by gradient descent, the deep learning paradigm of end-to-end training, the standard types of deep neural nets (perceptrons, convolutional, and recurrent nets) and training losses. We refer the reader to online available material if necessary[†].

0.3.1 Parcel-based classification

In Chapter 1 we address crop type mapping as a parcel-based classification problem. We assume that the geo-referenced polygons delineating each parcel's border are already known. The methods we investigate thus focus on predicting the semantic label, *i.e.*, the crop type cultivated in the parcel. Following the deep learning paradigm, we aim to design a neural architecture that can directly operate on the raw satellite image time series and learn to extract discriminative spatial, spectral, and temporal descriptors.

In Section 1.1, we present a preliminary study aiming at assessing the relative importance of the spatial and temporal structures of Sentinel-2 for crop type mapping. We show that the multitemporal

[†] https://www.deeplearningbook.org
http://introtodeeplearning.com

nature of Sentinel-2 is key for an accurate crop type classification. We additionally show experimentally, that the convolutional features learnt on Sentinel-2 imagery are only marginally more discriminative than handcrafted features.

In Section 1.2, we leverage these insights to design the Pixel-Set Encoder (PSE). The coarse resolution of Sentinel-2, in the context of parcel classification, motivates us to consider images as unordered sets of pixels that can be encoded with a deep set-based encoder [121] architecture. We show that such an approach outperforms convolutional encoding and favorably circumvents a costly preprocessing step, reducing both computation time and memory usage.

In Section 1.3, we adapt the Transformer architecture [171] to address the temporal dimension of Sentinel-2 time series. Indeed, the Transformer achieved state-of-the-art performance on NLP problems involving sequential data. We analyse the key differences between the typical NLP setting and our parcel-based classification task, and propose subsequent modifications to the original architecture. The resulting TAE temporal encoder, combined with the PSE, sets a new state-of-the-art for parcel-based classification while being faster and more memory efficient than other approaches. We also present in Section 1.5 an improvement to the TAE we developed. The new variant, dubbed Lightweight-TAE (L-TAE), outperforms the TAE while performing an order of magnitude fewer operations.

0.3.2 Segmentation methods

In Chapter 2, we successively broaden the problem statement to semantic segmentation and panoptic segmentation. Indeed, parcel-based methods are not applicable in locations where accurate parcel databases are not available. Such is actually the case for a majority of countries. The aim of this chapter is thus to design methods that are able to retrieve the full LPIS from the input satellite image time series: delineating each parcel's border and predicting its crop type.

We start in Section 2.1 by addressing crop type mapping as a semantic segmentation problem. In

this setting, a semantic prediction is made for each pixel of a given region of interest. This requires a different treatment of the spatial dimensions of the satellite time series than in the parcel-based classification setting. To this aim, we introduce U-TAE (U-Net with TAE), a spatio-temporal encoding architecture for satellite image time series segmentation. We use a typical U-Net like structure of convolutional encoding and decoding. We encode each image of the time series with the shared convolutional encoder and obtain sequences of feature maps. We use the L-TAE to collapse the time series of feature maps into a single feature map. Relying on self-attention allows us to reuse the attention masks produced at a certain depth at other levels of the U-Net structure. We show experimentally that this feature gives the U-TAE an important edge over other existing methods, and thus sets a new state-of-the art for semantic segmentation of satellite image time series for crop type mapping.

In Section 2.3, to also retrieve the parcels' borders, we frame crop type mapping as panoptic segmentation, which corresponds to assigning each pixel of an image a *single* instance id and semantic label. Panoptic predictions are by design *non-overlapping* instance masks with associated class predictions. This is appropriate for crop type mapping as agricultural parcels do not overlap. Inspired by the recent computer vision literature on single-stage instance segmentation algorithms, we devise Parcel-as-Points (PaPs), an instance segmentation module that we combine with the U-TAE to perform panoptic segmentation of satellite image time series. Instance segmentation of satellite images has not, to the best of our knowledge, been explored on multitemporal data, and we thus set the first state-of-the-art on panoptic segmentation from SITS.

0.3.3 Leveraging multiple modalities

In Chapter 3, we explore the opportunity of leveraging multiple modalities to improve the performance of crop type mapping models. Specifically, we focus on the joint use of the optical imagery of Sentinel-2 with the C-band radar acquisitions of Sentinel-1. The latter produces open access observations containing complementary information to the multispectral measurements of Sentinel-2.

While the spectral channels of Sentinel-2 convey information on the physiological activity of crops, the radar measurements of Sentinel-1 capture information on the surface geometry of agricultural parcels. Moreover, SAR observations are not sensitive to cloud obstruction and can thus complement Sentinel-2 data during cloudy periods.

The joint use of optical and radar acquisitions for crop type mapping has been extensively explored by the remote sensing community [170,102,20]. However, few works use modern deep learning architectures yet [64]. We thus explore how to combine SITS from multiple modalities with the temporal attention models introduced in Chapters 1 and 2. We implement different feature fusion schemes commonly encountered in the literature and evaluate the schemes on parcel-based classification, semantic segmentation, and panoptic segmentation. We show that the addition of the radar modality improves the overall performance on these tasks, as well as the robustness to cloud obstruction.

0.3.4 Leveraging the class hierarchy

In the context of subsidy allocation, it can be valuable to reduce misclassifications between semantically distant classes (*e.g.*, wheat and apple trees), as they also tend to have different subsidy levels. This motivates us to focus in Section 4 on the hierarchical structure of the different crop types, which can be efficiently captured by a tree structure designed by experts. This hierarchical tree induces a distance between the different classes, and this distance can be used to measure the severity of classification errors. We set out to develop a method to leverage this hierarchical knowledge to reduce the severity of errors. This endeavor is an active field of research in the ML and CV communities. Hence, in this section we widen our scope from crop type mapping to generic classification problems with a known hierarchical tree on the class set. We introduce a method based on prototype learning [147], allowing us to incorporate the hierarchical structure between classes into the arrangement of their respective prototypes in the embedding space. We show experimentally that our method consistently reduces the severity of errors. Furthermore, our experiments demonstrate that our method also reduces the

overall number of classification errors. This suggests that the hierarchical class tree of classes provides valuable information on the structure of the data, and that classification models' performance can be improved with the addition of a simple regularizer and no additional architectural or dataset changes. In particular, the classification performance of crop type mapping models can be significantly improved using this hierarchical knowledge.

0.3.5 Publications

Most of the work presented in the following manuscript was published in international journals and conferences during the completion of the doctorate.

International Journal

- Garnot, V.S.F., Landrieu, L. and Chehata, N.,"Multi-Modal Temporal Attention Models for Crop Mapping from Satellite Time Series", ISPRS journal, 2021 *Under Review*

International Conferences

- Garnot, V.S.F. and Landrieu, L., "Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks", ICCV 2021
- Garnot, V.S.F. and Landrieu, L.,"Leveraging Class Hierarchies with Metric-Guided Prototype Learning", BMVC 2021
- Garnot, V.S.F., Landrieu, L., Giordano, S. and Chehata, N.,"Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-attention" CVPR 2020
- Garnot, V.S.F., Landrieu, L., Giordano, S. and Chehata, N., "Time-Space Tradeoff in Deep Learning Models for Crop Classification on Satellite Multi-Spectral Image Time Series", IGARSS 2019

International Workshop

- Garnot, V.S.F. and Landrieu, L., "Lightweight Temporal Self-attention for Classifying Satellite Images Time Series", International Workshop on Advanced Analytics and Learning on Temporal Data, ECML/KDD 2020.

0.3.6 Other Contributions

Datasets. Along with the published articles, we released two benchmark datasets to encourage comparable research in crop type mapping. Our ready-to-use benchmarks can hopefully benefit practitioners unfamiliar with RS data.

- *S2-Agri* was released with our earliest work on parcel-based classification. It contains 200k agricultural parcels in a single region of France. For each parcel, we prepared a time series of 24 Sentinel-2 observations during the 2017 agricultural year. This dataset was downloaded around 50 times since its publication.

- PASTIS (Panoptic Satellite image TIme Series) was released in 2021. While our first dataset could only be used to evaluate parcel-based classification approaches, PASTIS can be used as a benchmark for object-based classification, semantic segmentation, and panoptic segmentation. The dataset is composed of 2433 image time series of 128×128 resolution and with 10 spectral bands. For each patch, we gather all available acquisitions between September 2018 and November 2019 amounting to 115k Sentinel-2 images. The patches were selected in four different French regions and cover and area of 4000km^2, and contain around 120k agricultural parcels. We also released PASTIS-R, which extends PASTIS with the corresponding Sentinel-1 radar acquisitions for each patch for a total of 339 174 radar images. At the time of writing, PASTIS was already downloaded more than 260 times.

Code. In an commitment to reproducible research, we made all our research code publicly available in the following repositories:

- `github.com/VSainteuf/pytorch-psetae`
- `github.com/VSainteuf/lightweight-temporal-attention-pytorch`
- `github.com/VSainteuf/utae-paps`
- `github.com/VSainteuf/pastis-benchmark`
- `github.com/VSainteuf/metric-guided-prototypes-pytorch`

Time and space - time to be alone, space to move about -

these may well become great scarcities of tomorrow.

Edwin Way Teale

1

Spatial and Temporal encoding for parcel-based classification

In this chapter, we consider the problem of crop classification on optical multispectral time series, when the parcel segmentation is already known, *i.e.*, parcel-based classification. In this setting, only the pixels contained in the parcel boundaries are considered as shown on Figure 1.1, and the parcel is classified based on the sequence of satellite observations. Leveraging recent advances in the deep

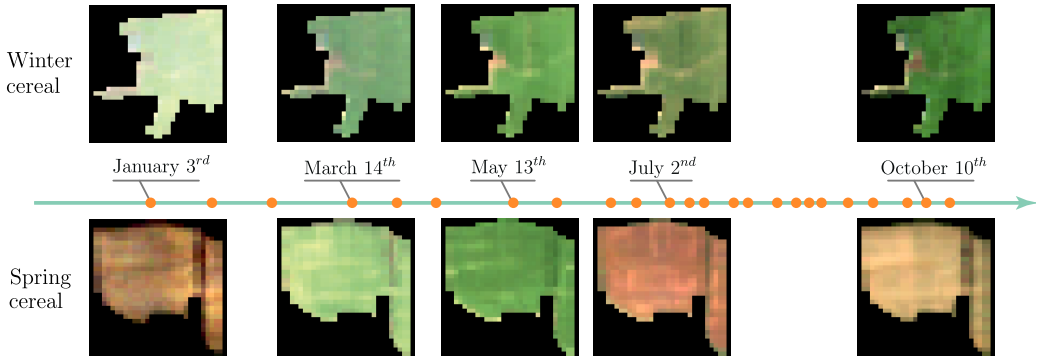


Figure 1.1: Input data. Example of Sentinel-2 time series (shown: RGB bands, 10m per pixel) for two parcels of the *Winter cereal* and *Spring cereal* classes. The dots on the horizontal axis represent the unevenly distributed acquisition dates over the period of interest.

learning literature, and accounting for the specificities of the problem at hand, we design tailored spatial and temporal encoders that outperform existing approaches both in classification performance and computational efficiency.

In the first section, we carry out a preliminary study to explore the relative importance of the temporal and spatial structures of Satellite Image Time Series (SITS) for parcel-based crop type classification. We then present our two encoding modules: the Pixel-Set Encoder (PSE) and the Temporal Attention Encoder (TAE) for spatial and temporal encoding, respectively. Lastly, we present the L-TAE, a variant of our TAE temporal encoder with improved memory and computational efficiency. We conduct extensive numerical experiments on a large dataset of 200k agricultural parcels. On this dataset, our combined spatial and temporal encoders improve the state-of-the-art for parcel-based classification by 9.6pts of mean Intersect over Union (mIoU).

1.1 Time-space tradeoff for parcel-based classification

In this section, we propose a preliminary study to help in the design of deep learning architectures for parcel-based crop type classification from SITS. In such architectures, one key design choice is the size of the different parts of the model dedicated to the different dimensions structuring the input data. In the case of SITS, we aim at determining empirically the relative size of the spatial encoding and temporal encoding modules to achieve the best classification performance. Hence, we propose to answer the following question: given a fixed budget of trainable parameters, should one prioritize modeling the spatial structure (with CNN), temporal structure (with RNN), or address both with recurrent convolutional models? We compare the crop classification performance of several architectures with the same number of parameters on a Sentinel-2 dataset of agricultural parcels.

The key highlights of this experiment are as follows:

- We provide empirical evidence that the temporal structure of Sentinel-2 data is more discriminative than the spatial structure in the context of parcel-based crop type classification. Consequently, most of the model's trainable parameters should be devoted to temporal encoding.
- We show that recurrent architectures are acting as a memory combining multiple observations, as well as a model for temporal evolution.

1.1.1 Methods

1.1.1.1 Neural Network Architectures

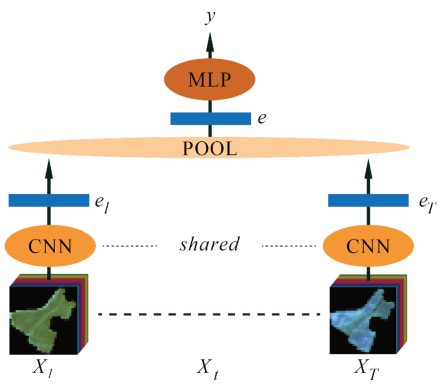
In order to assess the influence of the temporal and spatial structure, we implement four neural network architectures. All of them follow the typical deep learning paradigm: first learn to extract an embedding — spatial, temporal or both — from the input image sequence and then to classify the sample based on this embedding. We show an illustration of each architecture on Figure 1.2. The

same design of classification module is used across all architectures: a Multi Layer Perceptron (MLP) with two hidden layers of dimension 128 and 64.

- **CNN :** We first implement a convolution-based neural network whose goal is to leverage the spatial structure of image time series. The images corresponding to each date are embedded independently through the same three layers composed of the following units: convolution with 3×3 kernel size and no padding, batch normalisation[67], ReLu activation[49] and MaxPool with 2×2 kernel size. We then compute a global embedding for the whole sequence by concatenating all the image embeddings and taking the maximum for each channel, in the manner of PointNet[121]. Finally, the global embedding is passed to the classification module for prediction.

- **RNN :** Unlike the CNN network, the RNN architecture focuses purely on the temporal dimension of image sequences. For each image, we compute a vector of parcel-level handcrafted features. Following the common approach of using statistical descriptors[87], we compute the spatial mean and standard deviation of each spectral band. These vectors are then processed in chronological order by a recurrent net. We choose a Gated Recurrent Unit (GRU)[26] over Long Short Term Memory (LSTM)[59] for its better parameter efficiency. The last hidden state of the GRU is used as the embedding for classification with the classification module.

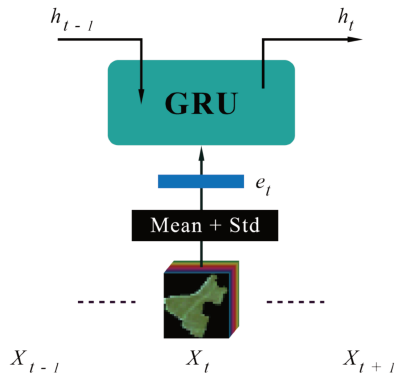
- **CNN+GRU :** Our first hybrid implementation successively extracts spatial and temporal embeddings. Each image of the sequence is first embedded with a shared CNN network. The resulting sequence of spatial embeddings is then processed by a recurrent GRU in chronological order. The last hidden state of the GRU is used as a spatio-temporal embedding for classification. We implement three such models with varying ratios of parameters allocated to the temporal structure. Indeed, we can reduce the number of convolutional kernels and chose a larger hidden state size to increase this ratio (see Table 1.1).



CNN

$$\text{pool}_T(\{\text{CNN}(x_t)\}_t)$$

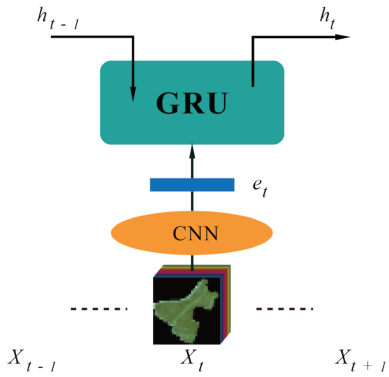
(a) CNN architecture



GRU

$$\text{RNN}([\text{pool}_{H,W}(x_t)]_t)$$

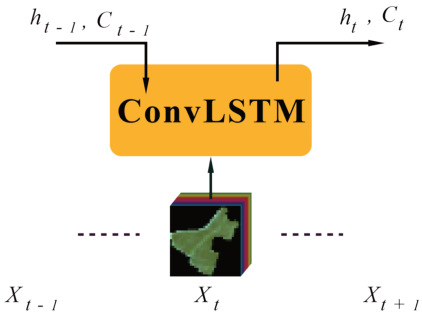
(b) RNN architecture



CNN+GRU

$$\text{RNN}([\text{CNN}(x_t)]_t)$$

(c) CNN+GRU architecture



ConvLSTM

$$\text{ConvLSTM}([x_t]_t)$$

(d) ConvLSTM architecture

Figure 1.2: Neural architectures. Illustration of the four different architectures used in the experiment. Each architecture addresses the spatial and temporal dimension of the input sequence in a specific way.

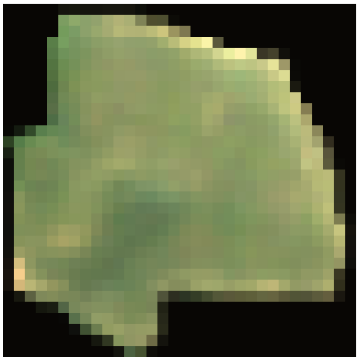
- **ConvLSTM :** Our second hybrid implementation follows the ConvLSTM architecture introduced by Xingjian *et al.* [183]. which directly performs spatial encoding within the recurrent cell. ConvLSTM uses image-shaped hidden and cell states, as well as convolutions instead of MLP layers in an LSTM architecture [59]. We refer the reader to Rußwurm & Körner [132] for more details on this architecture.

We arbitrarily set the budget of trainable parameters to 100k, which proved sufficient for the models to generalize in our experiments. Table 1.1 summarizes the hyper-parameters used for the models: the number of kernels for each of the convolutions and the size of the hidden state of the recurrent unit. We do not consider three-dimensional convolutional architectures to cover the spatial and temporal dimension of the data, contrarily to previous studies such as Ji *et al.* [69]. Indeed, convolutions are local computations, and hence are not as well suited for modeling long term dependencies as recurrent architectures. Additionally, applying convolutions along the temporal axis assumes that the images of the sequence are regularly sampled in time, which is not necessarily the case in practice.

Table 1.1: Hyperparameters. Summary of the models' hyperparameters. When applicable, we show the number of kernels in the successive layers of the CNNs, the sizes of the hidden state of the GRU cells, the ratio of trainable parameters allocated to the temporal dimension of the data, as well as the total number of trainable parameters of each model.

Model	Number of Kernels	Hidden State Size	Temporal Parameter Ratio	Total Number of Parameters
CNN	16 : 32 : 96	-	0	92 899
CNN+GRU$_7$	16 : 32 : 64	64	0.7	92 035
CNN+GRU$_8$	16 : 32 : 36	96	0.8	93 807
CNN+GRU$_9$	16 : 16 : 16	128	0.9	90 179
GRU	-	156	1	94 587
ConvLSTM	30 : 64	64	-	95 353

(a) Fragment of tile T31TFM of Sentinel-2.



(b) Example of an image patch for one observation of a parcel.

Figure 1.3: Parcel-based data. Example of input image and one observation of a dataset sample.

1.1.1.2 Implementation details

All models are trained for 50 epochs with Adam optimizer[76] set with a batch size of 32 and a learning rate of 10^{-3}. We use 5-fold cross-validation: for each fold, the dataset is split into train, validation and test set with a 3:1:1 ratio. The epoch achieving the best results on the validation set is used for performance evaluation on the test set. The test metrics we report are computed on the total confusion matrix, equal to the sum of the confusion matrices of each fold's test predictions.

1.1.2 Numerical Experiments

1.1.2.1 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 bilinearly 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 central 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 during the 2017 agricultural year. The values of cloudy pixels are linearly interpolated from the first previous and next available pixels using Orfeo Toolbox[25]. We retrieve the geo-referenced polygon and class label of each parcel from the French Land Parcel Identification System (LPIS).* We crop the satellite images using this polygon to constitute the image time series. We set to zero all pixels outside the parcel (see Figure 1.3(b)). The patches are resized to 32×32 pixels using linear interpolation. This implies upsampling for a majority of parcels.

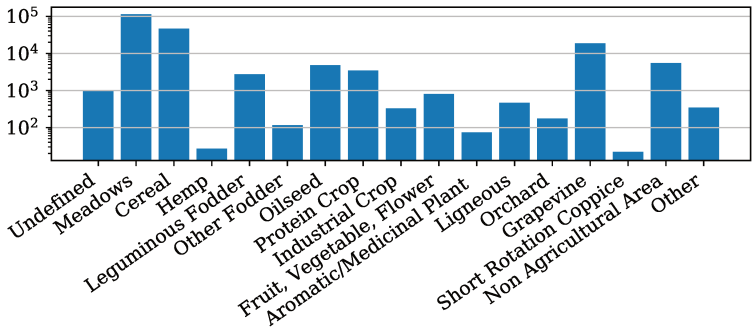


Figure 1.4: Class breakdown. Class distribution in our dataset.

The dataset is thus comprised of 199 464 tensors of size $24 \times 10 \times 32 \times 32$. The images are normalised channelwise for each date individually. We use the French LPIS to associate a label to each parcel. The labels are drawn from a comprehensive terminology that covers all observable parcel types and regroups them in 18 high-level classes. Figure 1.4 shows the breakdown of the different classes in the AOI, where meadows, cereals, and grapevine are dominant and only the *rice* class is not represented.

*http://professionnels.ign.fr/rpg

1.1.2.2 Analysis

We present the performance of each model in Table 1.2. Given the high imbalance of the dataset under consideration (see Figure 1.4), we report the unweighted class-wise average F-score along with the Overall Accuracy (OA) of each model. We insist on the fact that all models are designed with approximately the same number of trainable parameters (see Table 1.1). Thus, the differences in performance can only be attributed to the way the spatial and temporal dimensions are handled and not to differences in model size.

Table 1.2: Classification experiment. Performance metrics of the different models. We report the Overall Accuracy (OA), and class-averaged F-score, precision, and recall.

Model	OA	F-score	Precision	Recall
CNN	89.5	34.9	53.0	31.4
CNN+GRU$_7$	93.4	53.2	59.7	49.9
CNN+GRU$_8$	93.7	55.1	63.3	51.6
CNN+GRU$_9$	**93.7**	**55.1**	**65.1**	**51.8**
GRU	93.5	53.8	63.0	50.0
ConvLSTM	93.1	49.2	64.0	44.5

Surprisingly, the purely recurrent GRU model outperforms the ConvLSTM and the CNN+GRU$_7$ hybrid models. Only the CNN+GRU$_8$ and CNN+GRU$_9$ models achieve higher overall performance. This shows that extracting both spatial and temporal structures allows for a higher classification performance, provided that most of the parameters are allocated to the temporal structure. Only using 10% of the parameters budget for the spatial feature extractor seems sufficient for spatial feature extraction with convolutions.

This suggests that the features extracted by RNNs are more discriminative than those extracted by CNNs. Indeed, the purely convolutional model performs significantly worse than its purely recurrent counterpart (by 19 pts of F-score).

This performance gap can be explained by the fact that convolutional features are not completely

Table 1.3: Per-class performance. F-score on the test set reported per class.

	CNN	GRU	CNN+ GRU_9	Conv LSTM
Undefined	0.0	**45.1**	35.4	38.2
Meadows	92.9	95.8	**96.0**	96.0
Cereal	93.3	97.1	**97.5**	97.5
Hemp	21.6	60.5	64.4	**72.7**
Leguminous Fodder	15.6	42.9	**43.8**	34.8
Other Fodder	0.0	25.5	**32.9**	14.8
Oilseed	93.1	95.9	**96.0**	95.1
Protein Crop	79.2	87.9	**89.1**	87.6
Industrial Crop	19.6	40.3	**47.3**	23.9
Fruit, Vegetable, Flower	42.5	60.3	**63.1**	55.1
Aromatic/Medicinal Plant	0.0	**32.8**	28.9	8.3
Ligneous	5.5	**39.8**	38.6	32.1
Orchard	0.0	**8.0**	6.0	2.5
Grapevine	81.8	**95.1**	94.3	92.4
Short Rotation Coppice	0.0	8.3	**18.2**	17.4
Non Agricultural Area	46.7	54.9	**59.1**	55.2
Other	1.2	24.0	**27.0**	14.1

relevant for the problem at hand. CNNs are well suited for extracting shape and texture information, and it appears that parcels' shape does not strongly correlate with crop type. Furthermore, the resolution of Sentinel-2 images may not allow to capture rich texture information (see Figure 1.3(b)). This would also explain why the ConvLSTM model — which relies on convolutions for spatial encoding — performs slightly worse that the CNN+GRU ones.

Additionally, these results highlight the importance of choosing a large hidden state size when using RNNs to fully leverage the temporal structure of the data. Comparing the GRU and CNN+GRU_7 models indicates that allocating too many parameters to extract convolutional features reduces the performance compared to a model operating on simple handcrafted features with a larger hidden state.

We show the per-class performances of the four types of architectures in Table 1.3. For 11 out of

17 classes, the use of convolutional features in the CNN+GRU$_9$ improves the performance compared to the GRU model. Yet some classes such as *Grapevine* or *Ligneous* present the opposite behaviour, showing that the relevance of recurrent or convolutional feature extractors is class-dependent.

Table 1.4: Time shuffling experiment. F-score while trained on the regular image sequences, and with randomly shuffled sequences.

	Ordered	Shuffled	Δ
ConvLSTM	49.2	47.9	-1.3
CNN+GRU$_7$	53.2	**52.0**	-1.2
CNN+GRU$_9$	**55.1**	48.8	-6.3
GRU	53.8	45.3	**-8.5**

Finally, to assess the importance of the temporal structure for the features extracted by the recurrent networks, we retrain several models with randomly shuffled input sequences, such that the temporal structure is lost. Table 1.4 summarizes the F-scores obtained.

Time-shuffling of the image sequence is detrimental to all models. This impact is all the more important as the ratio of temporal parameters is high. Yet, all models still outperform the purely convolutional model. These results suggest that the hidden states of the recurrent units act in two ways: first, as a memory storing information regardless of their order, and second, as a model for the temporal evolution of the crops. As our dataset only covers a single year, this chronological evolution is most probably capturing the phenology of the crops.

1.1.3 Concluding Remarks

In this section, we compared the performance of four deep learning architectures extracting spatial, temporal, or spatio-temporal features for crop type mapping from SITS with a fixed budget of parameters. Our results showed that architectures with 90% of their parameters are allocated to the extraction of temporal patterns achieve the best classification performance.

This suggests that simple convolutional architectures are sufficient to extract expressive features from Sentinel-2 images. Moreover, this emphasizes the importance of the temporal dimension of Sentinel-2 data for crop type classification. We showed that RNNs can successfully leverage this structure by acting as a memory combining multiple observations and foremost by taking into account the temporal evolution of the different observations over a year.

More generally, our results highlight the potential of deep learning models for agricultural parcel classification: all RNN and RNN+CNN models outperform the RF baseline, which achieves an average F-score of 36.9 on the same dataset (not shown here).

1.2 Pixel-Set Encoder (PSE)

In this section, we present a spatial encoder architecture designed according to the previous conclusions and inspired by recent advances in the deep learning literature.

1.2.1 Motivation

Sentinel-2 has a spatial resolution of 10m per pixel. Such a level is considered high resolution in the remote sensing literature as other satellite sensors can have kilometer-sized pixels. Yet, in the context of crop type mapping, the resolution of Sentinel-2 is coarser than typical agricultural textural information such as furrows or crop rows. However, CNNs rely heavily on texture to extract spatial features [45]. As a matter of fact, the results of the previous section showed that convolutional features outperform handcrafted descriptors by only a slight margin. Given this limitation, we propose to view coarse 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 [121,189].

We show in this section that set-based encoders can successfully extract learned statistics of the distribution of spectral observations across the spatial extent of the parcels. Furthermore, we show that this approach handles the highly variable size of parcels more efficiently than CNNs.

1.2.2 Methods

We denote the observations of a given parcel by a spatio-spectro-temporal tensor $[x^{(0)}, \cdots, x^{(T)}]_{t=1}^{T}$ 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 1.1.

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 Sentinel-2 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 (see Figure 1.5) makes this process very memory intensive. Indeed, to limit information loss for large parcels, this amounts to oversampling most smaller parcels several times over.

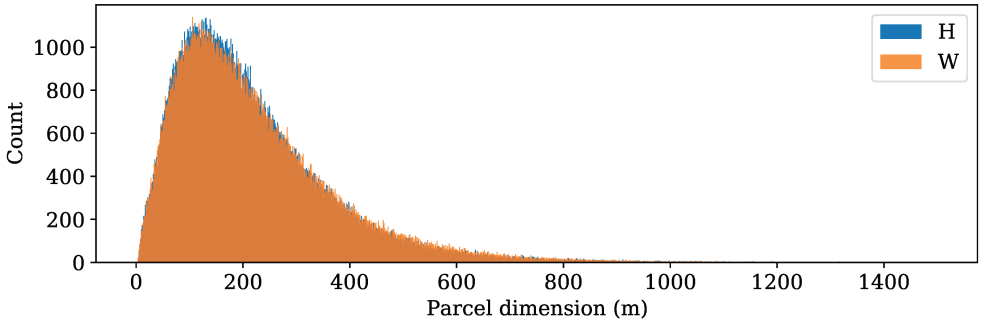


Figure 1.5: Distribution of parcel sizes. We plot the distribution of agricultural parcel's height (H) and width (W) in the dataset presented in Section 1.4. Note the high variability of these dimensions: both height and width range from a few meters to a thousand meters and have a relative standard deviation of ~ 0.7.

To circumvent both issues, we propose an alternative architecture called *Pixel-Set Encoder* (PSE) and inspired by the point-set encoder PointNet[121] and the Deep-Set architecture[189] 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.

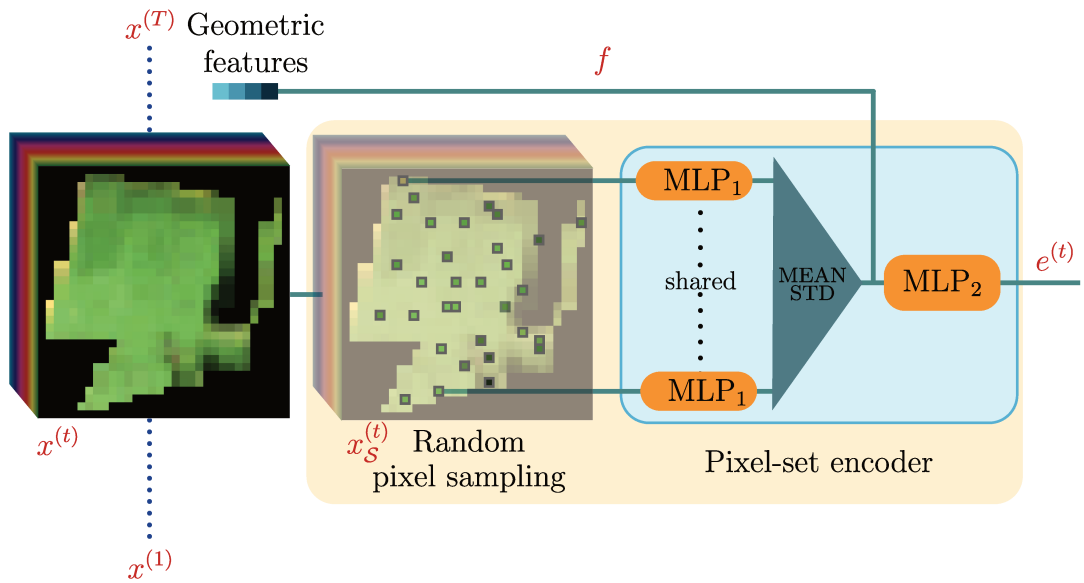


Figure 1.6: Pixel-Set Encoder. First, a fixed number of pixels is randomly drawn from the input image $x^{(t)}$. These pixels, $x_s^{(t)}$, are encoded with the shared MLP$_1$. The resulting set of vectors is pooled into a single vector that is in turn encoded by MLP$_2$, which outputs the final embedding $e^{(t)}$ of the input image. We also concatenate some geometrical descriptors f of the parcel's shape before MLP$_2$, as it can be a relevant information for crop type classification.

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

i) A set $\mathcal{S} \subset [1, \cdots, N]$ of S pixels is randomly drawn from the N pixels within the parcel, as described in Equation 1.1. When the total number of pixels in the image is smaller than S, an arbitrary pixel is repeated to match this fixed size. The same set $\mathcal{S}$ is used for sampling all T acquisitions of a given parcel.

ii) Each sampled pixel s is processed by a shared multi-layer perceptron MLP$_1$, as seen in Equation 1.2, composed of a succession of fully connected layers, Batch Normalisation [67], and Rectified Linear Units [109].

iii) The resulting set of values is pooled along the pixel axis—of dimension S—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 vector precomputed geometric features f: perimeter, pixel count N, cover ratio (N divided by the number of pixels in the bounding box) and the ratio between the perimeter and surface of the parcel.

iv) This vector is processed by another perceptron MLP_2, as shown in Equation 1.3, to yield $e^{(t)}$ the parcel's spatio-spectral embedding at time t.

The PSE architecture is represented in Figure 1.6, and can be summarised by the following equations:

$$\mathcal{S} = \text{sample}\left(S, N\right) \tag{1.1}$$

$$\hat{e}_s^{(t)} = \text{MLP}_1\left(x_s^{(t)}\right) \ , \ \forall s \in \mathcal{S} \tag{1.2}$$

$$e^{(t)} = \text{MLP}_2\left(\left[\text{pooling}\left(\{\hat{e}_s^{(t)}\}_{s\in\mathcal{S}}\right), f\right]\right) \ . \tag{1.3}$$

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 available information. We provide extensive numerical experiments to assess the efficiency of the PSE in later Section 1.4.

1.3 Temporal Attention Encoder (TAE)

At the time of carrying out this work, hybrid neural architectures combining convolutions and recurrent units in a single architecture constituted the state-of-the-art for crop type classification [131,43]. In the previous section, we argued for shifting away from convolutions for spatial encoding. In this section, we advocate for using self-attention-based methods for temporal encoding.

1.3.1 Motivation

The results we presented in Section 1.1 established the significance of the temporal dimension for crop type classifyication [43]. While RNNs have been widely used to analyse temporal sequences, recent work in NLP has introduced a promising new approach based on self-attention mechanisms [171]. The improved parallelism brought by this approach is particularly valuable for automated crop monitoring, as its typical spatial scale spans entire continents: one year of Sentinel-2 observations amounts to 25TB of data for agricultural areas in the European Union. Therefore, we propose to adapt attention-based approach for the classification of time series.

1.3.2 Methods

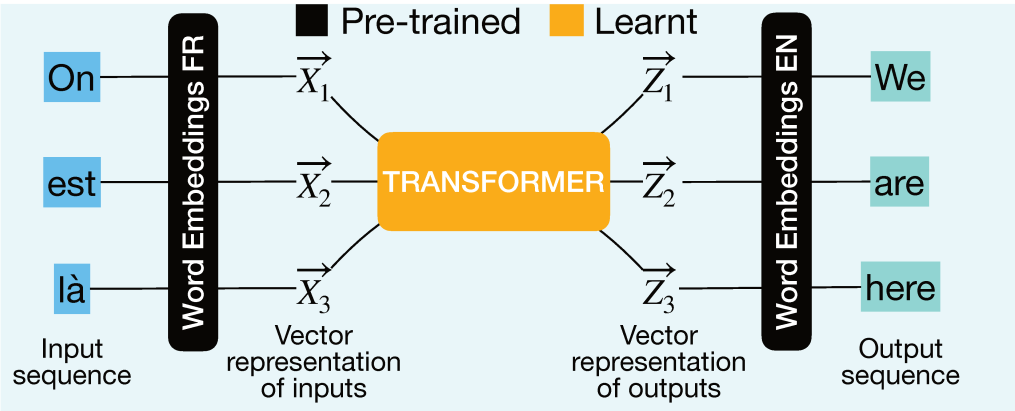
RNNs have proven efficient for encoding sequential information [93]. However, since RNNs process the elements of the sequence successively, they prevent parallelisation and incur long training times. Vaswani *et al.* [171] introduce the Transformer architecture, an attention-based network achieving equal 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 Transformer model, a *query-key-value* triplet $\left(q^{(t)}, k^{(t)}, v^{(t)}\right)$ is simultaneously computed 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.

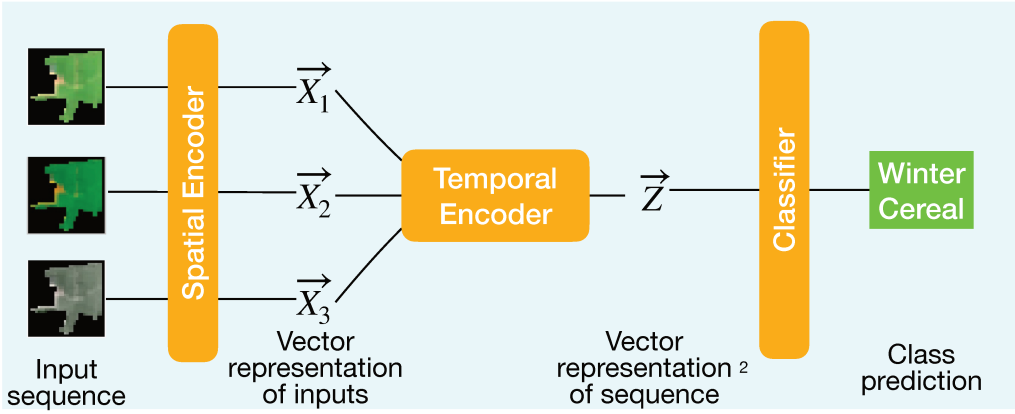
Since the computation of the triplets $\left(q^{(t)}, k^{(t)}, v^{(t)}\right)$ and their multiplications can be performed in parallel, the Transformer takes full advantage of modern GPU architecture and boasts a significant speed increase compared to recurrent architectures. This procedure can be computed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

Positional Encoding. In their paper on text translation, Vaswani et al. [171] add order information to elements of the input sequence by adding a positional encoding tensor to each element. Equation 1.4 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 $\tau = 1\,000$—instead of $10\,000$. Additionally, instead of encoding the position in the sequence, we encode the date observation $\mathrm{day}(t)$, expressed in number of days since the beginning of the agricultural year. This helps to account for inconsistent temporal sampling (see Figure 1.1).

$$[p^{(t)}]_{i=1}^{d_e} = \sin\left(\mathrm{day}(t) \backslash \tau^{\frac{2i}{d_e}} + \frac{\pi}{2}\mathrm{mod}(i, 2)\right) \tag{1.4}$$



(a) Schematic pipeline for a typical NLP task of translation from French (FR) to Engligh (EN). In such settings the Transformer transforms a sequence into another sequence, and operates on pre-trained word embeddings.



(b) Schematic pipeline of our image time series classification setting. The input sequence is mapped to a single embedding and the spatial encoder is trained end-to-end with the temporal encoder.

Figure 1.7: Crop mapping and NLP. Schematic view of the key differences between a typical NLP task (top) and our setting (bottom).

End-to-End Encoding. The original Transformer network takes pretrained word embeddings as inputs, as depicted in Figure 1.7a. In our setting however, the parameters of the network producing the inputs are learnt simultaneously with the attention parameters. Therefore, we propose that each head only computes key-query pairs from the spatial embeddings (1.5) since these embeddings can directly serve as values: $v^{(t)} = e^{(t)} + p^{(t)}$. 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 the entire time series into a single embedding. Consequently, we only retain the *encoder* part of the Transformer and define a single *master query* $\hat{q}_h$ 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 (1.6). As shown in Equation 1.7, 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 (1.8).

Multi-Head Self-Attention. We concatenate the output o_h of each head h for the n_h different heads and process the resulting tensor with MLP_3, to obtain the final output $\hat{o}$ of the Temporal Attention Encoder (TAE), as shown in Equation 1.9. Note that unlike the Transformer network, we directly use $\hat{o}$ as the spatio-temporal embedding instead of using residual connections. This is a direct consequence of the TAE returning a single embedding, as opposed to the Transformer, which returns a sequence of embeddings (see Figure 1.7).

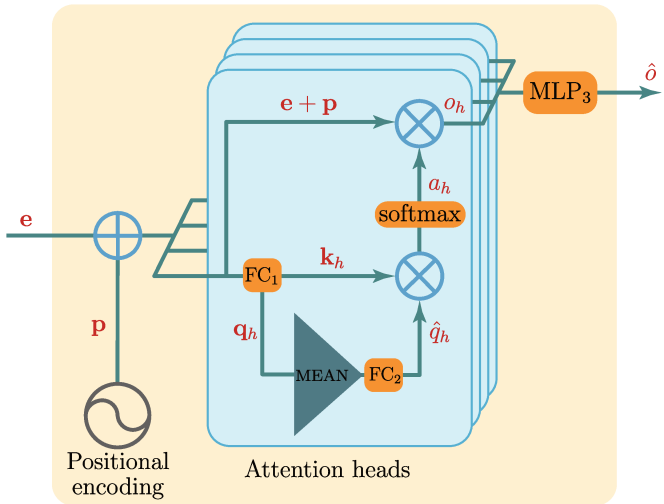


Figure 1.8: Temporal Attention Encoder. Variables in bold are tensors concatenated along the temporal dimension, *e.g.* $\mathbf{e} = [e^{(0)}, \cdots, e^{(T)}]$. The input sequence is first complemented with additive positional encoding $\mathbf{p}$. Then several self-attention heads encode the sequence in parallel. With our *master query* scheme ($\hat{q}_h$) only one output vector is computed by each head. The output of all heads o_h are concatenated into a vector that is eventually processed by MLP_3 to produce the final embedding $\hat{o}$.

Temporal Attention Encoder (TAE). 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 summarised by the following equations for all $t \in [1, \ldots, T]$ and $h \in [1, \cdots, n_h]$:

$$k_h^{(t)}, q_h^{(t)} = \text{FC}_1^{(h)}\left(e^{(t)} + p^{(t)}\right) \tag{1.5}$$

$$\hat{q}_h = \text{FC}_2^{(h)}\left(\text{mean}\left(\{q_h^{(t)}\}_{t=1}^{T}\right)\right) \tag{1.6}$$

$$a_h = \text{softmax}\left(\frac{1}{\sqrt{d_k}}\left[\hat{q}_h \cdot k_h^{(t)}\right]_{t=1}^{T}\right) \tag{1.7}$$

$$o_h = \sum_{t=1}^{T} a_h[t]\left(e^{(t)} + p^{(t)}\right) \tag{1.8}$$

$$\hat{o} = \text{MLP}_3\left([o_1, \cdots, o_{n_h}]\right) . \tag{1.9}$$

1.4 Numerical Experiments: PSE+TAE

In this section, we study experimentally the validity of our proposed spatial and temporal encoders. We compare them to several reimplemented state-of-the-art approaches on the dataset introduced in Section 1.1.

1.4.1 Experimental setting

1.4.1.1 PSE+TAE spatio-temporal encoder

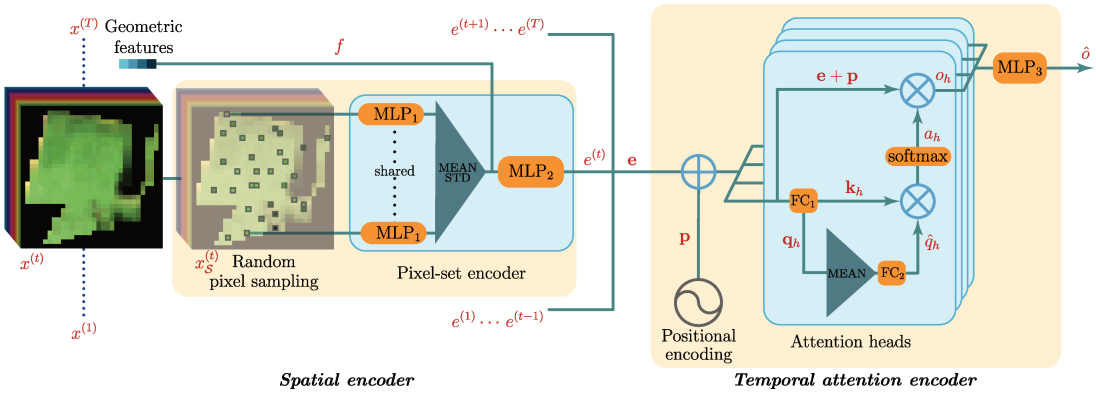


Figure 1.9: PSE-TAE. Schematic view of our spatio-temporal encoder. The input image sequence is encoded by a shared PSE and the resulting sequence of embeddings is processed by a TAE. Variables in bold are tensors concatenated along the temporal dimension, *e.g.*, $\mathbf{e} = [e^{(0)}, \cdots, e^{(T)}]$.

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 1.9. Finally, the resulting embedding is processed by an MLP decoder MLP$_4$ to produce class logits y:

$$y = \text{MLP}_4\left(\hat{o}\right) \ . \tag{1.10}$$

1.4.1.2 Competing Methods

We compare our approach to recent state-of-the-art deep learning algorithms operating on similar datasets, which we have reimplemented. All share the same decoding layer configuration MLP_4. As in Section 1.1, we ensure a fair comparison by implementing models with around 150k parameters.

CNN+GRU We first use the CNN+GRU architecture introduced in Section 1.1.

CNN+TempCNN Pelletier *et al.* [119] propose to use one-dimensional temporal convolution to address the sequential nature of the observations. While their approach is applied to a per-pixel classification task and therefore not comparable, we have implemented a variation of CNN+GRU in which the GRUs are replaced with one-dimensional convolutions as the closest translation of their ideas. 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.

Transformer Rußwurm & Körner [133] perform object-based classification with the encoder part of the Transformer network. They do not use a spatial encoder and compute the average values of the different spectral bands over each parcel. Furthermore they produce a single embedding for the whole sequence with a global maximum pooling through the temporal dimension of the output sequence. We reimplemented the same pipeline and simply modified the hyperparameters to match the 150k parameter constraint.

ConvLSTM Rußwurm *et al.* [131] process the time series of *patch* images with a ConvLSTM network [183] for pixel-based classification. We adapt the architecture to the parcel-based setting by using the spatially-averaged last hidden state of the ConvLSTM cell to be processed by MLP_4.

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[8].

1.4.1.3 Implementation details

All architectures presented here are implemented in PyTorch and released on GitHub[†]. 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. We chose the hyperparameters of each architecture presented in the numerical experiments such that they all have approximately 150k trainable parameters. The exact configuration of our network and the competing methods are displayed in Table 1.5 and Table 1.6 respectively. We use the Adam optimizer[76] with its default values ($lr = 10^{-3}, \quad \beta = (0.9, 0.999)$) and a batch size of 128 parcels. We train the models with focal loss[92] ($\gamma = 1$) 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 5.10^{-2} on the values of the pixels, normalised channel-wise and for each date individually.

1.4.2 Dataset

We use the same dataset as in Section 1.1.2.1, with a slightly more fine-grained 20-class nomenclature (see Figure 1.10). In order to evaluate both ours and convolution-based methods, we organize the parcels into two different formats: patches and pixel sets.

[†] github.com/VSainteuf/pytorch-psetae

Table 1.5: PSE+TAE hyperparameters. Configuration of our model chosen for the numerical experiments. The dimension of each successive feature space is given for MLPs and fully connected layers. We show the corresponding number of trainable parameters on the last column. In agreement with our conclusion of Section 1.1, most of the trainable parameters are allocated to temporal encoding.

Modules	Hyperparameters	Number of parameters
PSE		19 936
S	64	
MLP_1	$10 \to 32 \to 64$	
MLP_2	$132 \to 128$	
TAE		116 480
d_e, d_k, n_h	128, 32, 4	
FC_1	$128 \to (32 \times 2)$	
FC_2	$32 \to 32$	
MLP_3	$512 \to 128 \to 128$	
Decoder		11 180
MLP_4	$128 \to 64 \to 32 \to 20$	
Total		147 604

Table 1.6: Hyperparameters of the competing architectures. For all models we use the same values for the decoder MLP_3.

	Number of parameters
CNN+GRU	144 204
• 3×3 convolutions: 32, 32, 64 kernels • Global average pooling • Fully connected layer: 128 neurons • Hidden state size: 130	
CNN+TempCNN	156 788
• 3×3 convolutions: 32, 32, 64 kernels • Global average pooling • Fully connected layer: 64 neurons • Temporal convolutions: 32, 32, 64 kernels of size 3 • Flatten layer	
Transformer	178 504
• $d_k = 32, d_v = 64, d_{model} = 128, d_{inner} = 256$ • $n_h = 4, n_{layer} = 1$	
ConvLSTM	178 356
• Hidden feature maps: 64	
RF	
• Number of trees: 100	

In the *patch* format, we resize each parcel into a tensor of size $T \times C \times 32 \times 32$ by interpolating each spectral channel and temporal acquisition independently into patches of fixed size 32×32. 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 1.1). This same size of 32 pixels was used in [43], while a larger 48×48 patch size was used in [131], albeit for a pixel-wise classification task.

For the *pixel-set* format, the pixels of each parcel are stored in arbitrary order into a tensor of

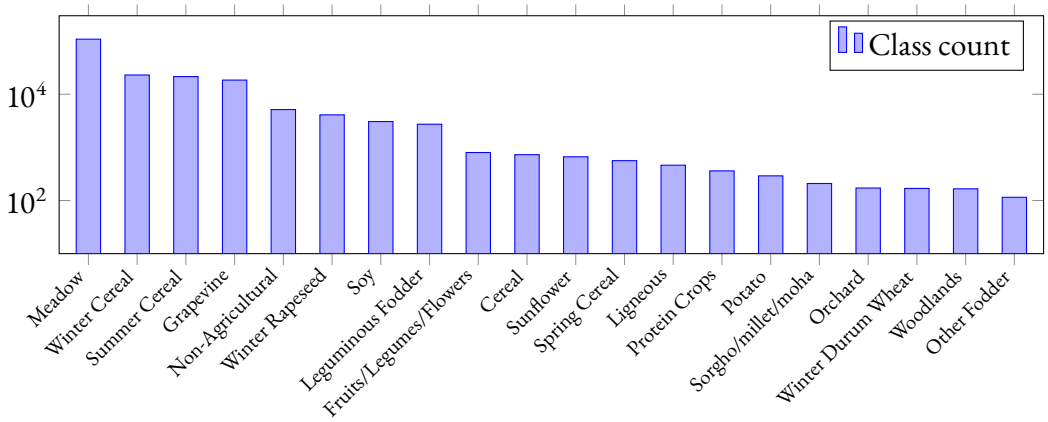


Figure 1.10: Class breakdown. We plot the number of agricultural parcels belonging to each class on a semi logarithmic scale.

size $T \times C \times N$, with N the total number of pixels in a given parcel. Note that this format will neither loose 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. We show the class break-down on the AOI in Figure 1.10. The dataset is highly imbalanced as is often the case in such real word applications, and this motivates the use of the focal loss to train our models.

1.4.3 Results

1.4.3.1 Comparison with state-of-the-art

We present the results of our experiments in Table 1.10. Our proposed architecture outperforms the other deep learning models in Overall Accuracy (OA) by 0.4pt, and mean per-class Intersect over Union (mIoU) by 3 to 9pts. 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

Table 1.7: Classification experiment. 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.

	OA	mIoU	Training (s/epoch)	Inference (s/dataset)	Disk Size Gb
PSE+TAE	**94.2** ±0.1	**50.9** ±0.8	158	**149**	**28.6 / 12.3**[1]
CNN+GRU (Section 1.1)	93.8 ±0.3	48.1 ±0.6	656	633	98.1
CNN+TempCNN [119]	93.3 ±0.2	47.5 ±1.0	635	608	98.1
Transformer [133]	93.0 ±0.2	46.3 ±0.9	13	420 + 4[3]	**28.6 / 0.22**[4]
ConvLSTM [131]	92.5 ±0.5	42.1 ±1.2	1 283	666	98.1
Random Forest [8]	91.6 ±1.7	32.5 ±1.4	**293**[2]	420 + 4[3]	**28.6 / 0.44** [4]

task alone, *i.e.*, when only S pixels per parcel are kept. This speed-up is due to the improved loading time as the pixel set dataset is smaller, but also to the inference and backpropagation time, as detailed in Table 1.9. While the temporal convolutions of TempCNN are faster to train, they yield worse performance and suffer from the limitations discussed earlier. The Transformer method, which processes precomputed parcel means, is also faster to train, but only achieves a 46.3 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.

1.4.3.2 Ablation Studies

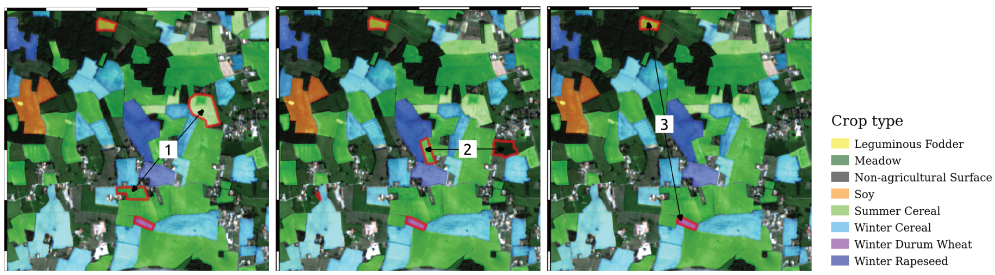
In order to independently assess the contribution of the spatial and temporal components of our proposed architecture, we present in Table 1.8 the results obtained when alternatively replacing the PSE by a CNN (CNN+TAE) or the TAE by a GRU (PSE+GRU).

Table 1.8: Ablation study. We assess the impact of our different design choices.

	OA	mIoU
PSE+TAE (ours)	**94.2** ±0.1	**50.9** ±0.8
$\hat{q} = q^{(T)}$	94.2 ±0.1	50.7 ±0.5
$S = 16$	94.3 ±0.2	50.5 ±0.8
$\hat{q} = \max_t q^{(t)}$	94.2 ±0.2	50.3 ±0.7
$S = 32$	94.2 ±0.1	50.1 ±0.5
No geometric features	93.9 ±0.1	50.0 ±0.7
PSE+Transformer+$\hat{q}$	94.1 ±0.2	49.5 ±0.7
CNN+TAE	94.0 ±0.1	49.2 ±1.1
MS+TAE	93.7 ±0.1	48.9 ±0.9
PSE+GRU+$\mathbf{p}$	93.6 ±0.2	48.7 ±0.3
PSE+GRU	93.6 ±0.2	47.3 ±0.3
PSE+Transformer	93.4 ±0.2	46.6 ±0.9

Contribution of the PSE. As seen in Table 1.8, the PSE accounts for an increase of 1.7pts of mIoU compared to the CNN-based model (CNN+TAE). This supports both the hypothesis that CNNs are only partly relevant for parcel classification on Sentinel-2 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 (see Table 1.9). Additionally, we train a TAE on precomputed 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 1.8, the performance of our architecture without geometric features f. The resulting 0.9pt decrease in mIoU confirms that geometric information plays a role in the classification process. We note that, even without such features, our proposed approach out-



(a) CNN+TAE **(b)** PSE+TAE *no f* **(c)** PSE+TAE

Figure 1.11: Qualitative results. Example of test-errors of three architectures on a sub-region of the dataset. The images consist in the RGB channels of a single Sentinel-2 observation overlayed with a color-coded representation of the different parcels' crop types. Those parcels that were wrongly classified by the model are highlighted with a solid red stroke. The scale is given by the 500 meter zebra strips. We compare the errors of the CNN+TAE (a), the PSE+TAE *without geometric features* (b), and the complete PSE+TAE (c).

performs the convolution-based model (CNN+TAE). We also show a visual representation of our model's prediction errors compared to those of a CNN+TAE architecture on Figure 1.11. While the PSE+TAE without f corrects some errors made by the CNN+TAE (the two parcels marked with (1) on Figure 1.11a) , it produces new errors ((2) on Figure 1.11b) . The geometric features in the full PSE+TAE architecture allow to correctly classify the latter and yield a wrong classification only for the two parcels (3) (Figure 1.11c) that belong to hard classes (*Winter Durum Wheat* and *Leguminous Fodder*) and where incorrectly classified by all models.

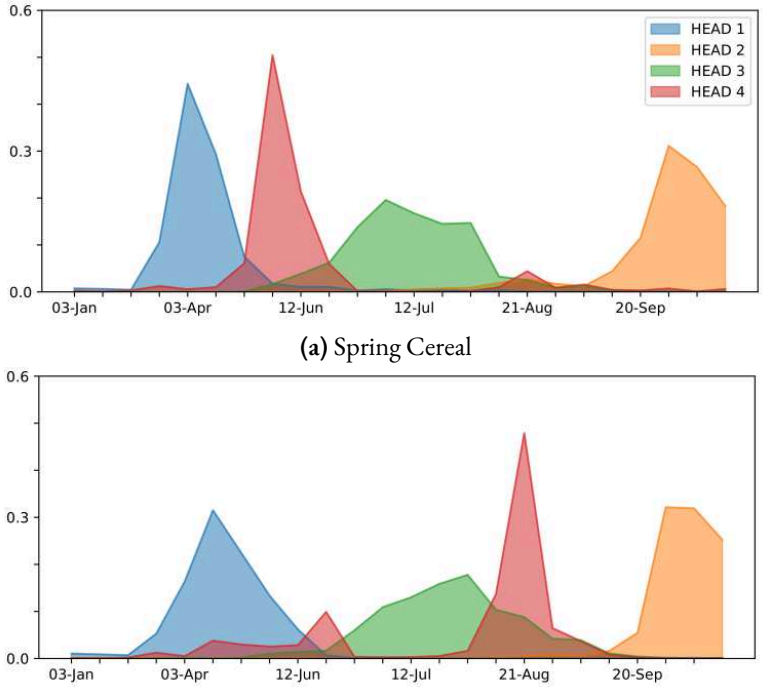
We have tried replacing the handcrafted geometric features f with a CNN operating on the binary mask of the parcel. However, the gains were minimal, and we removed this extra step for the sake of simplicity.

Lastly, we tried training our architecture with a reduced number of sampled pixels ($S = 16$ and $S = 32$). The model maintains a good performance with an mIoU over 50pts. 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 3.6pts mIoU (Table 1.8). The TAE not only produces a better classification but also trains faster thanks to parallelisation.

Unlike the comparison between Transformer and RNNs architectures in [133], 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 2.2pts mIoU. This shows that the improvement brought by the TAE is due to both its structure and the use of positional encoding.



(a) Spring Cereal

(b) Summer Cereal

Figure 1.12: Attention masks. Average attention masks of the TAE heads, obtained from 128 samples of spring (a), and summer (b) cereal parcels.

Design of the TAE. To evaluate the benefits of our different contributions over the Transformer, we adapted the architecture presented in [133] to use a PSE network instead of spectral means for embedding parcels (PSE+Transformer), for a performance 4.3pts below our TAE. By replacing the proposed temporal max-pooling by our our master query forming scheme (PSE+Transformer+$\hat{q}$), we observed an increase of 2.9pts mIoU. The remaining 1.4pts mIoU 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 ($\hat{q} = \max_t q^{(t)}$) 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.

On Figure 1.12, we show a qualitative illustration of head specialization in the TAE. We plot the average attention masks of each attention head for two classes of cereal parcels. We note that each head focuses on a different period of the agricultural year and can be adaptive to the time series being processed: head 4 focuses on the end of spring for *Spring Cereal* samples, and on the end of summer for *Summer Cereal* samples.

We also provide a breakdown of the processing times during training for the different architectures in Table 1.9. The average time per batch is decomposed into data loading time, forward pass, and gradient back-propagation. Note that the Transformer model operates on precomputed spatial descriptors and is hence significantly faster than the other models.

1.4.4 Concluding remarks

In this section, 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

Table 1.9: Processing times. Comparison of processing time for different methods for batches of 128 parcels. We can see that the processing time is dominated by the loading time except for the Transformer which processes pre-computed means.

Time in ms/batch	Total	Loading	Forward	Backward
PSE+TAE (ours)	**107**	**85**	11	**11**
CNN+TempCNN	381	365	**4**	12
CNN+GRU	437	365	14	58
Transformer	8	1	2	5
ConvLSTM	530	365	61	104

series, we adapted the Transformer architecture [171] for embedding time sequences. We introduced a master query forming strategy and exploited the fact that our network learns end-to-end to simplify some operations.

Evaluated on our benchmark of agricultural parcels, our method produces a better classification than all other reimplemented 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.

Our results suggest that set-based encoders are a promising and overlooked paradigm for working with the coarser resolutions of remote sensing applications. Likewise, attention-based models are an interesting venue to explore for analysing the temporal profiles of satellite time series.

1.5 Lightweight Temporal Attention Encoder (L-TAE)

1.5.1 Motivation

Time series of remote sensing data provide a wealth of useful information for Earth monitoring. However, they are also typically very large, and their analysis is resource-intensive. For example, the Sentinel-2 satellites gather over 25 TB of data every year in the EU. This motivates the design of parsimonious methods. In this section, we build on the previous adaptation of the Transformer to crop type classification. We aim for a lighter model, both in terms of computation and trainable parameters, without sacrificing on classification performance.

1.5.2 Methods

Throughout this section, we consider a generic input time series of length T comprised of d_e-dimensional feature vectors $\mathbf{e} = [e^{(1)}, \cdots, e^{(T)}] \in \mathbb{R}^{d_e \times T}$. For example, such vectors can be PSE encodings of an input SITS (see Section 1.2).

We build on our efforts to adapt multi-headed self-attention (see Section 1.3) to the task of sequence embedding for crop type mapping. Our focus is on efficiency, both in terms of parameter count and computational load. We thus propose the following modifications to our TAE encoder.

Channel Grouping. we propose to split the d_e channels of the input elements into n_h groups of size $d_{e'} = d_e/n_h$ with n_h being the number of heads[‡], in the manner of Wu *et al.* [181]. We denote by $e_h^{(t)}$ the groups of input channels for the h-th group of the t-th element of the input sequence (1.11).

As with the TAE, we encode the number of days elapsed since the beginning of the growing season into an $d_{e'}$-dimensional positional vector p of characteristic scale $\tau = 1000$ (1.12). Since this information is required by each head, p is duplicated and added to each channel group. Each head

[‡] d_e and n_h are typically powers of 2 and $d_e > n_h$, ensuring that $d_{e'}$ remains integer.

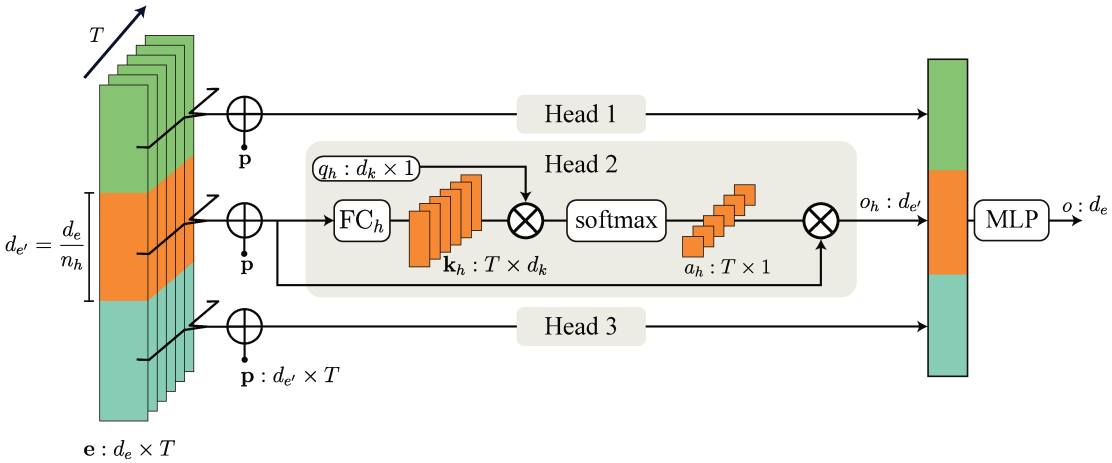


Figure 1.13: L-TAE. Our module processes an input sequence $\mathbf{e}$ of T vectors of size d_e, with $n_h = 3$ heads and keys of size d_k. The channels of the input embeddings are distributed among heads. Each head uses a learnt query q_h, while a linear layer FC_h maps inputs to keys. The outputs of all heads are concatenated into a vector with the same size as the input embeddings, regardless of the number of heads.

operates in parallel on its corresponding group of channels, thus accelerating the costly computation of keys and queries. This also allows for each head to specialize alongside its channel group and avoid redundant operations between heads.

Note that our channel grouping strategy differs from the channel reduction approach implemented in the original Transformer[171]. In the Transformer, the input embeddings are mapped to value vectors v which are typically smaller than the input embeddings. As a result, each head uses the complete information contained in the input embeddings. In our setting, we do not resort to mappings but only split the channel dimension, as a result each head has only access to a fragment of the information contained in the input embeddings.

Query-as-Parameter. We define the d_k-dimensional master query q_h of each head h as a model parameter instead of the results of a linear layer. The immediate benefit is a further reduction of the number of parameters, while the lack of flexibility is compensated by the larger number of available

heads.

Attention Masks. As a result, only the keys are obtained with a learned linear layer (Equation 1.13), while values are bypassed ($v^{(t)} = e^{(t)}$), and the queries are model parameters. The attention masks $a_h \in [0,1]^T$ of each head h are defined as the scaled *softmax* of the dot-product between the keys and the master query (Equation 1.14). The outputs o_h of each head are defined as the sum in the temporal dimension of the corresponding inputs weighted by the attention mask a_h (Equation 1.15). Finally, the heads' outputs are concatenated into a vector of size d_e and processed by a multi-layer perceptron MLP to the desired size (Equation 1.16).

In Figure 1.13, we represent a schematic representation of our network. The different steps of the L-TAE can also be condensed by the following operations, for $h = 1 \cdots n_h$ and $t = 1 \cdots T$:

$$e_h^{(t)} = \left[e^{(t)} \left[(h-1)d_{e'} + i \right] \right]_{i=1}^{d_{e'}} \tag{1.11}$$

$$p^{(t)} = \left[\sin\left(\text{day}(t) / \tau^{\frac{i}{d_{e'}}} \right) \right]_{i=1}^{d_{e'}} \tag{1.12}$$

$$k_h^{(t)} = \text{FC}_h(e_h^{(t)} + p^{(t)}) \tag{1.13}$$

$$a_h = \text{softmax}\left(\frac{1}{\sqrt{d_k}} \left[q_h \cdot k_h^{(t)} \right]_{t=1}^{T} \right) \tag{1.14}$$

$$o_h = \sum_{t=1}^{T} a_h[t] \left(e_h^{(t)} + p^{(t)} \right) \tag{1.15}$$

$$o = \text{MLP}([o_1, \cdots, o_{n_h}]) \,. \tag{1.16}$$

1.6 Numerical experiments: PSE+L-TAE

1.6.1 Experimental setting

We combine our L-TAE with a PSE into an end-to-end trainable architecture. We use the same dataset and competing methods as in Section 1.4. In order to perform a fair comparison, we chose configurations corresponding to around 150k parameters for all methods. We report the results in Table 1.10 alongside the theoretical number of floating point operations (in FLOPs) required for the sequence embedding modules to process a single sequence at the inference time.

Moreover, we complement this first experiment by comparing the performance of different configurations of sequence embedding algorithms, and plot the performance with respect to the number of parameters. To remove the effects of the different spatial encoders, we use the same spatial encoder (a PSE) in all models for this experiment. We only adapt the last linear layer of the spatial encoder to produce embeddings of the desired dimensions.

1.6.2 Implementation details

All training and implementation details are the same as in Section 1.4. The hyperparameters of all models presented in this section are given in Table 1.11.

1.6.3 Results

1.6.3.1 Comparison with state-of-the-art

In Table 1.10, we report the performances of competing methods and L-TAE, all obtained with 5-fold cross-validation. Our L-TAE architecture outperforms other methods on this dataset both in overall accuracy and mIoU. While the OA is essentially unchanged compared to the TAE, the increase of

Table 1.10: Classification experiment. Performance of our model and competing approaches parameterised to all have 150k parameters approximately. MFLOPs is the number of floating points operations (in 10^6 FLOPs) *in the temporal feature extraction module* and for one sequence. This only applies to networks which have a clearly separated temporal module.

	OA	mIoU	MFLOPs
PSE+L-TAE	**94.3** ±0.2	**51.7** ±0.4	**0.18**
PSE+TAE (Section 1.4)	94.2 ±0.1	50.9 ±0.8	1.7
CNN+GRU (Section 1.1)	93.8 ±0.3	48.1 ±0.6	3.6
CNN+TempCNN[119]	93.3 ±0.2	47.5 ±1.0	0.81
Transformer[133]	92.2 ±0.3	42.8 ±1.1	1.1
ConvLSTM[131]	92.5 ±0.5	42.1 ±1.2	-
Random Forest[8]	91.6 ±1.7	32.5 ±1.4	-

0.8pt mIoU is noteworthy since our model is not only simpler but also less computationally demanding by almost an order of magnitude.

We would like to emphasize that FLOP counts do not necessarily reflect the computational speed of the model in practice. In our non-distributed implementation, the total inference times are dominated by loading times and the spatial embedding module. However, this metric serves to illustrate the simplicity and efficiency of our network.

In Figure 1.14, we represent the average attention masks of a 16-head L-TAE for two different classes. We observe that the masks of the different heads focus on narrow and distinct time-extents, *i.e.*, display a high degree of specialisation. We also note that the masks are adaptive to the parcels' crop types. This suggests that the attention heads are able to cater the learned features to the plant types considered. We argue that our channel grouping strategy, in which each head processes distinct time-stamped features, allows for this specialisation and leads to an efficient use of trainable parameters.

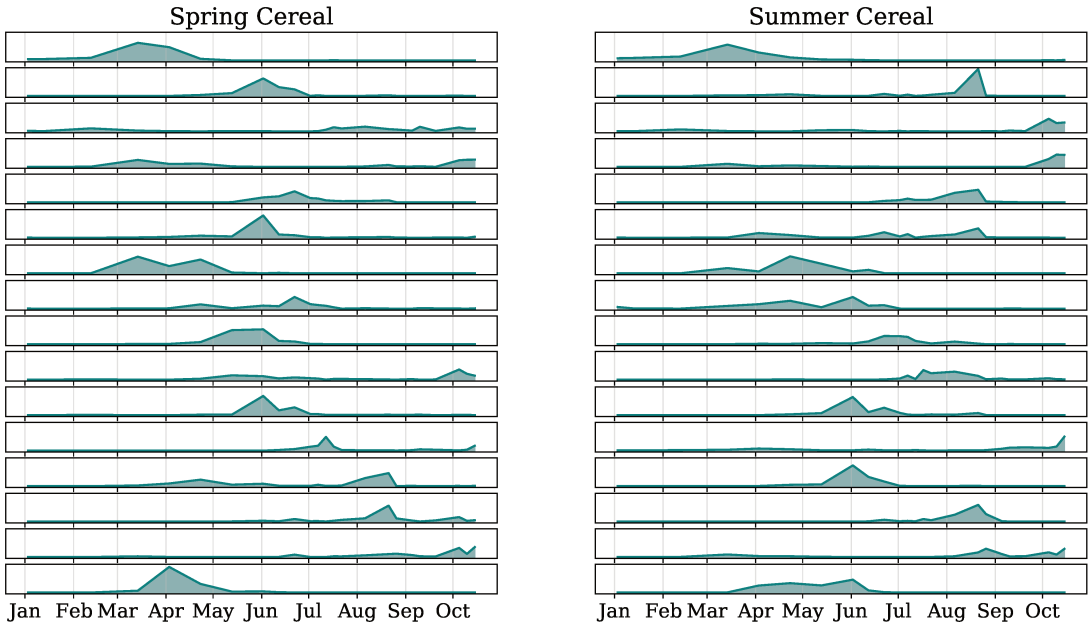


Figure 1.14: Attention masks. Average attention masks of the L-TAE for parcels of classes Spring Cereal (left) and Summer Cereal (right), for a model with 16 heads (from top to bottom). The masks illustrate how each head focuses on short temporal intervals which depend on crop type.

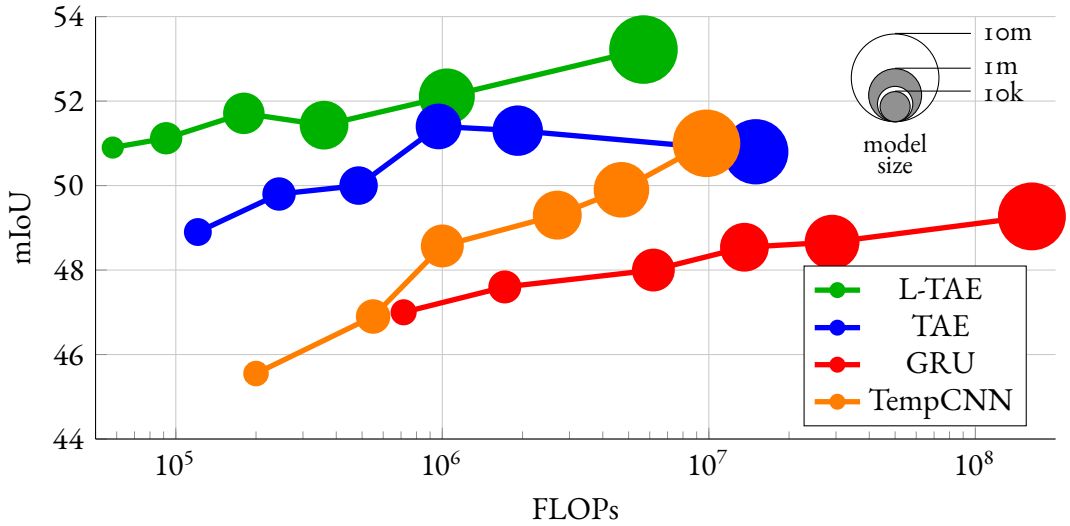


Figure 1.15: Performance complexity tradeoff. Performance (in mIoU, average over 5 runs) of different temporal encoders plotted with respect to the number of FLOPs necessary to process one sequence.The model size (number of trainable parameters) is represented by the size of the markers. The L-TAE outperforms other models across all model sizes and processing requirements. The smallest L-TAE instance—with under 9k parameters—outperforms all non-TAE configurations while only necessitating 58k FLOPs per sequence.

1.6.3.2 Parameter efficiency

Furthermore, our network maintains a high precision even with a drastic decrease in the parameter count, as illustrated in Figure 1.15. We evaluate the four best performing sequence embedding modules (L-TAE, TAE, GRU, TempCNN) in the previous experiment with different configurations, ranging from $9k$ to $3M$ parameters. These algorithms all operate with the same decoder and spatial module: a PSE and decoder layer, totaling 31k parameters. The smallest L-TAE configuration, with only $9k$ parameters, achieves a better mIoU score than a TAE with almost $110k$ parameters, a TempCNN with over $700k$ parameters, and a GRU with $3M$ parameters. See Table 1.11 for the detailed configurations corresponding to each point.

Table 1.11: Hyperparameters. Configurations of the L-TAE, TAE, GRU, and TempCNN instances used to obtain Figure 1.15.

Parameters	d_e	n_h	d_k	MLP
L-TAE				
9 k	128	8	8	128
34 k	128	16	8	128 - 128
112 k	256	16	8	256 - 128
288 k	512	32	8	512 - 128
740 k	1024	32	8	1024 - 256 - 128
3840 k	2048	64	8	2048 - 1024 - 256 - 128
TAE				
19 k	64	2	8	128 - 128
39 k	64	4	8	256 - 128
76 k	128	4	8	512 - 128
195 k	256	4	8	1024 - 128
360 k	256	4	8	1024 - 256 - 128
641 k	256	8	8	2048 - 256 - 128
2592 k	1024	8	16	8192 - 256 - 128

Parameters	Hidden Size
15k	32
37k	64
134k	156
296k	256
636k	400
3545k	1024

GRU

Parameters	Kernels	FC
14k	16 - 16 - 16	16 - 16
45k	32 - 32 - 32	32 - 32
136k	64 - 64	64
296k	128 - 128	64
702k	128 - 128 - 128	180
3362k	64 - 128 - 256	512 - 128

TempCNN

1.6.3.3 Ablation Study and Robustness Assessment

In Table 1.12, we report the performance of our proposed L-TAE architecture with different configurations of the following hyper-parameters: the number of heads n_h, dimension of keys d_k, and number of channels d_e in the input sequence. We note that our model retains a consistent performance throughout all configurations.

Number of heads. The number of heads seems to only have a limited effect on the performance. We hypothesize that while a higher number of heads n_h is beneficial, a smaller group size $d_{e'}$ is, however, detrimental.

Key Dimension. Our experiments show that smaller key dimensions than the typical values used in NLP or for the TAE ($d_k = 32$) perform better on our problem. Even 2-dimensional keys allow for the L-TAE to achieve performances similar to the TAE.

Input Dimension. The variation in performance observed with larger input embeddings is expected: it corresponds to a richer representation. However, the returns are decreasing on the considered dataset with respect to the number of incurred parameters.

Query-as-Parameter. In order to evaluate the impact of our different design choices, we train a variation of our network with the same master query scheme than the TAE. The larger resulting linear layer increases the size of the model for a total of 170k parameters, resulting in a mIoU of only 49.7. This indicates that the query-as-parameter scheme is not only beneficial in terms of compactness but also performance.

Table 1.12: Hyperparameter robustness. Impact of several hyper-parameters on the performance of our method. Underlined, the default parameters values in this study; in **bold**, the best performance.

n_h	Params.	mIoU
2	114k	51.6
4	118k	51.0
8	127k	51.2
16	143k	**51.7**
32	176k	51.2

d_k	Params.	mIoU
2	118k	50.7
4	127k	51.3
8	143k	**51.7**
16	176k	50.8
32	242k	51.2

d_e	Params.	mIoU
32	46k	49.6
64	59k	49.6
128	65k	51.1
256	143k	**51.7**
512	254k	51.4

Table 1.13: Computational complexity. Asymptotic complexity of different temporal extraction modules for the computation of keys, attention masks, and output vectors. For the GRU, the complexity of the memory update is given in the Keys and Mask columns. X is the size of the output vector. d_r is the size of the hidden state of the GRU.

Method	Keys	Mask	Output
L-TAE	$O(T d_e d_k)$	$O(n_h T d_k)$	$O(d_e X)$
TAE	$O(T n_h d_e d_k)$	$O(n_h T d_k)$	$O(n_h d_e X)$
Transformer	$O(T n_h d_e d_k)$	$O(n_h T^2 d_k)$	$O(n_h d_e X)$
GRU	$O\left(T d_r (d_e + d_r)\right)$		$O(d_r X)$

1.6.4 Computational Complexity

In Table 1.13, we report the asymptotic complexity of different sequence embedding algorithms. For the L-TAE, the channel grouping strategy removes the influence of n_h in the computation of keys and outputs compared to a TAE or a Transformer. Note as well, that the fact that a single query is computed in TAE and L-TAE removes the unnecessary quadratic complexity in sequence length T that hinders mask computation in the Transformer. The complexity of the L-TAE is also lower than the GRU's as M, the size of the hidden state, is typically larger than d_k (130 vs 8 in the experiments presented in Table 1.10).

1.6.5 Concluding remarks

We presented a new lightweight network for embedding sequences of observations such as satellite time series. Thanks to a channel grouping strategy and the definition of the master query as a trainable parameter, our proposed approach is more compact and computationally efficient than other attention-based architectures. Evaluated on our open-access satellite dataset *S2-Agri*, the L-TAE performs better than state-of-the-art approaches, with significantly fewer parameters and a reduced computational load, opening the way for continent-scale automated analysis of Earth observation.

1.7 Conclusion

In this chapter, we considered the problem of crop type mapping as a parcel-based classification task. We started our analysis with a study on the spatial and temporal structures of satellite image time series for crop type mapping. This study showed the significance of the temporal dimension of Sentinel-2 data for better classification performance. We also showed that convolutional nets only performed marginally better than handcrafted spatial features. We attributed this to the limited spatial resolution of Sentinel-2 compared to the typical scale of texture on agricultural parcels. This motivated our design of the PSE which, inspired by the PointNet architecture, considers images as unordered sets of pixels. We also adapted the Transformer, and taking into account the key differences between the typical NLP task and our crop type classification problem, we introduced the TAE and its more efficient variant the L-TAE. Together, these methods set a new state-of-the-art for parcel-based crop type mapping from SITS. These results show that advances in active fields of deep learning such as computer vision and natural language processing are also relevant for remote sensing applications. Moreover, we showed that adapting these methods, accounting for the specificities of the problem and the data at hand, is a key step to push the state-of-the-art forward. As a matter of fact, we show on Table 1.14 the results obtained by an independent research team[80] for parcel-based crop type classification on DENETHOR, a new large-scale dataset they curate. Their experiments confirm the superior performance of our PSE+L-TAE compared to architectures combining "off-the-shelf" solutions for spatial (*e.g.*, ResNet18) and temporal encoding (*e.g.*, Transformer).

Table 1.14: DENETHOR. Overall accuracies of different architectures on the DENETHOR dataset for parcel-based classification, taken from Kondmann *et al.* [80]. These experiments, led by an independant research team , confirm both the superior performance of our PSE+L-TAE method (see Section 1.5), and our finding that convolutional spatial encoders are not well suited for parcel-based classification (see Section 1.1).

Spatial Encoder	Temporal Encoder			
	TempCNN	MSResNet	LSTM	Transformer
ResNet18	52.2%	49.5%	44.6%	43.6%
SqueezeNet	53.9%	49.8%	35.9%	42.6%
MobileNetv3	53.2%	54.3%	43.5%	48.1%
PixelAverage	64.5%	58.8%	48.4%	52.6%
	Pixel-Set Encoding and Self-Attention			
PSE+TAE	65.0%			
PSE+L-TAE	**67.3%**			

Form and substance are one and the same. Form is the life expression and substance the living painting.

Asger Jorn

2

Pixel-based segmentation methods

In this chapter, we cast crop type mapping as a segmentation problem. Indeed, in many countries the precise land parcel identification system is not available and parcel-based methods are thus not applicable. Our aim now consists in retrieving from the Satellite Image Time Series (SITS) all the information contained in the land parcel identification system: the shape of each individual parcel as well as its content.

In segmentation, predictions are made at pixel level, and thus require different encoders than

those seen previously for parcel-based classification. In a first section we present our U-Net with Temporal Attention Encoder (U-TAE) architecture for spatio-temporal encoding of SITS for segmentation problems. U-TAE allows to encode a sequence of images into a feature map with the same spatial resolution. We evaluate this architecture for semantic segmentation and set a new state-of-the-art on this task. Second, we combine this encoder with a single-stage instance segmentation module that we adapted to perform the desired task of retrieving non-overlapping instance masks with associated semantic predictions. This allows us to set the first state-of-the-art for the task of panoptic segmentation of agricultural parcels on SITS and tease out several key challenges of this task.

2.1 U-Net with Temporal Attention Encoder (U-TAE)

In this section, we introduce U-TAE, a novel spatio-temporal encoder combining multi-scale spatial convolutions [127] and a temporal self-attention mechanism [40] which learns to focus on the most salient acquisitions across the sequence. While convolutional-recurrent methods are limited to extracting temporal features at the highest [131] or lowest [135] spatial resolutions, our proposed method can use the predicted temporal masks to extract specialised and adaptive spatio-temporal features at different resolutions simultaneously. Additionally, we introduce PASTIS, a large-scale dataset of SITS with semantic and panoptic annotations.

2.1.1 Motivation

Pixel-precise segmentation of satellite image time series entails producing a feature map of the same resolution as the input images. In this feature map, each pixel contains the embedding of the corresponding spatial location in the area of interest. Such a map can be obtained by encoding the sequence of observations for each pixel separately and assembling the resulting embeddings along the two spatial dimensions. However, in this scheme, the embedding of each pixel ignores the spatial structure of the acquisitions. This motivates the design of segmentation methods that encode all pixels of an area of interest together and thus leverage the spatial structure. In practice, in the computer vision literature, it has been observed [90] that allowing a segmentation model to access the spatial context at different scales or resolutions is key for performance.

In the context of Earth Observation, models are applied at a large scale to considerable volumes of data. Computational efficiency is thus a key aspect to consider when designing such algorithms. At the time of writing, existing methods for satellite image time series encoding for segmentation either rely on recurrent neural nets or convolutions for temporal encoding. As seen in Section 1.4, recurrent neural nets, incur long training and inference times, and temporal convolutions, although faster,

are not well suited for irregularly sampled satellite image time series. We propose instead to design a spatio-temporal encoder leveraging our advances on self-attention-based temporal encoding. Furthermore, as per our previous remarks, we ensure that both spatial encoding and temporal encoding are performed at different spatial resolutions, to fully leverage the spatial structure of SITS.

2.1.2 Methods

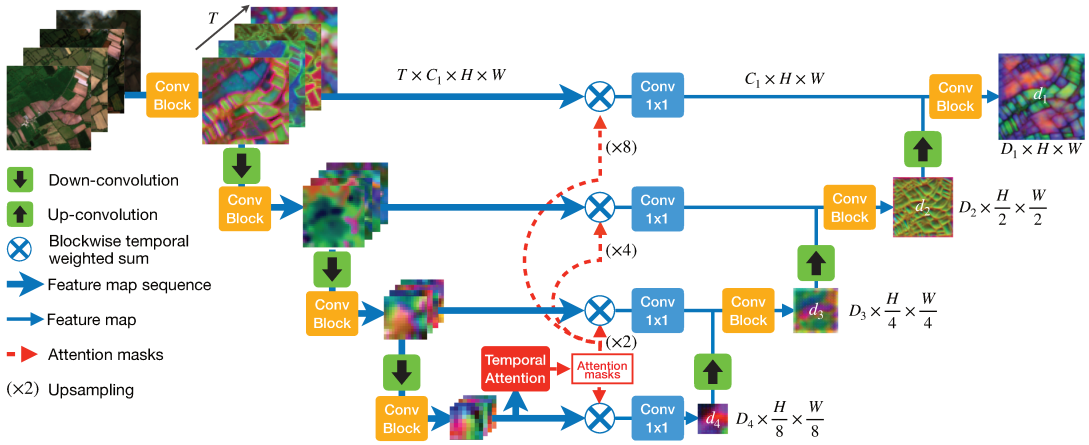


Figure 2.1: Spatio-temporal Encoding. A sequence of images is processed in parallel by a shared convolutional encoder. At the lowest spatial resolution, an attention-based temporal encoder produces a set of temporal attention masks for each pixel, which are then spatially interpolated at all resolutions. These masks are used to collapse the temporal dimension of the feature map sequences into a single map per resolution. A convolutional decoder then computes features at all resolution levels. All convolutions operate purely on the spatial and channel dimensions, and we use strided convolutions for both spatial up and down-sampling. The feature maps are projected in RGB space to help visual interpretation.

We consider an image time sequence X, organised into a four-dimensional tensor of shape $T \times C \times H \times W$, with T the length of the sequence, C the number of channels, and $H \times W$ the spatial extent.

Our model, dubbed U-TAE, encodes a sequence X in three steps: **(i)** each image in the sequence is embedded independently by a shared multi-level spatial convolutional encoder, **(ii)** a temporal attention encoder collapses the temporal dimension of the resulting sequence of feature maps into a single

map for each level, **(iii)** a spatial convolutional decoder produces a single feature map with the same resolution as the input images, see Figure 2.1.

Spatial Encoding. We consider a convolutional encoder $\mathcal{E}$ with L levels $1, \cdots, L$. Each level is composed of a sequence of convolutions, Rectified Linear Unit (ReLu) activations, and normalisations. Except for the first level, each block starts with a strided convolution, dividing the resolution of the feature maps by a factor 2.

For each time stamp t simultaneously, the encoder $\mathcal{E}_l$ at level l takes as input the feature map of the previous level e_t^{l-1}, and outputs a feature map e_t^l of size $C_l \times H_l \times W_l$ with $H_l = H/2^{l-1}$ and $W_l = W/2^{l-1}$. The resulting feature maps are then temporally stacked into a feature map sequence e^l of size $T \times C_l \times H_l \times W_l$:

$$e^l = [\mathcal{E}_l(e_t^{l-1})]_{t=0}^{T} \text{ for } l \in [1, L] \,, \tag{2.1}$$

with $e^0 = X$ and $[\,\cdot\,]$ the concatenation operator along the temporal dimension. When constituting batches, we flatten the temporal and batch dimensions. Since each sequence comprises images acquired at different times, the batches' samples are not identically distributed. To address this issue, we use Group Normalisation [181] with 4 groups instead of Batch Normalisation [67] in the encoder.

Temporal Encoding. To obtain a single representation per sequence, we need to collapse the temporal dimension of each feature map sequence e^l before using them as *skip connections*. Convolutional-recurrent U-Net networks [153,135,118] only process the temporal dimension of the lowest resolution feature map with a temporal encoder. The rest of the skip connections are collapsed with a simple temporal average. This prevents the extraction of spatially adaptive and parcel-specific temporal patterns at higher resolutions. Conversely, processing the highest resolution would result in small spatial receptive fields for the temporal encoder, and an increased memory requirement. Instead, we propose

an attention-based scheme which only processes the temporal dimension at the lowest feature map resolution, but is able to utilize the predicted temporal attention masks at all resolutions simultaneously.

Based on its performance and computational efficiency, we choose the Lightweight-Temporal Attention Encoder (L-TAE) introduced in Section 1.5 to handle the temporal dimension. The L-TAE is a simplified multi-head self-attention network [171] in which the attention masks are directly applied to the input sequence of vectors instead of predicted *values*. Additionally, the L-TAE implements a channel grouping strategy similar to Group Normalisation [181].

We apply a shared L-TAE with G heads independently at each pixel of e^L, the feature map sequence at the lowest level resolution L. This generates G temporal attention masks for each pixel, which can be arranged into G tensors $a^{L,g}$ with values in $[0, 1]$ and of shape $T \times H_L \times W_L$:

$$a^{L,1}, \cdots, a^{L,G} = \text{L-TAE}(e^L) \text{ , applied pixelwise.} \tag{2.2}$$

In order to use these attention masks at all scale levels l of the encoder, we compute spatially-interpolated masks $a^{l,g}$ of shape $T \times H_l \times W_l$ for all l in $[1, L-1]$ and g in $[1, G]$ with bilinear interpolation:

$$a^{l,g} = \text{resize } a^{L,g} \text{ to } H_l \times W_l \,. \tag{2.3}$$

The interpolated masks $a^{l,g}$ at level l of the encoder are then used as if they were generated by a temporal attention module operating at this resolution. We apply the L-TAE channel grouping strategy at all resolution levels: the channels of each feature map sequence e^l are split into G contiguous groups $e^{l,1}, \cdots, e^{l,G}$ of identical shape $T \times C_l/G \times W_l \times H_l$. For each group g, the feature map sequence $e^{l,g}$ is averaged on the spatial dimension using $a^{l,g}$ as weights. The resulting maps are concatenated along the channel dimension and processed by a shared 1×1 convolution layer $\text{Conv}^l_{1\times1}$ of width C_l. We

denote by f^l the resulting map of size $C_l \times W_l \times H_l$ by :

$$f^l = \mathrm{Conv}^l_{1\times 1}\left(\left[\sum_{t=1}^{T} a_t^{l,g} \odot e_t^{l,g}\right]_{g=1}^{G}\right) , \tag{2.4}$$

with $[\,\cdot\,]$ the concatenation along the channel dimension and $\odot$ the termwise multiplication with channel broadcasting.

Spatial Decoding. We combine the feature maps f^l learned at the previous step with a convolutional decoder to obtain spatio-temporal features at all resolutions. The decoder is composed of $L-1$ blocks $\mathcal{D}_l$ for $1 \leq l < L$, with convolutions, ReLu activations, and BatchNorms[67]. Each decoder block uses a strided transposed convolution $\mathcal{D}_l^{\mathrm{up}}$ to up-sample the previous feature map.

The decoder at level l produces a feature map d^l of size $D_l \times H_l \times W_l$. In a U-Net fashion, the encoder's map at level l is concatenated with the output of the decoder block at level $l-1$:

$$d^l = \mathcal{D}_l([\mathcal{D}_l^{\mathrm{up}}(d^{l+1}), f^l]) \text{ for } l \in [1, L-1] , \tag{2.5}$$

with $d^L = f^L$ and $[\,\cdot\,]$ is the channelwise concatenation.

2.2 Numerical experiments: semantic segmentation

2.2.1 Implementation details

Our U-TAE has $L = 4$ resolution levels and a L-TAE with $G = 16$ heads and a key-query space of dimension $d_k = 4$. We use Group Normalisation with 16 groups at the input and output of the L-TAE, meaning that that the inputs of each head are layer-normalised. To produce semantic predictions, the feature map d_1 with highest resolution is set to have K channels, with K the number of classes. We can then interpret d_1 as pixel-wise predictions to be supervised with the cross-entropy

Table 2.1: Spatial encoding hyperparameters. Width of the feature maps outputted at each level of the encoding and decoding branches of the spatial module.

Encoder		Decoder	
e_1	64	d_1	32
e_2	64	d_2	32
e_3	64	d_3	64
e_4	128	d_4	128

loss. Note that we do not use the focal loss in this experiment, as the PASTIS dataset has a smaller class imbalance, see Section 2.2.3. In Table 2.1, we report the width of the feature maps outputted by each level of the U-TAE's encoder and decoder. Across the network, we use the the same convolutional block shown in Figure 2.2 and constituted of one 3×3 convolution from the input to the output's width, and one residual 3×3 convolution. In the encoding branch, we use Group Normalisation with 4 groups and Batch Normalisation in the decoding branch.

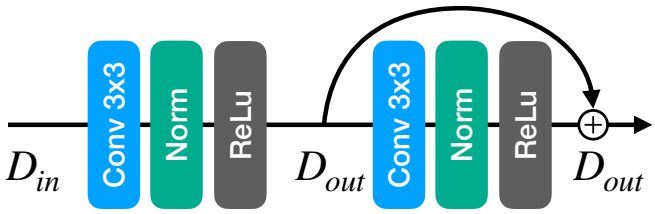


Figure 2.2: Convolutional block. Structure of the convolutional block used in the spatial encoder-decoder network. This block maps a feature map with D_{in} channels to a feature map with D_{out} channels.

2.2.2 Competing methods

We reimplemented six of the top-performing SITS encoders proposed in the literature. We present them succinctly here and refer the reader to the cited references for more details. We also assess the spatial scale at which the spatial and temporal encoding modules of each approach operates, as summarised in Table 2.2. Lastly, we provide the hyper-parametrisation we used for each method, to obtain models of similar size.

- **ConvLSTM**[131,142] and **ConvGRU**[9]. These approaches are recurrent neural networks in which all linear layers are replaced by spatial convolutions. These convolutions only operate at full resolution, hence both spatial and temporal encoding are performed at a single spatial scale. In our experiments, we set hidden sizes of 160 and 188 for the ConvLSTM and ConvGRU models respectively.

- **U-ConvLSTM**[135] and **U-BiConvLSTM**[98]. To reproduce these UNet-based architectures, we replaced the L-TAE in our architecture by either a ConvLSTM[142] or a bidirectional ConvLSTM. Skip connections are temporally averaged. In contrast to the original methods, we replaced the batch normalisation in the encoders with group normalisation, which significantly improved the results across-the-board. In these architectures, the successive downsampling operations in the U-Net ensure that spatial encoding is performed at different scales. Temporal encoding, on the other hand, only occurs at the lowest level. Indeed, since the skip connections on other levels are simple temporal means, only the feature maps with the coarsest spatial resolution are temporally encoded by the recurrent cell. The hidden state's size of the biConvLSTM is chosen as 32 in both directions, and 64 for ConvLSTM.

- **FPN-ConvLSTM**[98]. This model combines a Feature Pyramid Network (FPN)[91] to extract spatial features and a bidirectional ConvLSTM for the temporal dimension. For this architecture, the input sequence of images is first mapped to feature maps of 64 channels with two consecutive 3×3 convolution layers, followed by Group Normalisation and ReLu. A 5-level feature pyramid is then extracted for each date of the sequence by applying to the feature maps 4 different 3×3 convolution of respective dilation rates of 1, 2, 4 and 8. We obtain the fifth level of the pyramid with the spatial global average of the feature map. These 5 maps are concatenated along the channel dimension, and processed by a ConvLSTM with a hidden state size of 88. We found it beneficial to use a supplementary convolution before the ConvLSTM

to reduce the number of channels of the feature pyramid by a factor 2. Producing a feature pyramid before passing it to a ConvLSTM network, ensures that the ConvLSTM extracts spatio-temporal features taking into account multiple spatial scales. Yet, this scheme is computationally costly as the depth of the feature maps extracted by the FPN increases with the number of spatial resolutions. This is all the more problematic as these feature pyramids are then processed sequentially by a recurrent net. In practice, the experiments of the next section show that this architecture is the slowest of all methods we evaluate.

- **3D-Unet**[135]. A U-Net in which the convolutions of the encoding branch are three-dimensional to handle simultaneously the spatial and temporal dimensions. For this network, we use the official PyTorch implementation* of Rustowicz *et al.*[135]. This network is composed of five successive 3D-convolution blocks with spatial down-sampling after the second and fourth blocks. Each convolutional block doubles the number of channels of the processed feature maps, and the innermost feature maps have a channel dimension of 128. Skip connections are also implemented with 3D-convolutions, ensuring that temporal encoding is performed at different spatial resolutions. Yet, as seen in Section 1.4, temporal convolutions are not as well suited as self-attention for unevenly sampled satellite image time series. This architecture uses Leaky ReLu, and 3D Batch Normalisations are used across its convolutional blocks. The sequence of feature maps is averaged along the temporal dimension to produce the final embedding of the image sequence. In their implementation, the authors used a linear layer to collapse the temporal dimension, yet this was not a valid option for our dataset: the sequences have highly variable lengths (see the next subsection) and the sequence indices do not correspond to the same acquisition date from one sequence to another.

*https://github.com/roserustowicz/crop-type-mapping

Table 2.2: Review of recent methods. Summary of our analysis of existing approaches for satellite image time series encoding, regarding the spatial scale at which spatial and temporal encoding operates.

	Spatial encoding	**Temporal encoding**
ConvLSTM [131,142]	Single scale	For every pixel
U-Net + ConvLSTM [98,135]	Multi-scale	Only at coarsest spatial level
FPN-ConvLSTM [98]	Multi-scale	Multiple spatial resolutions Costly
3D U-Net [135]	Multi-scale	Multiple spatial resolutions convolutions<attention
Ours	Multi-scale	Multiple spatial resolutions Attention-based encoding

2.2.3 PASTIS DATASET

The PASTIS dataset is designed for the evaluation of semantic and panoptic segmentation of agricultural parcels from SITS. We made it publicly available at `github.com/VSainteuf/pastis-benchmark`.

Overview. The dataset is composed of 2433 square 128×128 patches with 10 spectral bands and at 10m resolution, obtained from the open-access Sentinel-2 platform [†]. For each patch, we stack all available acquisitions between September 2018 and November 2019, forming our four dimensional multi-spectral SITS: $T \times C \times H \times W$.

The publicly available French Land Parcel Identification System (LPIS) allows us to retrieve the extent and crop type of all parcels within the patches, as reported by the farmers. Each patch pixel is annotated with a semantic label corresponding to either the parcels' crop type or the background class. The pixels of each unique parcel in the patch receive a corresponding instance label. The French Payment Agency estimates the accuracy of the LPIS annotations as over 98% regarding crop types. While

[†] `https://scihub.copernicus.eu`

there are no official quantitative assessments regarding parcel surfaces, we performed an extensive visual inspection and failed to observe delineation errors.

Dataset Extent. The SITS of PASTIS are taken from 4 different Sentinel-2 tiles in different regions of the French metropolitan territory as depicted in Figure 2.3a. These regions cover a wide variety of climates and culture distributions. Sentinel tiles span 100×100km and have a spatial resolution of 10 meter per pixel. Each pixel is characterised by 13 spectral bands. We select all bands except the atmospheric bands B01, B09, and B10. Each of these tiles is subdivided in square patches of size 1.28×1.28km (128×128 pixels at 10m/pixel), for a total of around $24,000$ patches. We then select $2,433$ patches (10% of all available patches, see Figure 2.3b), favoring patches with rare crop types to decrease the otherwise extreme class imbalance of the dataset.

Satellite Imagery. We use the L2A Sentinel-2 imagery prepared by THEIA. All bands are spatially resampled to a 10m/pixel resolution with bilinear interpolation.

Nomenclature. The French LPIS uses a 73 class breakdown for crop types. We select classes with at least 400 parcels and with samples in at least 2 of the 4 Sentinel-2 tiles. This leads us to adopt a 18 classes nomenclature, presented in Figure 2.4. Parcels belonging to classes not in our 18-classes nomenclature are annotated with the *void* label.

Cross-Validation. The $2,433$ selected patches are randomly subdivided into 5 splits, allowing us to perform cross-validation. The official 5-fold cross-validation scheme used for benchmarking is given in Table 2.3. To avoid heterogeneous folds, each fold is constituted of patches taken from all four Sentinel tiles. We also chose folds with comparable class distributions, as measured by their pairwise Kullback-Leiber divergence. We show the resulting class distribution for each fold in Figure 2.5. Finally, we prevent adjacent patches from being in different folds to avoid data contamination. Geo-

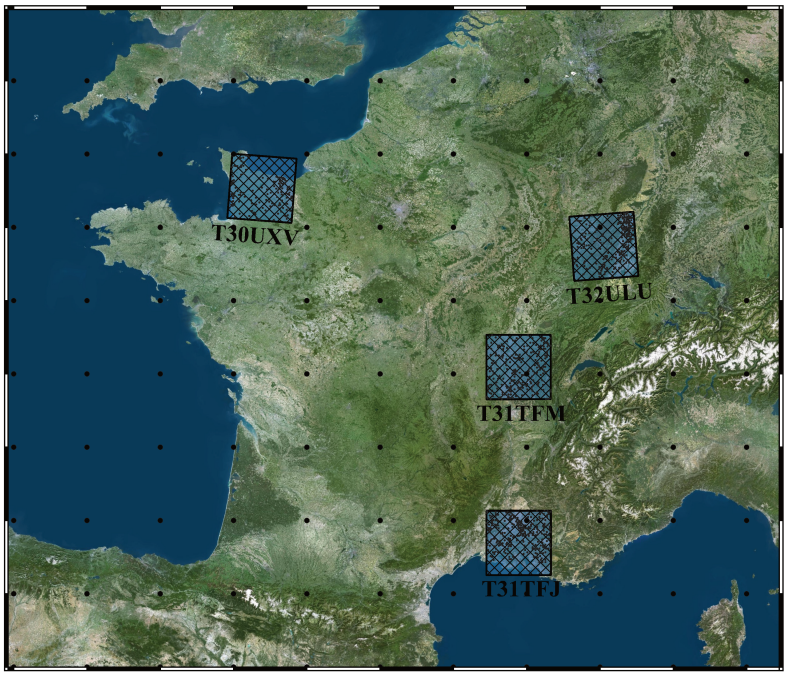
referencing metadata of the patches and parcels is included in PASTIS, allowing for the constitution of geographically consistent folds to evaluate spatial generalisation.

Table 2.3: Cross validation. Official 5-fold cross validation scheme. Each line gives the repartition of the splits into train, validation and test set for each fold.

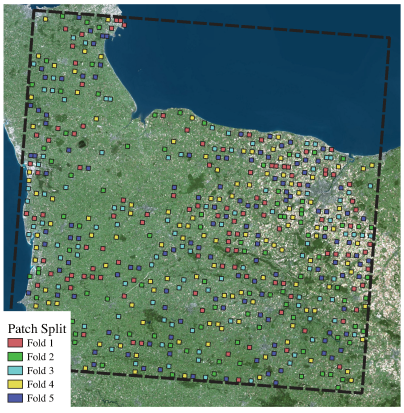
Fold	Train	Val	Test
I	1-2-3	4	5
II	2-3-4	5	1
III	3-4-5	1	2
IV	4-5-1	2	3
V	5-1-2	3	4

Temporal Sampling. The temporal sampling of the sequences in PASTIS is irregular: depending on their location, patches are observed a different number of times and at different intervals. This is a result of both the orbit schedule of Sentinel-2 and the policy of Sentinel data providers not to process tile observations identified as covered by clouds for more than 90% of the tile's surface. As this corresponds to the *real world* setting, we decided to leave the SITS as is, and thus to encourage methods that can favourably address this technical challenge. As a result, the proposed SITS are constituted of 33 to 61 acquisitions.

Cloud Cover. Even after the automatic filtering of predominantly cloudy acquisitions, some patches are still partially or completely obstructed by cloud cover. We opt to not apply further pre-processing or cloud detection, and produce the raw data in PASTIS. Our reasoning is that an adequate algorithm should be able to learn to deal with such acquisitions. Indeed, robustness to cloud-cover has been experimentally demonstrated for deep learning methods by Rußwurm and Körner[131,133].



(a) Location of the four tiles.



(b) Selected patches.

(c) Single patch.

Figure 2.3: Data Location. Spatial distribution of the four Sentinel tiles used in PASTIS (a), and of the selected patches of tile T30UXV (b). We show an example of patch in (c), and highlight with red circles examples of parcels that are mostly outside of the patch's extent and thus annotated with the void label. The green circle highlight a parcel partially cut off by the patch borders, but with sufficient overlap to be kept as a valid parcel.

Label and Color	Class Name	Number of parcels
0	*Background*	-
1	Meadow	31292
2	Soft winter wheat	8206
3	Corn	13123
4	Winter barley	2766
5	Winter rapeseed	1769
6	Spring barley	908
7	Sunflower	1355
8	Grapevine	10640
9	Beet	871
10	Winter triticale	1208
11	Winter durum wheat	1704
12	Fruits, vegetables, flowers	2619
13	Potatoes	551
14	Leguminous fodder	3174
15	Soybeans	1212
16	Orchard	2998
17	Mixed cereal	848
18	Sorghum	707
19	*Void label*	35924

Figure 2.4: Colormap. Color code of our class nomenclature, and the number of parcels per class.

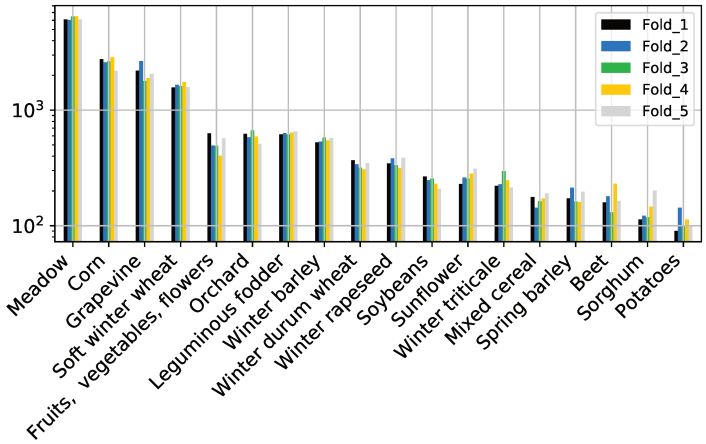


Figure 2.5: Class breakdown. Class distribution for the five folds (in log-scale). The imbalance ratio of PASTIS is ~ 50, an order of magnitude smaller than the dataset used in Chapter 1.

2.2.4 Results

Table 2.4: Semantic Segmentation. We report for our method and six competing methods the model size in trainable parameters, Overall Accuracy (OA), mean Intersection over Union (mIoU), and Inference Time for one fold of $\sim$ 490 sequences (IT). The second part of the table reports the results of our ablation study.

Model	# param $\times 1000$	OA	mIoU	IT (s)
U-TAE (Section 2.1)	1 087	**83.2**	**63.1**	**25.7**
3D-Unet [135]	1 554	81.3	58.4	29.5
U-ConvLSTM [135]	1 508	82.1	57.8	28.3
FPN-ConvLSTM [98]	1 261	81.6	57.1	103.6
U-BiConvLSTM [98]	1 434	81.8	55.9	32.7
ConvGRU [9]	1 040	79.8	54.2	49.0
ConvLSTM [131,142]	1 010	77.9	49.1	49.1
Mean Attention	1 087	82.8	60.1	24.8
Skip Mean + Conv	1 087	82.4	58.9	24.5
Skip Mean	1 074	82.0	58.3	24.5
BatchNorm	1 087	71.9	36.0	22.3
Single Date (August)	1 004	65.6	28.3	1.3
Single Date (May)	1 004	58.1	20.6	1.3

Comparison with the state of the art. In Table 2.4, we detail the performance obtained with 5-fold cross validation of our approach and the six reimplemented baselines. We report the Overall Accuracy (OA) as the ratio between correct and total predictions, and (mIoU) the class-averaged classification IoU. We observe that the convolutional-recurrent methods *ConvGRU* and *ConvLSTM* perform worse. Recurrent networks embedded in an U-Net or a FPN share similar performance, with a much longer inference time for FPN. Our approach significantly outperforms all other methods in terms of precision.

In Figure 2.6, we present the confusion matrix of U-TAE. Unsurprisingly, confusions seem to occur between semantically close classes such as different cereal types, or *Sunflower* and *Fruits, Veg-*

etable, Flower. We also note that confusions occur between background pixels and crop types such as *Grapevine* or *Orchard*. In Figure 2.7, we present a qualitative illustration of the semantic segmentation results. In particular, we show how the typical failure cases of each architecture can be related to our analysis of the spatial scales at which encoding is performed (see Table 2.2).

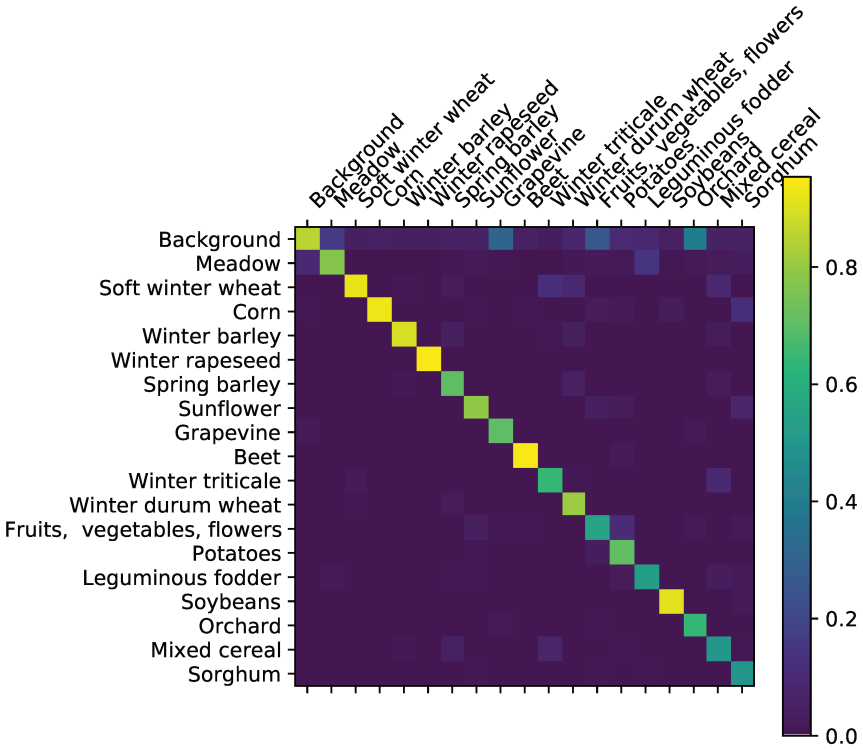
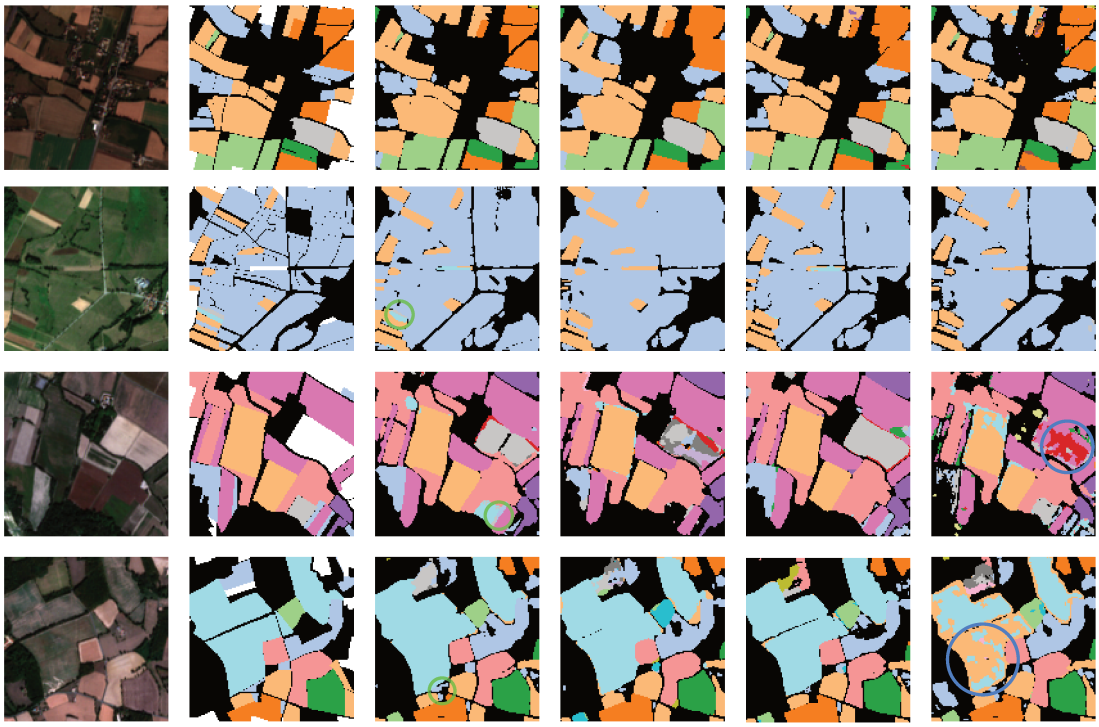


Figure 2.6: Class confusions. Confusion matrix of U-TAE for semantic segmentation on PASTIS. The color of each pixel at line i and column j corresponds to the proportion of samples of the class i that were attributed to the class j.

Ablation study. We first study the impact of using spatially interpolated attention masks to collapse the temporal dimension of the spatio-temporal feature maps at different levels of the encoder simultaneously. Simply computing the temporal average of skip connections for levels without temporal encoding as proposed by Stoian et al. [153], Rustowicz et al. [135], we observe a drop of 4.8 mIoU points

(Skip Mean). This puts our method performance on par with its competing approaches. Adding a 1×1 convolutional layer after the temporal average reduces this drop to 4.2pts (Skip Mean + Conv). Lastly, using interpolated masks but foregoing the channel grouping strategy by averaging the masks group-wise into a single attention mask per level results in a drop of 3.1pts (Mean Attention). This implies that our network is able to use the grouping scheme at different resolutions simultaneously. In conclusion, the main advantage of our proposed attention scheme is that the temporal collapse is controlled at all resolutions, in contrast to recurrent methods. The qualitative results shown in Figure 2.7, suggest that the temporal encoding performed by U-TAE at different spatial resolutions allows it to perform well both on large and small parcels, while other methods typically perform better on one of those two cases.

Using batch normalisation in the encoder leads to a severe degradation of the performance of 27.1pts (BatchNorm). We conclude that the temporal diversity of the acquisitions requires special considerations. This was observed for all U-Net models alike. We also train our model on a single acquisition date (with a classic U-Net and no temporal encoding) for two different cloudless dates in August and May (Single Date). We observe a drop of 24.8 and 42.5pts respectively, highlighting the crucial importance of the temporal dimension of Sentinel-2 for crop classification. We also observed that images with at least partial cloud cover received on average 58% less attention than their cloud-free counterparts. This suggests that our model is able to use the attention module to automatically filter out corrupted data.



(a) Single image. **(b)** Annotation. **(c)** U-TAE. **(d)** 3D-Unet. **(e)** UBiConvLSTM. **(f)** ConvGRU.

Figure 2.7: Qualitative Semantic Segmentation Results. We represent a single image from the sequence using the RGB channels (a), and whose ground truth parcel's limit and crop type are known (b). We then represent the pixelwise prediction from our approach (c), and for three other competing algorithms (d-f). The different predictions shown on this figure illustrate the importance of the resolution at which temporal encoding is performed. ConvGRU applies a recurrent-convolutional network at the highest resolution, which results in predictions with high spatial variability. As a consequence, the prediction over large parcels are inconsistent (blue circles ○). Conversely, U-BiConvLSTM applies temporal encoding to feature maps with a larger receptive field, resulting in more spatially consistent predictions. Yet, this architecture often fails to retrieve small or thin parcels. In contrast, our U-TAE produces spatially consistent predictions on large parcels, while being able to retrieve such small parcels (green circles ○). 3D-Unet also uses temporal encoding at different resolution levels, yet fails to recover these small parcels.

2.3 Panoptic segmentation: Parcels-as-Points (PaPs)

In this section, we build on the U-TAE architecture and complement it with a module that allows for the prediction of the border of each individual parcel as well as its content.

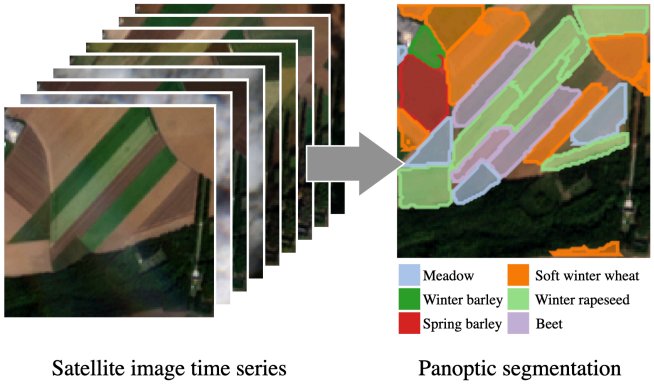


Figure 2.8: Overview. We propose an end-to-end, single-stage model for panoptic segmentation of agricultural parcels from time series of satellite images. Note the difficulty of resolving the parcels' borders from a single image, highlighting the need for modeling temporal dynamics.

2.3.1 Motivation

The task of monitoring both the content and extent of agricultural parcels can be framed as the panoptic segmentation of an image sequence. Panoptic segmentation consists in assigning to each pixel a class and a unique instance label, and has become a standard visual perception task in computer vision[77,107]. However, panoptic segmentation is a fundamentally different task for SITS versus sequences of natural images or videos. Indeed, understanding videos requires tracking objects through time and space[162]. In yearly SITS, the targets are static in a geo-referenced frame, which removes the need for spatial tracking. Additionally, SITS share a common temporal frame of reference, which means that the time of acquisition itself contains information useful for modeling the underlying temporal dynamics. In contrast, the frame number in videos is often arbitrary. Finally, while objects

on the Earth surface generally do not occlude one another, as is commonly the case for objects in natural images, varying cloud cover can make the analysis of SITS arduous. For the specific problem addressed in this section, individualizing agricultural parcels requires learning complex and specific temporal, spatial, and spectral patterns not commonly encountered in video processing, such as differences in plant phenological profiles, subpixel border information, and swift human interventions such as harvests or mowing.

The first step of panoptic segmentation is to delineate all individual instances, *i.e.*, instance segmentation. Most remote sensing instanciation approaches operate on a single acquisition. For example, several methods have been proposed to detect individual instances of trees [122,191], buildings [175], or fields [126]. Plethora of algorithms start with a delineation step (border detection) [37,99,176], and require postprocessing to obtain individual instances. Other methods use segmentation as a preprocessing step and compute cluster-based features [21,32], but do not produce explicit cluster-to-object mappings. Petitjean *et al.* [120] propose a segmentation-aided classification method operating on image time series. However, their approach partitions each image separately and does not attempt to retrieve individual objects consistently across the entire sequence. In this section, we propose the first end-to-end framework for directly performing joint semantic and instance segmentation on SITS. Our approach, dubbed Parcels-as-Points (PaPs), is built upon the efficient CenterMask network [178], which we modify to fit our problem.

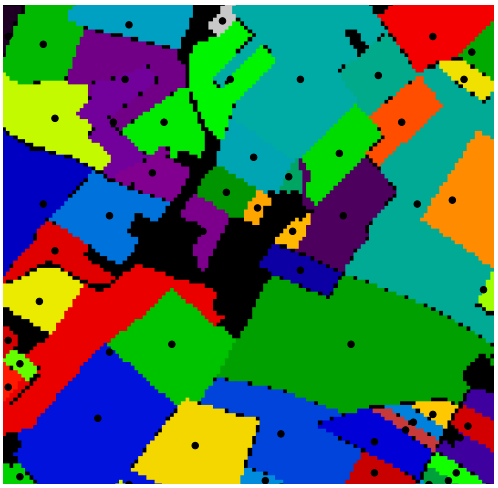
2.3.2 Methods

Our goal is to use the multi-scale feature maps $\{d^l\}_{l=1}^{L}$ learnt by the U-TAE spatio-temporal encoder to perform panoptic segmentation of a sequence of satellite images over an area of interest. The first stage of panoptic segmentation is to produce instance proposals, which are then combined into a single panoptic instance map. Since an entire sequence of images (often over 50) must be encoded to compute $\{d^l\}_{l=1}^{L}$, we favor a simple approach for our panoptic segmentation module. Furthermore, given

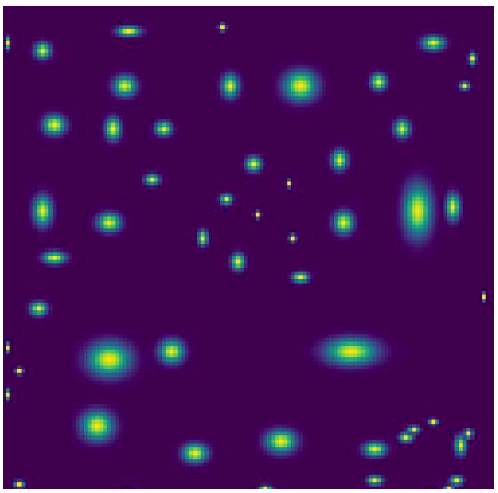
the relative simplicity of parcels' borders, we avoid complex region proposal networks such as Mask-RCNN. Instead, we adapt the single-stage CenterMask instance segmentation network [178], and detail our modifications in the following paragraphs. We name our approach *Parcels-as-Points* (PaPs) to highlight our inspiration from CenterNet/Mask [193,178]. Indeed, the original paper of CenterMask introduces the objects-as-points paradigm, where detection is addressed as centerpoint regression. This contrasts with region proposal approaches [124,56] in which candidate boxes are regressed from anchors in a first stage, and segmentation is performed in a second stage. CenterMask allows to perform detection in a single stage, and we choose to start from this approach to avoid the costly computations of two-stage approaches.

We denote by P the set of ground truth parcels in the image sequence X. Note that the position of these parcels is time-invariant and hence only defined by their spatial extent. Each parcel p is associated with (i) a centerpoint $\hat{\imath}_p, \hat{\jmath}_p$ with integer coordinates, (ii) a bounding box of size $\hat{h}_p, \hat{w}_p$, (iii) a binary instance mask $\hat{s}_p \in \{0,1\}^{H \times W}$, (iv) a class $\hat{k}_p \in [1, K]$ with K the total number of classes.

Centerpoint Detection. Following CenterMask, we perform parcel detection by predicting *centerness heatmaps* supervised by the ground truth parcels' bounding boxes. In the original approach [193], each class has its own heatmap: detection doubles as classification. This is a sensible choice for natural images, since the tasks of detecting an object's nature, location, and shape are intrinsically related. In our setting, however, the parcels' shape and border characteristics are mostly independent of the cultivated crop. For this reason, we use a single centerness heatmap and postpone class identification to a subsequent specialised module. See Figure 2.9 for an illustration of our parcel detection method.

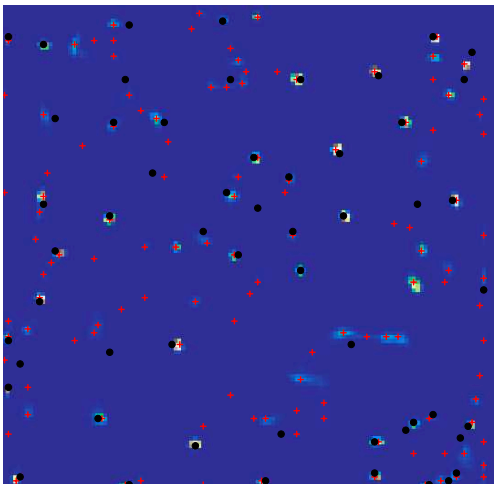


(a) Instance masks



(b) Target heatmap

(c) Sentinel-2 observation.



(d) Predicted centerpoints

Figure 2.9: Centerpoint Detection. The ground truth instance masks (a) is used to construct a target heatmap (b). Our parcel detection module maps the raw sequence of observation (c) to a predicted heatmap (d). The predicted centerpoints (red crosses) are the local maxima of the predicted heatmap (d). The black dots are the true parcels centers.

As in Centermask, we associate each parcel p with a Gaussian kernel of deviations σ_p^{ver} and σ_p^{hor} taken respectively as $1/20$ of the height and width of the parcels' bounding box (Figure 2.9b). Yet, we use heteroschedastic kernels to reflect the potential narrowness of parcels. We then define the target centerness heatmap $\hat{m} \in [0,1]^{H \times W}$ as the maximum value of all parcel kernels at each pixel (i,j) in $H \times W$:

$$\hat{m}_{i,j} = \max_{p \in P} \exp\left(-\left[\frac{(i-\hat{\imath}_p)^2}{2(\sigma_p^{\text{ver}})^2} + \frac{(j-\hat{\jmath}_p)^2}{2(\sigma_p^{\text{hor}})^2}\right]\right) \tag{2.6}$$

A convolutional layer takes the highest-resolution feature map d^1 as input and predicts a centerness heatmap $m \in [0,1]^{H \times W}$ (Figure 2.9d). The predicted heatmap is supervised using a logistic regression loss with a focal factor as defined in (2.7) with $\beta = 4$:

$$\mathcal{L}_{\text{center}} = \frac{-1}{|P|} \sum_{\substack{i=1\cdots H \\ j=1\cdots W}} \begin{cases} \log(m_{i,j}) \text{ if } \hat{m}_{i,j} = 1 \\ (1-\hat{m}_{i,j})^{\beta} \log(1-m_{i,j}) \text{ else.} \end{cases} \tag{2.7}$$

We define the predicted centerpoints as the local maxima of m, *i.e.*, pixels with larger values than their 8 adjacent neighbors. This set can be efficiently computed with a single max-pooling operation. Replacing the max operator by argmax in (2.6) defines a mapping $H \times W \mapsto P$ between pixels and parcels. During training, we associate each true parcel p with the predicted centerpoint $\mathbf{c}(p)$ with highest predicted centerness m among the set of centerpoints which coordinates are mapped to p. If this set is empty, then $\mathbf{c}(p)$ is undefined: the parcel p is not detected. We denote by P' the subset of detected parcels, *i.e.*, for which $\mathbf{c}(p)$ is well defined.

Size and Class Prediction. We associate with a predicted centerpoint c of coordinate (i_c, j_c) the multi-scale feature vector $\tilde{d}_c$ of size $D_1 + \cdots + D_L$ by concatenating channelwise the pixel features at

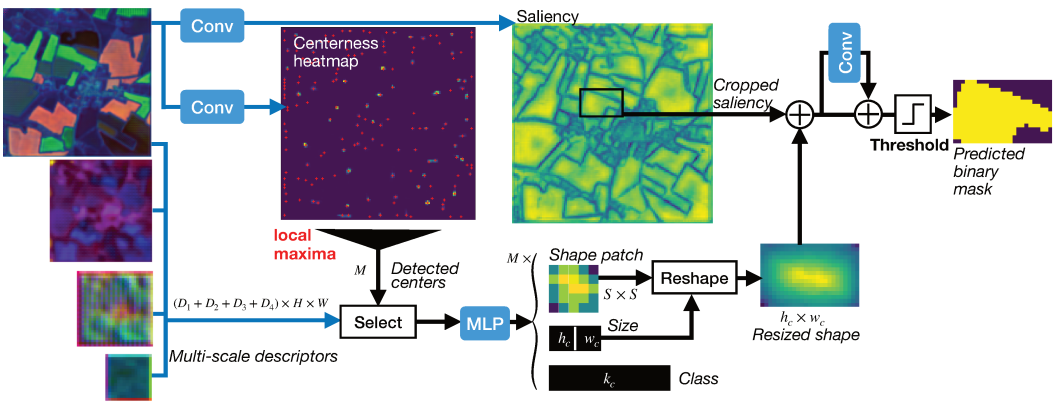


Figure 2.10: PaPs module. The local maxima of the predicted centerness heatmap defines M tentative parcels. For each one, the pixel features at all levels are concatenated and used to predict a bounding box size, a semantic class, and an $S \times S$ shape patch. The latter is combined with a global saliency map for predicting pixel-precise masks. The instance predictions are combined into a panoptic segmentation using the centerness as quality.

location (i_c, j_c) in all maps d^l:

$$\tilde{d}_c = \left[d^l \left(\left\lfloor i_c / 2^{l-1} \right\rfloor, \left\lfloor j_c / 2^{l-1} \right\rfloor \right) \right]_{l=1}^{L}, \tag{2.8}$$

with $[\cdot]$ the channelwise concatenation. This vector $\tilde{d}_c$ is then processed by four different multilayer perceptrons (MLP) to obtain three vectors of sizes 2, K, and S^2 representing respectively: (i) a bounding box size h_c, w_c, (ii) a vector of class probabilities k_c of size K, and (iii) a shape patch s_c of fixed size $S \times S$. The latter is described in the next paragraph.

The class prediction $k_{\mathbf{c}(p)}$ associated to the true parcel p is supervised with the cross-entropy loss, and the size prediction with a normalised L1 loss. For all p in P', we have:

$$\mathcal{L}^p_{\text{class}} = -\log(k_{\mathbf{c}(p)}[\hat{k}_p]) \tag{2.9}$$

$$\mathcal{L}^p_{\text{size}} = \frac{|h_{\mathbf{c}(p)} - \hat{h}_p|}{\hat{h}_p} + \frac{|w_{\mathbf{c}(p)} - \hat{w}_p|}{\hat{w}_p}. \tag{2.10}$$

Shape Prediction. The idea of this step is to combine for a predicted centerpoint c a rough shape patch s_c with a full-resolution global saliency map z to obtain a pixel-precise instance mask, see Figure 2.10. For a centerpoint c of coordinates (i_c, j_c), the predicted shape patch s_c of size $S \times S$ is resized to the predicted size $\lceil h_c \rceil \times \lceil w_c \rceil$ with bilinear interpolation. A convolutional layer maps the outermost feature map d^1 to a saliency map z of size $H \times W$, which is shared by all predicted parcels. This saliency map is then cropped along the predicted bounding box $(i_c, j_c, \lceil h_c \rceil, \lceil w_c \rceil)$. The resized shape and the cropped saliency are added (2.11) to obtain a first local shape $\tilde{l}_c$, which is then further refined with a residual convolutional network CNN (2.12). We denote the resulting predicted shape by l_c:

$$\tilde{l}_c = \text{resize}_c(s_c) + \text{crop}_c(z) \tag{2.11}$$

$$l_c = \text{sigmoid}(\tilde{l}_c + \text{CNN}(\tilde{l}_c)) \,, \tag{2.12}$$

with resize_c and crop_c defined by the coordinates (i_c, j_c) and predicted bounding box size $(\lceil h_c \rceil, \lceil w_c \rceil)$.

The shape and saliency predictions are supervised for each parcel p in P' by computing the pixel-wise binary cross-entropy (BCE) between the predicted shape $l_{\mathbf{c}(p)}$ and the corresponding true binary instance mask $\hat{s}_p$ cropped along the predicted bounding box $(i_{\mathbf{c}(p)}, j_{\mathbf{c}(p)}, \lceil h_{\mathbf{c}(p)} \rceil, \lceil w_{\mathbf{c}(p)} \rceil)$:

$$\mathcal{L}^p_{\text{shape}} = \text{BCE}(l_{\mathbf{c}(p)}, \text{crop}_{\mathbf{c}(p)}(\hat{s}_p)) \,. \tag{2.13}$$

For inference, we associate a binary mask with a predicted centerpoint c by thresholding l_c with the value 0.4 as recommended in CenterMask.

Loss Function. These four losses are combined into a single loss with no weight and optimised end-to-end:

$$\mathcal{L} = \mathcal{L}_{\text{center}} + \frac{1}{|P'|} \sum_{p \in P'} \left(\mathcal{L}^p_{\text{class}} + \mathcal{L}^p_{\text{size}} + \mathcal{L}^p_{\text{shape}} \right). \tag{2.14}$$

Differences with CenterMask. Our approach differs from CenterMask in several key ways: (i) We compute a single saliency map and heatmap instead of K different ones. This represents the absence of parcel occlusion and the similarity of their shapes. (ii) Accounting for the lower resolution of satellite images, centerpoints are computed at full resolution to detect potentially small parcels, thus dispensing us from predicting offsets. (iii) The class prediction is handled centerpoint-wise instead of pixel-wise for efficiency. (iv) Only the selected centerpoints predict shape, class, and size vectors, saving computation and memory. (v) We use simple feature concatenation to compute multi-scale descriptors instead of deep layer aggregation [187] or stacked Hourglass-Networks [112]. (vi) A convolutional network learns to combine the saliency and the mask instead of a simple termwise product.

Converting to Panoptic Segmentation. Panoptic segmentation consists in associating to each pixel a semantic label and, for non-background pixels (our only *stuff* class), an instance label [77]. Our predicted binary instance masks can have overlaps, which we resolve by associating to each predicted parcel a quality measure equal to the predicted centerness m at its associated centerpoint. Masks with higher quality overtake the pixels of overlapping masks with lesser predicted quality. If a mask loses more than 50% of its pixels through this process, it is entirely removed from the predicted instances. Predicted parcels with a quality under a given threshold are dropped. This threshold can be tuned on a validation set to maximize the parcel detection F-score. All pixels not associated with a parcel mask are labelled as background.

Table 2.5: PaPs hyperparameters. Configuration of the three MLPs of PaPs

MLP	Layers	Final Layer
Shape	$256 \mapsto 128 \mapsto S^2$	-
Size	$256 \mapsto 128 \mapsto 2$	Softplus
Class	$256 \mapsto 128 \mapsto 64 \mapsto K$	Softmax

2.4 Numerical Experiments: panoptic segmentation

2.4.1 Implementation details

We use the same U-TAE configuration as earlier as the encoding backbone and a PaPs module with 190k parameters. We set the shape patch size S to 16. The saliency and heatmap predictions are obtained with two separate convolutional blocks operating on the high resolution feature map d_1 with 32 channels. These blocks are composed of two convolutional layers of width 32 and 1 respectively. We use Batch Normalisation and ReLu after the first convolution, and a sigmoid after the second.

The 256-dimensional multi-scale feature vector $(128 + 64 + 32 + 32)$ is mapped to the shape, class and size predictions by three different MLPs described in Table 2.5. The inner layers use Batch Normalisation and ReLu activation.

The residual CNN used for shape refinement is composed of three convolutional layers with kernel size: $1 \mapsto 16 \mapsto 16 \mapsto 1$, with ReLu activation, and instance normalisation on the first layer only.

Across our experiments, we use Adam[76] optimizer and a batch size of 4 sequences. We start with a learning rate of 0.01 for 50 epochs, and decrease it to 0.001 for the last 50 epochs.

A Pytorch implementation is available at `https://github.com/VSainteuf/utae-paps`.

2.4.2 Dataset

We use the satellite imagery of the PASTIS dataset and instance annotations to test the performance of our panoptic segmentation approach. We also use the same 5-fold cross-validation scheme as for semantic segmantation.

Patch Boundaries. The French LPIS allows us to retrieve the pixel-precise borders of each parcel. We also compute bounding boxes for each parcel. The parcels' extents are cropped along the extent of their 128×128 patch, and the bounding boxes are modified accordingly. Parcels whose surface is more than 50% outside of the patch are annotated with the *void* label, see Figure 2.3c.

Small parcels. To avoid degenerate cases where the size of the parcel is too small compared to the resolution of Sentinel-2, we chose to remove some agricultural parcels from the dataset based on the following geometrical criteria:

- Parcels that have a surface smaller that 800m^2 (*i.e.*, 8 Sentinel-2 pixels)
- Parcels for which the ratio of the area over the perimeter is smaller than 10 meters.

Such parcels are annotated with the background label.

Void and Background Labels. Pixels which are not within the extent of any declared parcel are annotated with the background "stuff" label, corresponding to all non-agricultural land uses. In the panoptic setting, this label is associated with pixels not within the extent of any predicted parcel. We do not compute the panoptic metrics for the background class, since our focus is on retrieving the parcels' extent rather than an extensive land-cover prediction. In other words, the reported panoptic metrics are the "things" metrics, which already penalize parcels predicted on background pixels by counting them as false positives.

Table 2.6: Panoptic Segmentation Experiment. We report class-averaged panoptic metrics: SQ, RQ, PQ.

	SQ	RQ	PQ
U-TAE + PaPs	**81.3**	**49.2**	**40.4**
U-ConvLSTM + Paps	80.9	40.8	33.4
$S = 24$	81.3	48.5	39.9
$S = 8$	81.0	48.6	39.8
Multiplicative Saliency	74.5	47.2	35.5
Single-image	72.3	16.9	12.4

The void class is reserved for *out-of-scope* parcels, either because their crop type is not in our nomenclature or because their overlap with the selected square patch is too small. We remove these parcels from all semantic or panoptic metrics and losses. Predicted parcels which overlap with an IoU superior to 0.5 with a void parcel are not counted as false positive or true positive, but are simply ignored by the metric, as recommended in Kirillov et al.[77].

2.4.3 Results

In Table 2.6, we report the class-averaged Segmentation Quality (SQ), Recognition Quality (RQ), and Panoptic Quality (PQ)[77]. We observe that while the network is able to correctly detect and classify most parcels, the task remains difficult. In particular, the combination of ambiguous borders and hard-to-classify parcel content makes for a challenging panoptic segmentation problem. We illustrate these difficulties in Figure 2.12, along with qualitative results.

Replacing the temporal encoder by a U-BiConvLSTM as described in Section 2.2 (U-BiConvLSTM+PaPs), we observe a noticeable performance drop of 8.4 RQ, which is consistent with the results of Table 2.4. As expected, our model's performance is not sensitive to changes in the size S of the shape patch ($S = 24; 8$). Indeed, the shape patches only determine the rough outline of parcels, while the pixel-precise instance masks are derived from the saliency map. Performing shape

prediction with a simple element-wise multiplication as in [178] (Multiplicative Saliency) instead of our residual CNN results in a drop of over −6.8 SQ. Using a single image (August) (Figure 2.13) leads to a low panoptic quality. Indeed, identifying crop types and parcel borders from a single image at the resolution of Sentinel-2 is particularly difficult.

Inference on 490 sequences takes 129s: 26s to generate U-TAE embeddings, 1s for the heatmap and saliency, 90s for instance proposals, and 12s to merge them into a panoptic segmentation. Note that the training time is also doubled compared to simple semantic segmentation.

In Figure 2.11, we compare the relative per-class performance of U-TAE and U-TAE+PaPs on semantic and panoptic segmentation. We note that some classes such as *Beet*, *Winter Rapeseed*, or *Corn* are well retrieved in both tasks, while other such as *Sorghum* and *Mixed Cereal* are challenging in both settings. However, other classes highlight the inherent difference of semantic and panoptic segmentation. *Meadow* or *Grapevine*, for instance, are classes for which the semantic segmentation model scores relatively high, while being among the hardest ones to detect in panoptic segmentation. This suggests that for such classes parcel detection and instance mask prediction are especially hard.

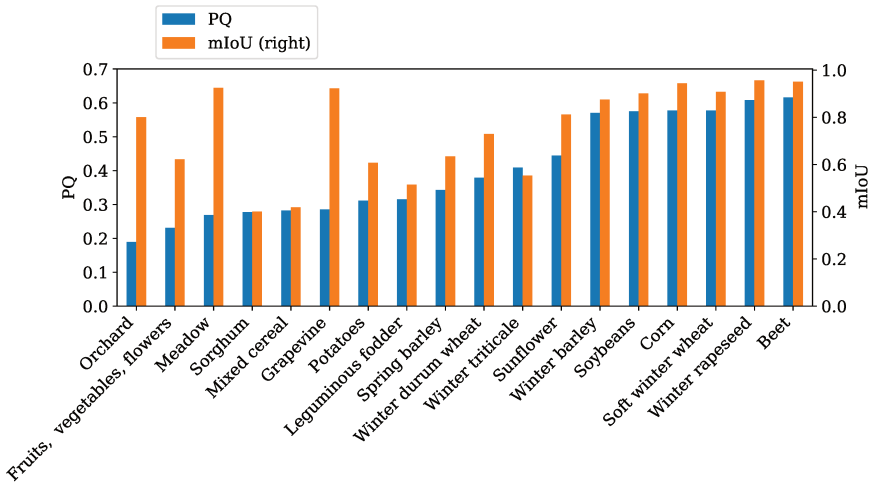


Figure 2.11: Semantic and panoptic per-class performance. Macro-averaged per-class performance of U-TAE+PaPs on panoptic segmentation (left scale), and of U-TAE on semantic segmentation (right scale).

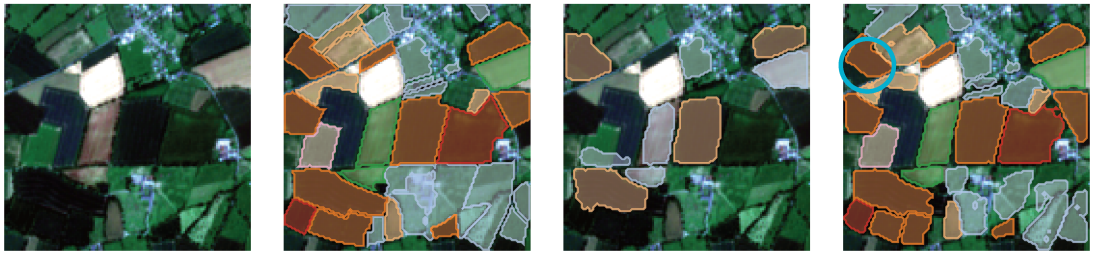
(a) Image from the sequence.

(b) Panoptic annotation.

(c) Panoptic segmentation.

(d) Semantic segmentation.

Figure 2.12: Qualitative Panoptic Segmentation Results. We represent a single image from the sequence using the RGB channels (a), and whose ground truth parcel's limit and types are known (b). We then show the parcels predicted by our panoptic segmentation model (c), and the pixelwise prediction of U-TAE (d). See Figure 2.4 for the color to crop type correspondence. We highlight with a green circle ○ a large, fragmented parcel declared as one single field. This leads to predictions with low confidence and a low panoptic quality. Conversely, the cyan circle ○ highlights such fragmented parcel which is correctly predicted as a single instance. This suggests that our network is able to use the temporal dynamics to recover ambiguous borders. We highlight a failure case with the red circle ○, for which many thin parcels are not properly detected, resulting in a low panoptic quality. We observe that the semantic segmentation model struggles as well for such thin parcels. Finally, we highlight with a blue circle ○ an example in which the panoptic prediction is superior to the semantic segmentation, indicating that detecting parcels' boundaries and extent can be informative for their classification.



(a) Single observation. **(b)** Panoptic annotation. **(c)** Mono-temporal prediction. **(d)** Multi-temporal prediction.

Figure 2.13: Mono-temporal Panoptic Segmentation. We train our mono-temporal model on a single Sentinel-2 observation in August (a), with panoptic annotation (b). We then compare the results of the mono-temporal model in (c) with the results our full model when performing inference on the full length sequence (d) from which the single patch (a) is drawn. First, we observe that many parcels are not detected by the mono-temporal model, indicating an overall low predicted quality. Second, we can see that most detected parcels are misclassified by the mono-temporal model. This is in accordance with the low semantic segmentation score of the mono-temporal model: crop types are hard to distinguish from a single observation. Last, adjacent parcels with no clear borders are predicted as a single parcel, when the multi-temporal model is able to differentiate between the two parcels (cyan circle ○). This illustrates how using SITS instead of single images can help resolve ambiguous parcels delineation.

2.5 Conclusion

We presented two segmentation methods for crop mapping without knowledge on parcel boundaries.

First, we designed a novel spatio-temporal encoder called U-TAE. This architecture builds on the successful results of the L-TAE (Section 1.5) and integrates it in a U-Net structure. By reusing the attention masks at different spatial resolutions, we ensure that temporal encoding is performed at different spatial resolutions, while keeping a reasonable computational load. This architecture can be readily used for semantic segmentation. We introduced PASTIS a large-scale dataset covering 1% of the French territory with semantic and instance annotations. We showed that our U-TAE outperforms existing approaches by a large margin evaluated on PASTIS. Our qualitative analysis showed that our method is able to make spatially consistent predictions both on large and small parcels, as opposed to other methods which seem to perform better at small scale (*e.g.*, ConvLSTM) or large scale (*e.g.*, U-ConvLSTM).

Second, we framed crop type mapping as a panoptic segmentation problem. In this setting, the aim is to recover the boundaries of each individual parcel as well as its crop type. Recovering both the extent and content of parcels is crucial for downstream applications such as subsidy allocation. Yet, we found no existing work to do this from satellite image time series. Our analysis of the differences between SITS and videos motivated us to design a dedicated instance segmentation module instead of applying an off-the-shelf solution from the Computer Vision literature on video panoptic segmentation. To this aim, we introduced PaPs, adapted from CenterMask [178]. Combined with our U-TAE encoder, PaPs set the first state-of-the-art for panoptic segmentation from SITS. We also identified several challenges of this task such as the detection of small parcels, or ambiguities in the way parcels are grouped in the annotations. We hope that these qualitative and quantitative results, as well as our public benchmark dataset will foster further explorations on panoptic segmentation from SITS.

Fusion food as a concept is kind of trying to quite consciously fuse things that are sometimes quite contradictory, sometimes quite far apart, to see if they'd work.

Yottam Ottolenghi

3

Leveraging multiple modalities

In this chapter, we explore the opportunity of leveraging multiple modalities to improve crop type mapping performance. Specifically, we focus on the joint use of the optical imagery of Sentinel-2 with the radar acquisitions of Sentinel-1.

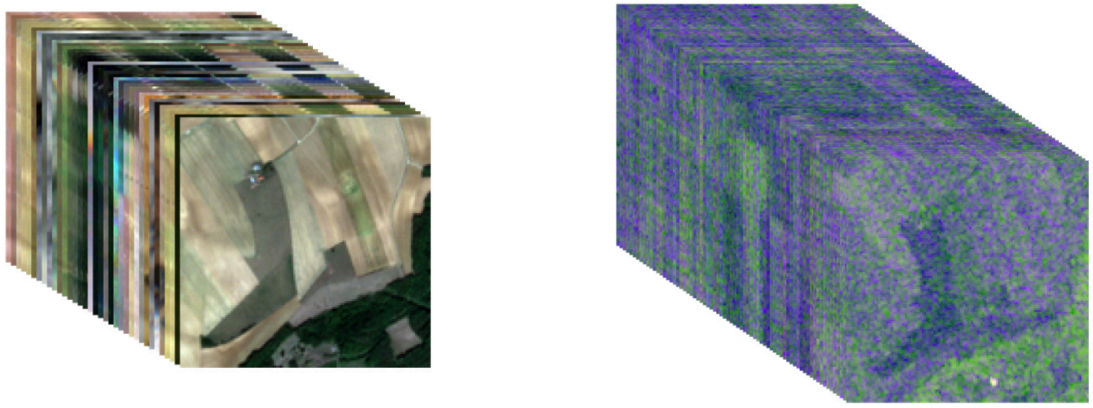


Figure 3.1: Multimodal time series. Optical (left) and radar (right) time series.

3.1 Multitemporal fusion

3.1.1 Motivation

C-band radar and optical images possess well-known synergies for automated crop mapping [170,152,20]. More specifically, multispectral time series contain highly relevant information for monitoring the evolution of plant phenology [173,140]. For example, the study of red and infrared reflectances helps monitoring photosynthetic activity [167]. However, passive optical sensors are highly susceptible to cloud cover and atmospheric distortion [154]. Conversely, due to the influence of extrinsic factors such as humidity and terrain, it is harder to extract discriminative information from radar images for crop mapping. On the other hand, the high revisit frequency and imperviousness to cloud cover makes them uniquely well-suited for monitoring the rapid-changing biological processes of agricultural parcels [100].

In the context of crop type mapping, the fusion of optical and radar time series has been extensively explored with traditional machine learning methods [170,152,58,20,117], and more recently recurrent neural networks [64]. However, despite the significant performance gain offered by methods based on temporal attention [133,44,80,40], these approaches are so far restricted to the analysis of *optical* Satellite

Image Time Series (SITS). In this chapter, we propose to explore different strategies for combining SITS from multiple modalities in temporal attention models, with a focus on crop mapping and the Sentinel-1 and 2 satellites. We implement several fusion schemes commonly encountered in the literature and propose a novel strategy. We present simple enhancements such as auxiliary supervision and temporal dropout to improve performance.

In the context of crop type mapping, the fusion of optical and radar time series has been extensively explored with traditional machine learning methods [170,152,58,20,117,48], and more recently recurrent neural networks [64]. However, despite the significant performance gain offered by methods based on temporal attention [133,44,80,40], these approaches are mostly restricted to the analysis of optical Satellite Image Time Series (SITS). Recently, Ofori-Ampofo et al. [115] proposed a first exploration of the benefit of fusion strategies for parcel-based crop type classification from Sentinel-1 and Sentinel-2 time series with attention-based methods. In this chapter, we extend their analysis to the broader set of crop mapping tasks introduced in 1 and 2: parcel classification, semantic segmentation, and panoptic segmentation. We also study the performance benefit of standard enhancements such as auxiliary supervision and temporal dropout.

To train and evaluate our models, we augment our PASTIS dataset (Section 2.2.3) with corresponding Sentinel-1 radar acquisitions for each of the 2 433 time series for a total of 339 174 radar images. We demonstrate that the right choice of fusion scheme can lead to improvement across the board for all tasks, as well as an increased robustness to varying cloud cover.

The main contributions of this chapter are as follows:

- We present an exhaustive reformulation of fusion strategies in the context of temporal attention-based SITS encoders, as well as common model enhancements.
- We present PASTIS-R, the first large-scale, multimodal, open-access SITS dataset with panoptic annotations.

- We evaluate all fusion schemes and their enhancements on parcel classification, and evaluate the best approaches for segmentation and panoptic segmentation, defining a new state-of-the-art for all tasks.
- We show that combining optical and radar imagery grants significant improvement in terms of robustness to varying cloud cover across all tasks.

3.1.2 Related work

In the following paragraphs, we review the recent literature on fusion approaches for multitemporal fusion of SITS. In particular, we outline the different fusion strategies that are commonly implemented.

Traditional Approaches for Multi-modal SITS Multiple traditional machine learning approaches such as random forest or support vector machines have been adapted to handle information from optical and radar images. As highlighted by the review of Joshi et al. [71], the joint processing of both modalities can mitigate the sensitivity of optical images to cloud cover. Most methods use an early fusion scheme in which the radar and optical features are stacked before being processed by the model [170,102]. This approach can be further improved by selecting the most relevant acquisitions [152] or features [20,48]. Orynbaikyzy et al. [117] compare this feature concatenation approach with a decision fusion approach in which two separate random forest classifiers predict posterior probabilities over classes, and the most confident prediction is retained as the final classification. Their results show that decision fusion performs slightly worse than early feature concatenation.

Deep learning for Multi-Modal SITS The first multimodal deep learning models advocated for an *early fusion* scheme: the channels of all acquisitions from optical and radar time series are concatenated to form a single image with both multimodal and multitemporal pixel features. The re-

sulting images are then processed pixelwise[159] or with convolutional networks[85]. In contrast, Ienco et al.[64] propose to encode each radar and an optical time series separately using a combination of dedicated convolutional and recurrent-convolutional networks. In a *late-fusion* fashion, all resulting embeddings are concatenated channelwise and classified pixelwise by a Multi-Layer Perceptron (MLP). They observe that, as long as each branch is also supervised separately with auxiliary loss terms, this fusion scheme outperforms early fusion. More recently, Ofori-Ampofo et al.[115] studied four fusion strategies for parcel-based classification with a PSE-TAE architecture[44]. Early fusion yields the best improvement on their dataset of Sentinel-2 time series and Sentinel-1 observations in descending orbit. We extend their analysis by evaluating the impact of multimodality for different tasks, evaluate the effects of typical enhancements such as auxiliary classifiers, and use both Sentinel-1 orbits in our analysis.

Other Fusion settings In a different setting, Benedetti et al.[11] use a late fusion approach to combine mono-temporal high spatial resolution images with low spatial resolution time series, and Tom et al.[164] exploit three different mono-temporal modalities for lake ice monitoring by training three encoders to map the different acquisitions to a common feature space. Liu et al.[94] explore multimodal change detection on mono-temporal pairs. They propose to train two encoders in an unsupervised fashion to map simultaneously-acquired images of different modalities to a common feature space. More broadly, the synergy between radar and optical SITS has motivated other exciting applications such as the regression of optical signals from radar images[38,101,58].

Radar processing Data analysis from Synthetic-Aperture Radar (SAR) relies on either extracting backscattering coefficients, interferometric, or polarimetric features from a measured radar signal[125]. Backscattering coefficients are most commonly used for crop type mapping applications[116]. These approaches derive information on the observed surface's geometric properties and dielectric

constant from the amplitude of the complex SAR signal, and discard the phase information. In contrast, interferometric SAR measure phase shift to detect potentially small deformations between two acquisitions. Interferometric features are traditionally used in geodesy [145] and surface [108,157] or structural [165,158,161] monitoring, but also proved discriminative for crop type mapping. Indeed, coeherence estimation in interferometry can help detecting mowing, harvesting, and seeding events [156,103,141], as well as providing information on crop height and density [149]. Lastly, polarimetric SAR data analysis relies on target decomposition of polarimetric information [27,184] to provide additional terrain information, and can be used for canopy structure estimation [148], topography [139], or land cover estimation [168,82]. However, such approaches require full polaristion radar images, *i.e.*, acquired with a sensor emitting radar waves along both polarisation directions. In this chapter, we focus on crop type mapping from data of the open acces Sentinel-1 sensor which does not allow such full polarimetric analyses. Furthermore, to limit the complexity of our experiments and avoid downloading very large Single Look Complex datasets, we focus on SAR backscattering coefficients and leave the extension to interferometric features to further work.

3.1.3 Methods

We consider a set of M image time series $\{X^m\}_{m=1}^{M}$ corresponding to M distinct modalities for a single geo-referenced patch containing one or several agricultural parcels. For simplicity's sake, we assume that all modalities are resampled to the same spatial resolution. Each time sequence X^m can be expressed as a tensor of size $T^m \times C^m \times H \times W$ with T^m the number of available temporal acquisitions for modality m, C^m the number of channels for each pixel for the modality m, and $H \times W$ the spatial extent of the patch.

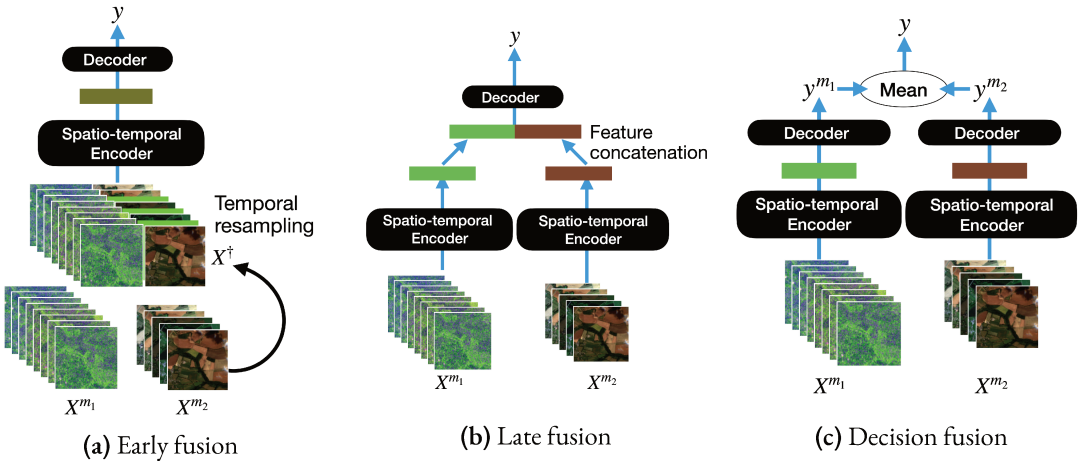


Figure 3.2: Fusion Schemes. We represent the three fusion strategies commonly found in the recent literature. (a) the raw features are interpolated and concatenated into a single sequence. (b) the learned spatio-temporal features of each modality are concatenated prior to classification. (c) each modality is processed independently and the resulting decision averaged.

3.1.3.1 Fusion Strategies

The methods reviewed previously can be categorised into three main strategies: *early*, *late*, and *decision* fusion, all represented in Figure 3.2. We also present *mid*-fusion, a novel fusion scheme specifically adapted for multimodal time sequences. Certain terms—such as "features"—have seen their accepted meaning evolve with the gradual adoption of the deep learning paradigm, leading to ambiguity in terms such as "early" or "late" feature fusion. We propose to redefine the terminology of fusion schemes in an unambiguous manner for the analysis of temporal sequences of images in the following.

Early Fusion. This approach combines the different modalities at the raw feature level. In our context, this amounts to concatenating the modalities *channel-wise* at each observation date. If the different acquisitions are simultaneous, and since the resolutions are identical, this is a straightforward step. However, when the modalities are captured at different times, a preprocessing step is required to

interpolate all modalities to a common temporal sampling. We denote by $T^\dagger$ the number of time steps in the chosen temporal sampling and by $X^\dagger$ the resulting aggregated tensor of size $T^\dagger \times C^\dagger \times H \times W$ with $C^\dagger = \sum_m C^m$ as defined in Equation 3.1.

This interpolation step can be costly in terms of computation and memory. Furthermore, the relevance of temporal interpolation for a fast-changing process such as plant growth and harvesting is questionable. This is only made worse by clouds obstructing the optical modalities. However, an advantage of this approach is the simplicity of encoding $X^\dagger$: a single spatio-temporal encoder $\mathcal{E}_{\text{spatio-temporal}}$ can be used to learn a truly cross-modal representation, and a unique decoder $\mathcal{D}$ produces the final prediction:

$$X^\dagger = \text{merge}^{(C)}\left(\left\{\text{interpolate}(X^m) \text{ to } T^\dagger\right\}_{m=1}^{M}\right) \tag{3.1}$$

$$y^{\text{early}} = \mathcal{D} \circ \mathcal{E}_{\text{spatio-temporal}}(X^\dagger)\,. \tag{3.2}$$

Late Fusion This fusion scheme starts by encoding each modality m separately with dedicated spatio-temporal encoders $\mathcal{E}^m_{\text{spatio-temporal}}$ into embeddings of size F^m. These vectors are then concatenated for all modalities along the channel dimension into a vector of size $\sum_m F^m$, which is ultimately mapped to a prediction y^{late} by a unique decoder $\mathcal{D}$:

$$y^{\text{late}} = \mathcal{D} \circ \text{merge}^{(C)}\left(\left\{\mathcal{E}^m_{\text{spatio-temporal}}(X^m)\right\}_{m=1}^{M}\right), \tag{3.3}$$

with $\text{merge}^{(C)}$ the channelwise concatenation operator. While each latent feature is derived from a single modality, this method allows the decoder to make decisions taking all modalities into account simultaneously.

Decision Fusion This approach ignores the interplay between modalities and makes a prediction for each modality independently. A set of M spatio-temporal encoder $\mathcal{E}^{m}_{\text{spatio-temporal}}$ maps each sequence of size $T^m \times C^m \times H \times W$ to a latent space of size F^m. Then, a set of M decoders $\mathcal{D}^m$ maps each spatio-temporal feature into a prediction. Finally, an aggregation rule is applied to combine all M predictions into a final prediction y^{decision}. Typically, predictions are averaged across all available modalities :

$$y^{\text{decision}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{D}^m \circ \mathcal{E}^{m}_{\text{spatio-temporal}} \left(X^m \right) . \tag{3.4}$$

Mid-Fusion Certain network architectures used to process temporal sequences such as SITS can be broken down into a spatial and a temporal encoder. In such cases, the spatial features can be interwoven, *i.e.*, temporally stacked, into a single multimodal sequence, see Figure 3.3. This approach can be seen as a compromise between early and late fusion and combines three of their advantages: (i) the temporal encoder can leverage all modalities simultaneously, (ii) only one temporal encoder is needed, (iii) no heavy preprocessing is necessary to merge the feature sequences as they are simply stacked.

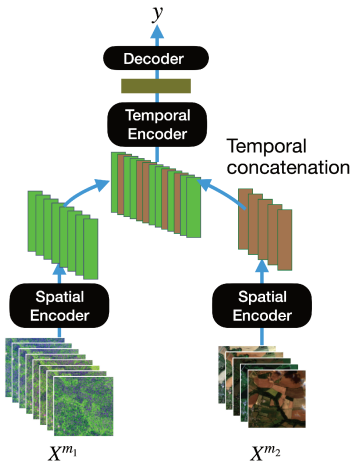


Figure 3.3: Mid-Fusion. Each modality is processed by a dedicated spatial encoder, and the resulting features are stacked into a single sequence of features.

Each modality m has a dedicated spatial encoder $\mathcal{E}^{m}_{\text{spatial}}$ mapping images to a feature vector of size F^m. These vectors are then concatenated chronologically along the temporal dimension into a unique sequence of length $\sum_m T^m$. A unique temporal encoder $\mathcal{E}_{\text{temporal}}$ maps this sequence of features into a unique vector, which is in turn classified by a unique

decoder $\mathcal{D}$:

$$y^{\text{mid}} = \mathcal{D} \circ \mathcal{E}_{\text{temporal}} \circ \text{merge}^{(T)} \left(\left\{ \mathcal{E}^{m}_{\text{spatial}} \left(X^{m} \right) \right\}_{m=0}^{M} \right), \tag{3.5}$$

with $\text{merge}^{(T)}$ the operator concatenating a set of tensors along the temporal dimension.

3.1.3.2 Auxiliary Supervision

We denote by $\text{criterion}(\cdot\,,\,\cdot)$ the function used to compare the prediction y with the target signal $\hat{y}$. This is typically the cross-entropy for parcel or pixel classification, and can be more complex for panoptic or instance segmentation (Section 2.3). The resulting function $\mathcal{L}_{\text{obj}}$ is called the objective loss and supervizes the prediction y of the network to realize the sought task:

$$\mathcal{L}_{\text{obj}} = \text{criterion}(y, \hat{y}) \tag{3.6}$$

A common problem in deep feature fusion is encountered when most (but not all) of the discriminative information is concentrated among a reduced number of modalities. In this case, the other modalities yield predictions and features which are less relevant for the task at hand. Consequently, the final decision taken by the multimodal network focuses on the *better* modalities, and the parts of the network operating on the *lesser* modalities receive a weaker supervisory signal. This results in a network that may not fully leverage the inter-modal patterns that would otherwise allow the multimodal prediction to outperform the *best* modality. This is typically the case for Sentinel SITS, as multispectral optical acquisitions are often more conductive to capture phenological patterns than SAR information. Sentinel-1 signal is indeed affected by local terrain angle[73], humidity[39], and is subject to speckle[3].

To mitigate this issue, we can use auxiliary losses to supervise each modality independently on

top of the objective loss $\mathcal{L}_{\text{obj}}$. This has been shown by Ienco et al.[64] to help combining optical and radar imagery. To this end, we associate a prediction y^m to each modality, which is supervised by the auxiliary loss $\mathcal{L}_{\text{aux}}$:

$$\mathcal{L}_{\text{aux}} = \sum_{m=1}^{M} \lambda^m \text{ criterion}(y^m, \hat{y}) \,, \tag{3.7}$$

with λ_m the strength associated to each modality. Note that, depending on the chosen fusion scheme, computing the single-modality prediction y^m may imply adding new modules to the backbone network. This requires M decoders $\mathcal{D}^m$, in the case of late fusion. For mid-fusion, we must add M temporal encoders $\mathcal{E}^m_{\text{temporal}}$ as well. No additional modules are necessary for decision fusion as single-modality predictions y^m are already necessary to produce the final prediction y. In contrast, auxiliary supervision in the case of early fusion would amount to duplicating the entire network making it both fruitless and costly.

3.1.3.3 Temporal Dropout

To promote a multimodal model that leverages all available modalities, we propose a simple data augmentation strategy dubbed temporal dropout. Inspired by the classical dropout strategy[150], we randomly drop observations from the input sequences. The idea is to prevent the network from over-relying on a single modality since its presence is never assured. Formally, we associate a dropout probability $p^m \in [0, 1]$ for each modality $m \in [1, M]$. During training, each observation of the sequence is dropped with probability p^m. At inference time, the network can use all available observations. Note that this technique can also be used on models operating on a single modality by randomly dropping some acquisitions.

3.2 Numerical Experiments: Fusion

In this section, we evaluate the different fusion strategies integrated in our temporal attention-based models on the three tasks we addressed in the previous chapters.

3.2.1 Implementation details

We use the official 5-fold cross-validation of PASTIS to evaluate the performance of the different models. We use the Adam optimizer[76] with default parameters $\text{lr} = 0.001$, $\beta = (0.9, 0.999)$ unless specified, and train all networks on a TESLA V100 GPU with 32Gb of VRAM.

Multimodality Configuration. We consider the two orbits of Sentinel-1 as separate modalities to account for their difference in incident angle, which corresponds to $M = 3$. When using auxiliary loss terms we set $\lambda^m = 0.5$ for all modalities. When using temporal dropout, we set $p_0 = 0.4$ for the optical modality and $p_1 = p_2 = 0.2$ for the radar time series. For early fusion, we interpolate the Sentinel-1 observations to the dates of the Sentinel-2 time series. Indeed, the opposite interpolation strategy would imply tripling the temporal length of the Sentinel-2 time series, which would significantly increase the memory usage. Interpolation is computed on the fly when loading dataset samples.

Parcel Classification. We first implement the different fusion strategies for parcel-based crop type classification. In this setting, the contour of parcels is known in advance and the task is to classify the cultivated crop in a corresponding yearly SITS. We use Pixel-Set Encoders (PSE) and Lightweight Temporal Attention Encoders (L-TAE) for spatial and temporal encoding. All spatio-temporal encoders $\mathcal{E}_{\text{spatio-temporal}}$ are a combination of a PSE encoding all images of the time series simultaneously and an L-TAE processing the resulting sequence of embeddings. All decoders $\mathcal{D}$ are simple Multi-

Layer Perceptrons (MLP). All models are trained with cross-entropy loss. We use the same hyper-parameter configuration as in Section 1.6. For this problem, we train the models for 100 epochs in batches of 128 parcels. We use the 18 class nomenclature of PASTIS and report the classification Intersection over Union macro-averaged over the class set (mIoU) to evaluate the parcel-level predictions.

Semantic Segmentation. In this setting, we use U-TAE as spatio-temporal encoder with the same parametrisation as in Section 2.2. We use a 2-layer convolutional neural net as decoders $\mathcal{D}$. The models are trained with cross-entropy loss. We train the semantic segmentation models for 100 epochs in batches of 4 temporal patches. In this setting, the models also predict *background* pixels, resulting in a 19 class nomenclature. We report the mIoU of the pixel-level predictions.

Panoptic Segmentation. For this task, we also use U-TAE for spatio-temporal encoding. To output panoptic predictions, we use as decoder the instance segmentation module Parcel-as-Points (PaPs) and its associated loss function for supervision (see Section 2.3). As in Section 2.4, we start with a higher learning rate of 0.01 for 50 epochs, and decrease it to 0.001 for the last 50 epochs. We report the class-averaged panoptic metrics introduced in Kirillov et al.[77]: Segmentation Quality (SQ), Recognition Quality (RQ), and Panoptic Quality (PQ).

3.2.2 Dataset - PASTIS-R

To evaluate the benefit of multimodality, we extend PASTIS dataset with the corresponding Sentinel-1 observations. As seen in Section 2.2, PASTIS is composed of 2433 time series of multi-spectral patches sampled in four different regions of France. Each patch has a spatial extent of 1.28km×1.28km and contains all available Sentinel-2 observations for the 2019 agricultural year for a total of 115k images.

We use Sentinel-1 in Ground Range Detected format processed into σ_0 backscatter coefficient in decibels, orthorectified at a 10m spatial resolution with Orfeo Toolbox[25]. We do not apply any

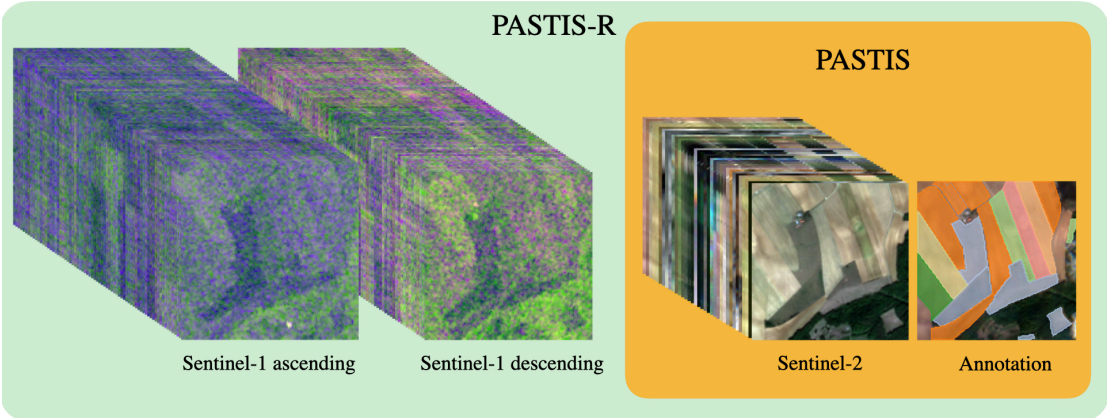


Figure 3.4: Pastis-R. We extend the PASTIS dataset with radar time series corresponding to ascending and descending orbits of Sentinel-1. For each square patch of 1.28km×1.28km, PASTIS-R thus provides the image time series of 3 different modalities, along with semantic and instance annotation for each pixel.

spatial or temporal speckle filtering, nor radiometric terrain correction: following the deep learning paradigm, we limit data preprocessing to the minimum. We assemble each Sentinel-1 observation into a 3-channel image: vertical polarization (VV), horizontal polarisation (VH), and the ratio of vertical over horizontal polarization (VV/VH). We separate observations made in ascending and descending orbit into two distinct time series. Indeed, the incidence angle of space-borne radar can significantly influence the return signal[146]. As represented in Figure 3.4, each time series comprises around 70 radar acquisitions for each of the 2433 patches. This amounts to a total of 339k added radar images. We use the annotations of PASTIS: semantic class and instance identifier for each pixel, allowing us to evaluate models for parcel-based classification, semantic segmentation, and panoptic segmentation. We make the PASTIS-R dataset publicly available at: `github.com/VSainteuf/pastis-benchmark`.

3.2.3 Results

3.2.3.1 Parcel Classification Experiment

We first implement and evaluate the different fusion schemes and enhancements in the case of parcel classification.

Table 3.1: Parcel Classification. We evaluate the performance of models operating on a single modality (top) and of different fusion strategies for parcel-based classification (bottom). For each model, we evaluate its baseline performance and the impact of the temporal dropout and/or auxiliary classifiers enhancements, when applicable. We report the 5-fold cross validated classification scores in terms of mean class-wise Intersect over Union, and the parameter count of the base model, and, when relevant, of the model with auxiliary classifiers.

	Base		Temp. dropout	Auxiliary supervision	Auxiliary & Temp. dropout	Parameter Count
	OA	mIoU	mIoU			
S2	91.7	73.9	74.5	-	-	114k
S1D	87.0	64.5	64.7	-	-	114k
S1A	86.4	63.3	62.9	-	-	114k
Early Fusion	91.8	74.9	76.5	-	-	117k
Mid Fusion	92.0	75.1	75.9	75.0	76.5	152k/185k
Late Fusion	91.1	73.0	73.6	76.1	**77.2**	254k/287k
Decision Fusion	91.0	72.5	72.8	75.2	75.8	259k

Analysis. In Table 3.1, we report the performance of all fusion schemes with and without enhancements. We first observe that the optical satellite S2 outperforms significantly the two radar time series by a margin of almost 10 points of mIoU, confirming the relevance of Sentinel-2 for crop type mapping. We remark that, without enhancement, multimodal models trained with early or mid-fusion schemes improve the performance compared to optical-only networks, while decision and late fusion perform slightly worse. This highlights the benefit of learning to mix modality features early on. In contrast, auxiliary supervision and temporal dropout provide more improvement to the later models. This shows that these enhancements can promote learning to combine efficiently features and

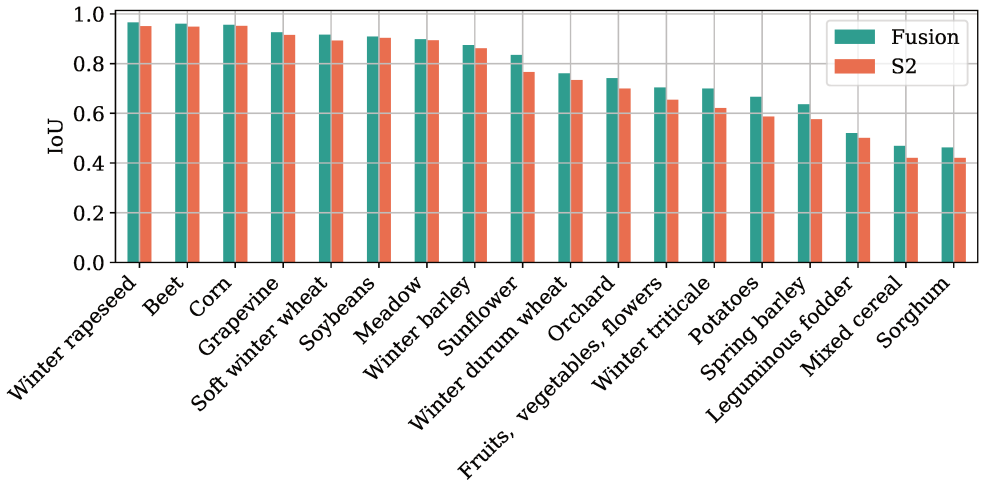


Figure 3.5: Classwise Performance for Parcel Classification. We report the IoU of the late fusion model with auxiliary supervision and temporal dropout and the model trained purely on the optical modality. The benefit brought by multimodality is consistent for all casses, and more notable for harders classes such as *Potatoes*, or *Winter triticale*.

decisions from different modalities, as observed in Ienco et al.[64]. All things considered, late fusion with both enhancements performs best with +3.3pts mIoU compared to a network operating purely on the optical modality, see Figure 3.5 for a classwise comparison. Mid-fusion without enhancement provides a good performance with a lower parameter count and none of the pre-processing necessary for early fusion. In practice, the mid-fusion scheme is 20% faster at inference time than late fusion, making it a valid choice when operating with limited computational resources.

Auxiliary Supervision and Gradient Flow. Motivated by the impact of auxiliary supervision on the performance of the late fusion approach, we propose to study its effect on the learning process further. Specifically, we wish to evaluate the different spatio-temporal encoders' contribution to the reduction of the objective loss $\mathcal{L}_{\text{obj}}$, with and without auxiliary supervision, and for the parcel classification task. Note that, as auxiliary decisions are not computed at inference time, we only consider the decrease of $\mathcal{L}_{\text{obj}}$: a decrease in the auxiliary losses does not directly affect the model's performance.

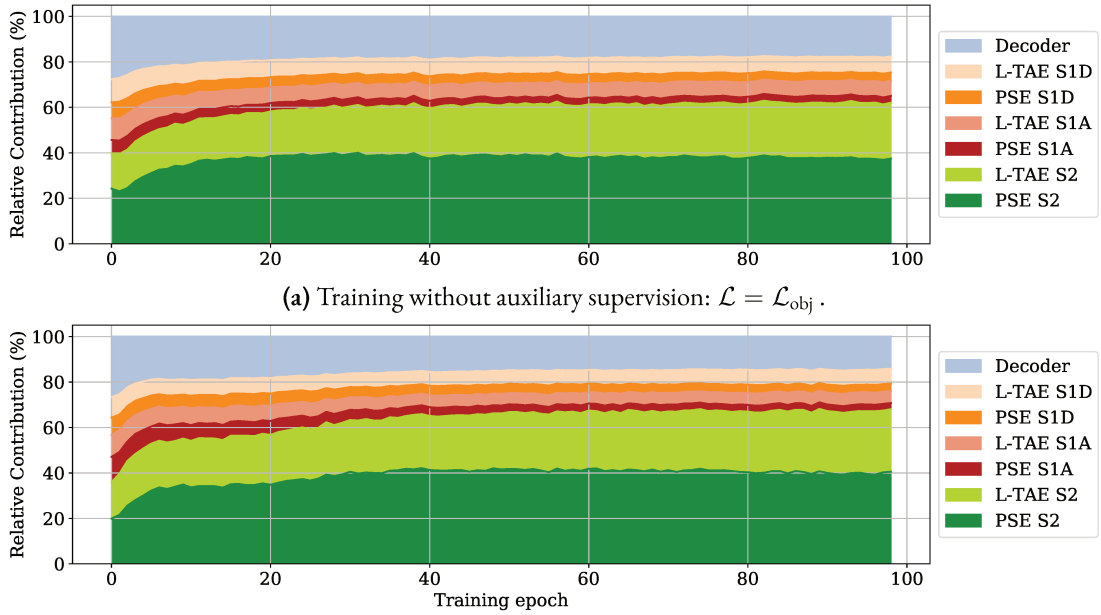
Following the insights of Wang et al. [177], we consider the following first-order approximation of the decrease of $\mathcal{L}_{\text{obj}}$ incurred by taking a gradient step:

$$\Delta\mathcal{L}_{\text{obj}} = \eta\langle\nabla\mathcal{L}, \nabla\mathcal{L}_{\text{obj}}\rangle \,, \tag{3.8}$$

with η the current learning rate.The term $\nabla\mathcal{L}$ of the scalar product in (3.8) corresponds to the step size in the gradient descent and the term $\nabla\mathcal{L}_{\text{obj}}$ to the slope of the objective loss. Their scalar product approximates the decrease in objective loss when taking a single gradient step. Note that this approximation, called gradient flow, is only valid when using stochastic gradient descent (SGD) and does not hold for momentum or adaptive optimization schemes such as ADAM[76]. We thus retrain the late fusion model with SGD for parcel classification. By considering each term in the scalar product in Equation 3.8, we can estimate the contribution of each parameter of the network to the decrease of the objective loss $\mathcal{L}_{\text{obj}}$.

In Figure 3.6, we represent the evolution of the gradient flow for different modules of our architecture by summing the contribution of their corresponding parameters. We observe that, as expected, the gradient flow is concentrated in the modules dedicated to the optical modality. Interestingly, the spatial encoders contribute as much or even more than the temporal encoders despite having four times fewer parameters.

We remark that auxiliary losses lead the model to a different training regime. While auxiliary supervision results in an increase of the proportion of gradient flow in some radar modules such as PSE-S1A, the flow also increases in proportion in some optical modules as well. We conclude that auxiliary supervision affects all modalities, not only the weaker modalities.



(a) Training without auxiliary supervision: $\mathcal{L} = \mathcal{L}_{\text{obj}}$.

(b) Training with auxiliary supervision: $\mathcal{L} = \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{aux}}$.

Figure 3.6: Gradient Flow. Evolution of the gradient flow for different modules of the late fusion model. The contribution of each modality is plotted as a fraction of the total flow, without auxiliary loss terms (top) and with the additional $\mathcal{L}_{\text{aux}}$ term (bottom). We report the flow for the spatial encoders (PSE), temporal encoders (LTAE), and the MLP decoder.

3.2.3.2 Semantic Segmentation

In this section, we evaluate the performance of fusion schemes compared to single modality baselines for semantic segmentation. While the mid-fusion scheme yields promising results on parcel-based experiments, its implementation into a semantic segmentation architecture is not trivial. Indeed, the state-of-the-art network for this task[42] relies a on U-Net architecture with temporal encoding (Section 2.2). In this architecture, spatial and temporal encoding are performed conjointly. After several unsuccessful attempts, we limit our study to the other fusion schemes for this task.

Analysis. We report the performance of the different models in Table 3.2. In our experimental setup, the late fusion model with over $\sim$ 200 total multimodal observations did not fit in the 32Gb

Table 3.2: Semantic Segmentation Experiment. We evaluate the semantic segmentation performance of models operating on a single modality and of multimodal models. For each model, we evaluate its baseline performance and the impact of temporal dropout and/or auxiliary classifiers, when applicable. We report the 5-fold cross validated classification scores in terms of mean classwise Intersect over Union (- not applicable). Note that temporal dropout is necessary for the late and decision fusion models to fit in memory.

	Base	Temporal dropout	Auxiliary & Temporal dropout	Parameter Count
S2	63.1	63.6	-	1 087k
S1D	54.9	54.7	-	1 083k
S1A	53.8	53.3	-	1 083k
Early Fusion	64.9	65.8	-	1 602k
Late Fusion	-	65.8	**66.3**	1 709k
Decision Fusion	-	64.7	64.3	1 742k

of memory of our GPU with a batch size of 4 image time series. By reducing the size of the input sequences, temporal dropout allowed us to train this memory-intensive model. The late fusion model improves the performance of the unimodal models by 2.7 mIoU points. The performance is further improved by another 0.5 point with the addition of auxiliary supervision. The early fusion model performs slightly below late fusion, even with temporal dropout. As represented in Figure 3.7, the radar modality allows for prediction with crisper contours, in particular between adjacent or nearly adjacent parcels. This suggests that the image rugosity of the radar acquisitions is can be valuable to detect inter-parcel zones. These areas, often of sub-pixel extent, may display optical reflectances similar to their neighboring parcels but often present surfaces such as fences or groves with a volumetric scatter and thus a distinct radar response .

Note that the performance of our models on semantic segmentation is around 10pts mIoU below that for parcel classification. This was expected as the semantic segmentation task prevents from exploiting knowledge about the contour of parcels, and adds the *background* class corresponding to non-agricultural land.

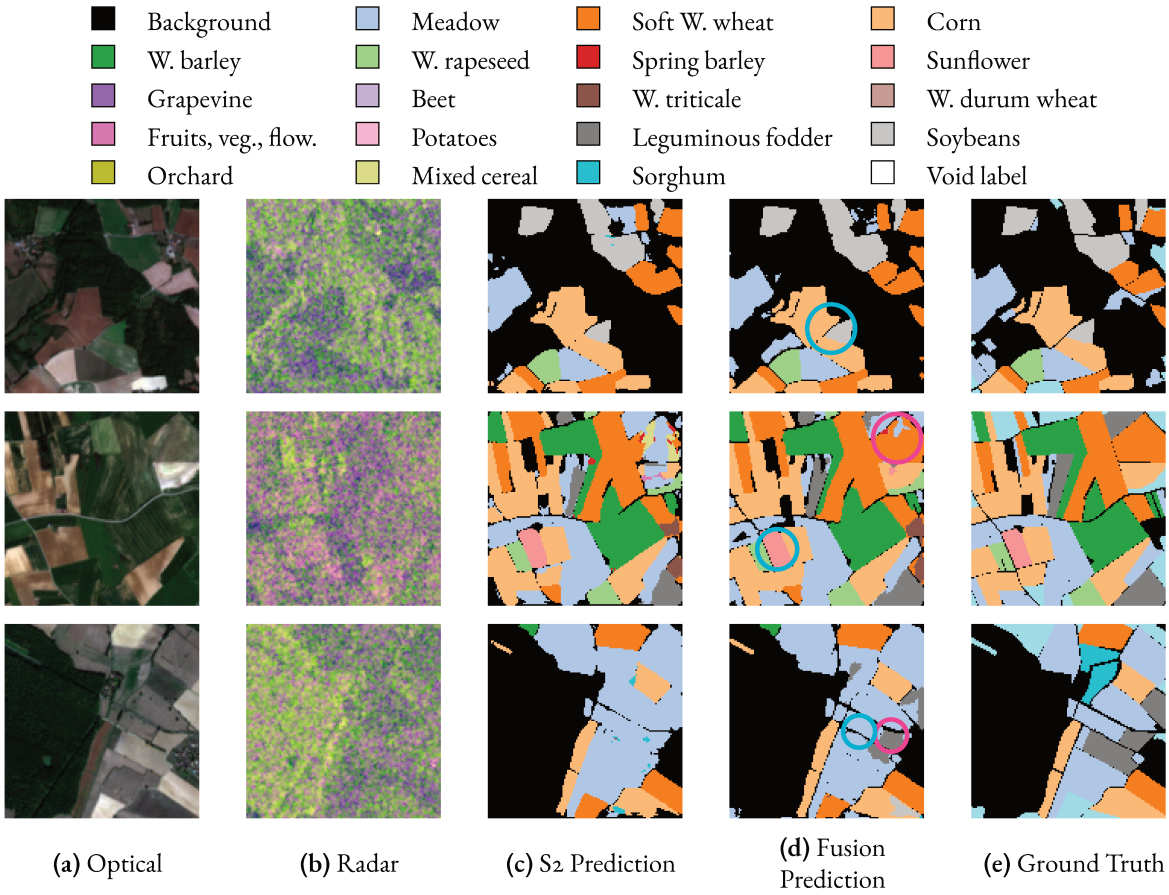
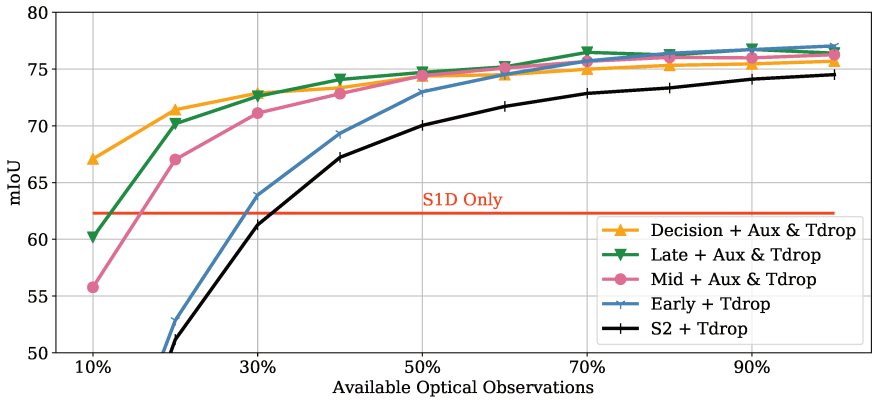


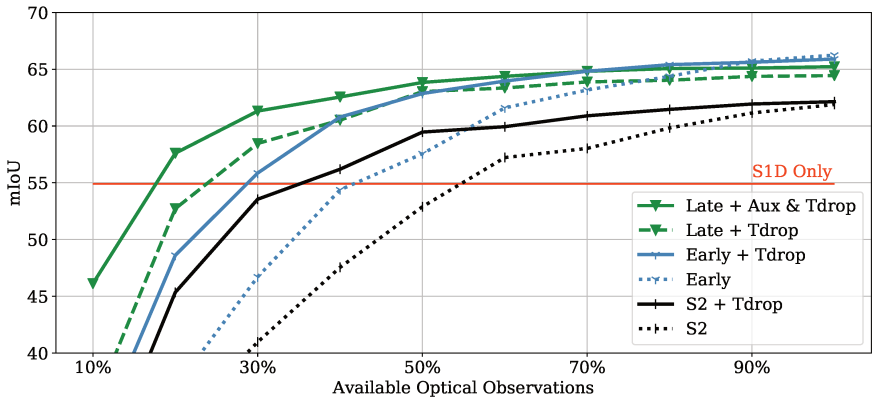
Figure 3.7: Qualitative Results for Semantic Segmentation. We show one observation from the optical time series in (a) and from the radar time series in (b). The prediction for the unimodal optical model is represented in (c) and the multimodal model in (d), and finally the ground truth in (e). We observe that the multimodal model produces results with clearer and more distinct borders between close parcels (cyan circle ○). The multimodal model also displays fewer errors for hard and ambiguous parcels, showing the benefit of learning intermodal features (magenta circle ○). Crop types are represented according to the color code above (W. stands for Winter). The same colormap is used in all subsequent figures representing crop labels.

Varying Cloud Cover Experiment. One of the motivations for using both optical and radar images in the context of crop type mapping is to exploit the imperviousness of radar signals to cloud cover. This potentially allows models to rely on the radar signal when optical observations are obstructed by clouds, which is particularly crucial in countries with pervasive cloud cover such as sub-

tropical regions[117]. The parcel-based and semantic experiments allow for a first exploration of this capacity, but remain bound to the specific cloud conditions of the metropolitan French territory and the year of acquisition (2019). We propose to further investigate the benefit of multimodality by artificially simulating increased cloud obstruction on the test set. To do so, we evaluate the performance of the different models when randomly removing optical acquisitions, while leaving the radar time series unchanged. We report the performance of the models in Figure 3.8, for different ratios of remaining optical observations, corresponding to different levels of cloud obstruction.



(a) Parcel-based classification



(b) Semantic segmentation

Figure 3.8: Varying cloud cover experiment. We evaluate the different models with varying ratios optical observations remaining. In both parcel-based classification (a) and semantic segmentation (b), the fusion models prove more robust to a reduced number of optical observations.

Expectedly, the performance of the S2-only model drops drastically as the number of available optical observations decreases for both parcel classification and semantic segmentation, performing worse than unimodal radar models for a ratio of 70% of artificial occlusion. Multimodal fusion models can maintain an almost constant level of performance for up to 50% missing optical acquisitions. For more extreme ratios, the performances of the multimodal models eventually drop. The magnitude of the drop seems to be related to the amount of interplay between modalities in the network. Early fusion proves the least robust to missing optical observations. Mid-fusion, and to a lesser extent the late fusion are also affected by obstruction. These models rely on multimodal encoders and decoders, which are likely to be affected by a severe decrease in the quality of the optical sequence. In contrast, the decision fusion scheme is composed of independent classifiers and proves to be the most resilient: even with 90% of optical images removed, it still outperforms the radar modality by $\sim$ 5pts mIoU on parcel classification. We conclude that decision fusion should be favored in regions with pervasive or inconsistent cloud cover.

We also observe that auxiliary supervision and temporal dropout contribute to make both unimodal and multimodal models more resilient to missing optical acquisitions for semantic segmentation. The same phenomenon can be observed for parcel classification, but was not represented for the sake of clarity.

3.2.3.3 Panoptic segmentation experiment

In this section, we evaluate the performance of the early and late fusion schemes compared to single modality baselines for panoptic segmentation. We do not evaluate auxiliary losses on the late fusion model as the use of auxiliary decoders in this setting comes at a prohibitive computational cost. Indeed, the auxiliary decoders would be PaPs instance segmentation modules which already significantly impact training times on single modality architectures. Decision fusion is not evaluated here for the same reason. Like in the semantic segmentation experiment, temporal dropout proved necessary to

Table 3.3: Panoptic Segmentation Experiment. We evaluate the panoptic segmentation performance of models operating on a single modality and multimodal models trained with the early and late fusion strategies with temporal dropout.

	SQ	RQ	PQ	Parameter count
S2	81.3	49.2	40.4	1 318k
S1D	77.0	39.3	30.9	1 318k
S1A	77.4	38.8	30.6	1 318k
Early Fusion + Tdrop	**82.2**	**50.6**	**42.0**	1 791k
Late Fusion + Tdrop	81.6	50.5	41.6	2 390k

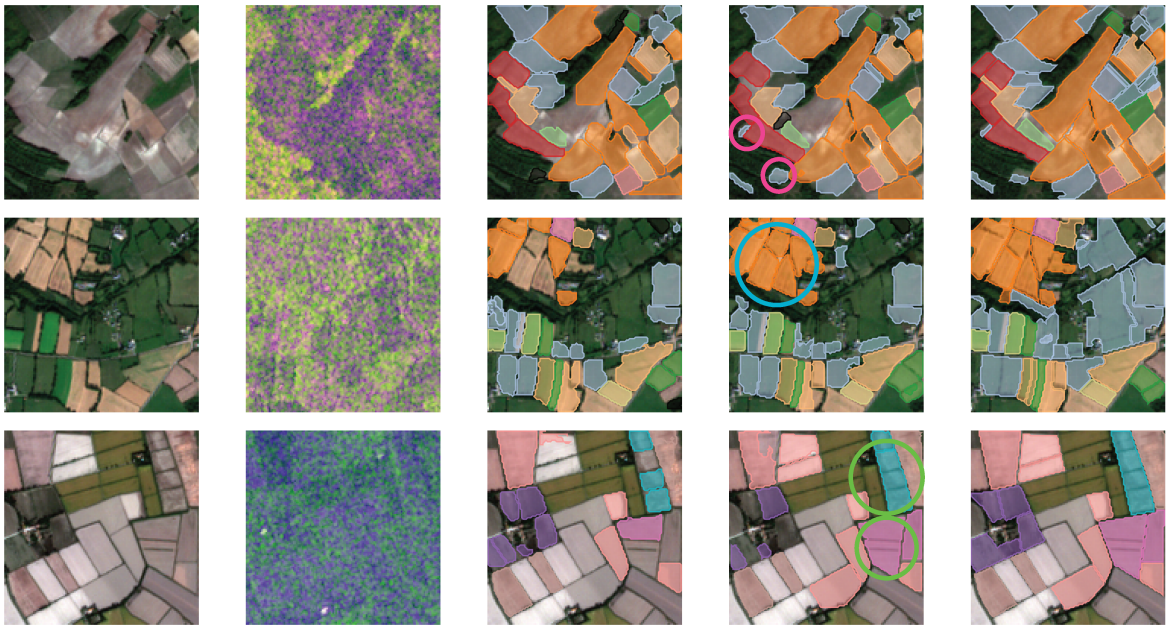
train the late fusion model.

Analysis. We report the results of this experiment on Table 3.3. Overall, the early and late fusion schemes increase the panoptic quality by 1.6pt and 1.2pt, respectively, compared to the optical baseline. This improvement is mostly driven by an increase in recognition quality, while the segmentation quality remains almost unchanged. This suggests that the radar modality helps in correctly detecting additional agricultural parcels, rather than refining the delineation of their boundaries. Although modest, this improvement is valuable for this notoriously complex task.

We show on Figure 3.9 the qualitative evaluation of the panoptic fusion model compared to the optical baseline. In practice, the fusion model seems to successfully retrieve more agricultural parcels, and also manages to retrieve small parcels that were missed by the optical model. We also display the predictions made by the unimodal models and compared to the predictions of the fusion model in Figure 3.10. These qualitative results show how the radar modality helps detecting more parcels than the optical baseline, or improving the semantic predictions of the fusion model. Additionally, given the relative noisiness of radar observations, the radar-only models retrieve surprisingly well the parcel boundaries. As mentioned previously, this could be attributed to the distinct volumetric radar response on parcel boundaries. We report the per-class performances on Figure 3.11.

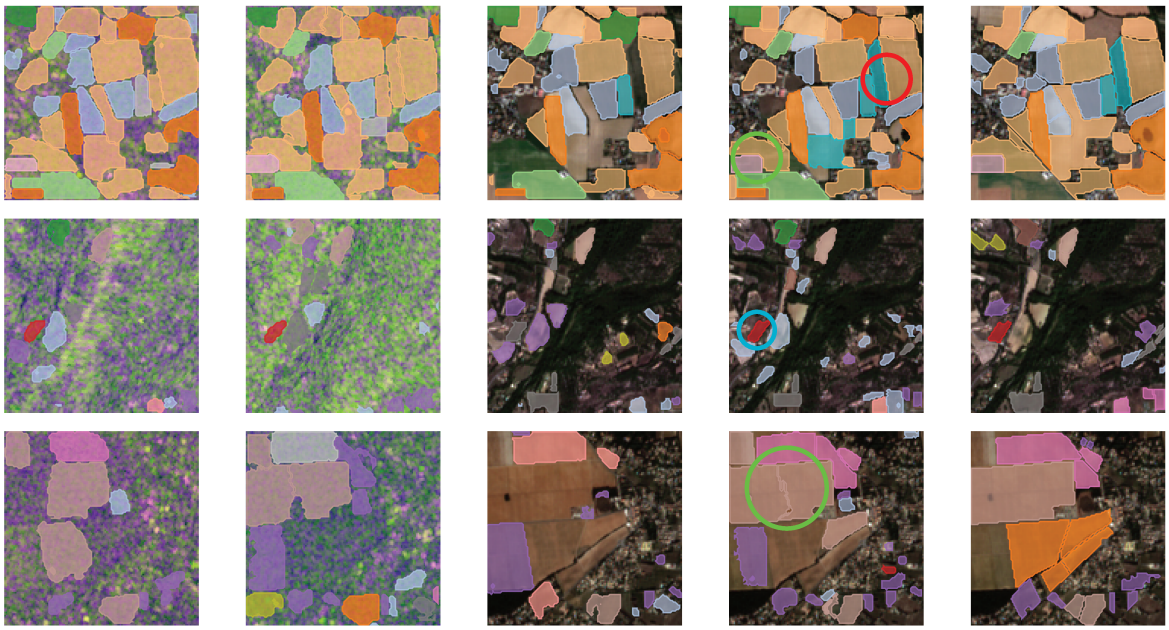
In terms of robustness to clouds, when performing inference on only 30% of the optical observa-

tions, the S2 baseline model drops to 33.0 PQ while our late fusion model maintains a score of 37.6 PQ. Consistently with the previous experiments, the addition of the radar modality helps improve the panoptic predictions with reduced availability of optical observations.



(a) Optical **(b)** Radar **(c)** S2 Prediction **(d)** Fusion Prediction **(e)** Ground Truth

Figure 3.9: Qualitative Results for Panoptic Segmentation. We show one observation from the optical time series in (a) and from the radar time series in (b). The prediction for the unimodal optical model is represented in (c) and the multimodal model in (d), and finally the ground truth in (e), with the same colormap as in Figure 3.7. The fusion model seems to retrieve more parcels (cyan circle ○), and even small-size parcels (magenta circle ○). We also note that the fusion model seems to handle parcels with internal subdivisions (green circle ○) better than the optical model.



(a) S1A Prediction **(b)** S1D Prediction **(c)** S2 Prediction **(d)** Fusion Prediction **(e)** Ground Truth

Figure 3.10: Qualitative Results for Panoptic Segmentation. We compare the predictions made by unimodal models operating on S1A (a), S1D (b), S2 (c), and the predictions made by the late fusion model (d). We also show the ground truth annotations (e). We observe cases where parcels are not detected by the optical model but successfully predicted by the radar-only models, and by the fusion model as well (green circle ○). We also note that some parcels are detected by the optical model, but the crop type is corrected by the addition of the radar modality (red circle ○). Conversely, some parcels are detected by the radar-only model with an incorrect crop type and not detected by the optical model. The combination of both modalities in the fusion model leads to a correct prediction. (cyan circle ○)

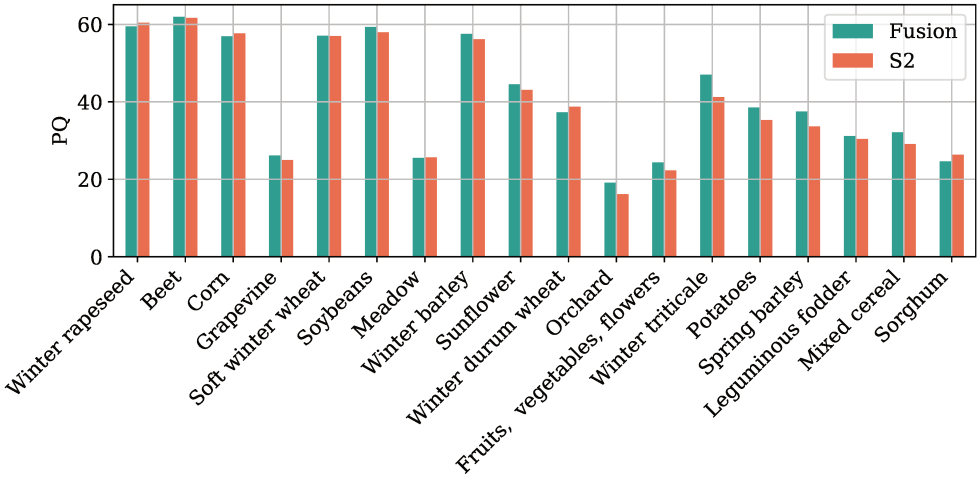


Figure 3.11: Classwise Performance for Panoptic Segmentation. We report the Panoptic Quality of the late fusion model with temporal dropout and the model trained purely on the optical modality. The classes are ordered as in Figure 3.5. In the panoptic setting, the radar modality is also specifically beneficial for hard classes such as *Winter triticale*.

3.3 Conclusion

To conclude, we discuss the relevance of the different modality fusion strategies, with a focus on Sentinel-1 & 2 data for crop mapping. Our experiments showed that combining optical and radar imagery allowed for an increase in performance for all tasks considered (Table 3.1, Table 3.2, Table 3.3) as well as robustness to cloud cover (Figure 3.8).

Table 3.4: Inference times. We report the inference times in seconds of Early and Late fusion for one fold of PASTIS (500 patches, 820km^2). We measure the combined data loading and prediction time, to accouint for the interpolation step in early fusion

	Parcel classification	Semantic segmentation	Panoptic segmentation
Early	192	280	**414**
Late	**149***	**259***	819

* with auxiliary loss.

3.3.1 Recommendations.

Our experiments showed that each fusion scheme has advantages and limitations influencing when its use is most relevant:

- **Early Fusion.** It is the most compact of the fusion models and shows competitive performance on all three tasks. The main drawback of this approach is the necessity of an expensive interpolation. As reported in Table 3.4, this preprocessing makes the early fusion scheme slower than late fusion despite relying on a smaller network for parcel classification and semantic segmentation. Early fusion is the least robust fusion scheme to cloud cover.

- **Mid Fusion.** Of all methods without preprocessing, this strategy leads to the fastest run time and the lowest memory requirement. It yields the second-best performance for parcel-based

classification but suffers more than late and decision fusion when the cloud cover is extensive. Its dependence on separate spatial and temporal encoders prevents its straightforward adaptation to pixel-based tasks. We recommend using this scheme for parcel classification in areas without extensive cloud cover and when inference speed is critical.

- **Late Fusion.** This fusion method, when combined with enhancement schemes, leads to the best performance and the highest adaptability, as well as excellent resilience to even extreme cloud cover. This method is our default recommendation when using temporal attention methods with multimodal time series.

- **Decision Fusion.** Despite having the highest parameter count, this method lags in terms of performance and is prohibitively costly for panoptic segmentation. However, it is the most resilient to cloud cover. We recommend using decision fusion when it is expected that only a few optical observations may be available for inference.

We also have evaluated the influence of two enhancement schemes:

- **Auxiliary Supervision.** This method consists in adding alongside the main prediction auxiliary predictions based on one modality alone. The rationale is to help each specialized module to learn meaningful features regardless of the interplay with other modalities. We observe a strong effect in precision for late and decision fusion, which have dedicated encoding modules for each modality.

- **Temporal Dropout.** This simple method consists in randomly dropping acquisitions of the time series considered. Its effect was beneficial to all fusion schemes and the optical baseline across our experiments. Another benefit of this scheme is that it reduces the memory footprint of networks during training.

3.3.2 Limitations.

Our study hinged on the PASTIS dataset, which contains annotated agricultural parcels from four different regions of the French metropolitan territory. In this regard, our results are most relevant for crop mapping applications with the same meteorological context, terrain conditions, and crop types as this region. Certain crop types not observed in PASTIS could benefit even more from the radar modality than our results show. For instance, rice fields are often filled with water and thus have a distinctive SAR response but are not represented in PASTIS.

Furthermore, our evaluation of cloud robustness focused on assessing the effect of a reduced number of optical observations *at inference time*. This corresponds to artificially increasing the cloud cover in the test set without affecting crop growth. A more rigorous approach would constitute a dataset comprising truly observed cloud coverage by varying the regions and years of acquisition. This is complicated by the lack of harmonization between LPIS across different countries in nomenclature and open-access policy. Lastly, we only used backscattering coefficients from the SAR data in our experiments, as is commonly done in the crop type mapping literature [116]. Mestre-Quereda et al. [103] found that the addition of interferometric radar features is beneficial to crop classification when using only radar inputs. Further work is needed to assess the benefit of interferometric radar features in a fusion setting with optical imagery. Moreover, we chose to prepare the SAR inputs with limited preprocessing. We do not apply speckle filtering or radiometric terrain correction to compensate for the effect of the local incident angle. Interestingly, our experiments showed that this does not prevent the radar modality from benefiting crop mapping models. However, further studies could evaluate the benefit of adding speckle filtering, elevation information, or meteorological context to networks using radar images for crop mapping.

Classifications are theories about the basis of natural order,
not dull catalogues compiled only to avoid chaos.

Stephen Jay Gould

4

Leveraging the class hierarchy

In this section, we explore how we can use the hierarchical structure of the class set to improve the precision of classification models. As this concerns virtually any classification problem, we widen our scope to other computer vision problems and datasets. In particular, this also applies very well to our crop type mapping problem.

4.1 Metric-guided prototype learning

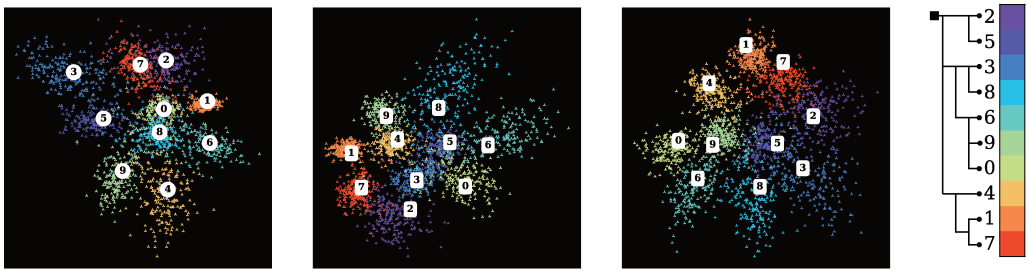
4.1.1 Motivation

Most classification models focus on maximizing the prediction accuracy, regardless of the semantic nature of errors. This can lead to high performing models, but puzzling errors such as confusing tigers and sofas, and casts doubt on what a model actually understands of the required task and data distribution. Neural networks in particular have been criticised for their tendency to produce improbable yet confident errors, notably when under adversarial attacks[4]. Training deep models to produce not only produce fewer but also *better* errors can increase their trustworthiness, which is crucial for downstream applications such as autonomous driving or land use and land cover monitoring[12,31].

In many classification problems, the target classes can be organised according to a tree-shaped hierarchical structure. Such a taxonomy can be generated by domain experts, or automatically inferred from class names using the WordNet graph[106] or from word embeddings[105]. A step towards more reliable and interpretable algorithms would be to explicitly model the difference of gravity between errors, as defined by a hierarchical nomenclature.

For a classification task over a set $\mathcal{K}$ of K classes, the hierarchy of errors can be encapsulated by a cost matrix $D \in \mathbb{R}_+^{K \times K}$, defined such that the cost of predicting class k when the true class is l is $D[k, l] \geq 0$, and $D[k, k] = 0$ for all $k = 1 \cdots K$. Among many other options[81], one can define $D[k, l]$ as the length of the shortest path between the nodes corresponding to classes k and l in the tree-shaped class taxonomy.

As pointed out by Bertinetto et al. [12], the first step towards algorithms aware of hierarchical structures would be to generalize the use of cost-based metrics. For example, early iterations of the ImageNet challenge[129,31] proposed to weight errors according to hierarchy-based costs. For a dataset indexed by $\mathcal{N}$, the *Average Hierarchical Cost* (AHC) between class predictions $y \in \mathcal{K}^{\mathcal{N}}$ and the true



(a) Cross entropy, distortion[1] = 0.47, ER= 15.2%, AHC = 0.81

(b) Learnt prototypes, distortion= 0.42, ER= 14.2%, AHC = 0.75

(c) Guided prototypes, distortion= 0.22, ER= 11.9%, AHC = 0.52

Figure 4.1: Illustrative example on MNIST. Mean class representation ○, prototypes □, and 2-dimensional embeddings ⅄ learnt on perturbed MNIST by a 3-layer convolutional net with three different classification modules: (a) cross-entropy, (b) learnt prototypes, and (c) learnt prototypes guided by a tree-shaped taxonomy (constructed according to the authors' perceived visual similarity between digits). The guided prototypes (d) embed more faithfully the class hierarchy: classes with low error cost are closer. This is associated with a decrease in the *Average Hierarchical Cost* (AHC), as well as *Error Rate* (ER), indicating that our taxonomy may contain useful information for learning better visual features.

labels $z \in \mathcal{K}^{\mathcal{N}}$ is defined as:

$$\mathrm{AHC}(y, z) = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} D[y_n, z_n] \,. \tag{4.1}$$

Along with the evaluation metrics, the loss functions should also take the cost matrix into account. While it is common to focus on retrieving certain classes through weighting[92,18] or sampling[155,143] schemes, preventing confusion between specific classes is less straightforward. For example, the cross entropy with one-hot target vectors singles out the predicted confidence for the true class, but treats all other classes equally. Beyond reducing the AHC, another advantage of incorporating the class hierarchy into the learning phase is that D may contain information about the structure of the data as well. Although it is not always the case, co-hyponyms (*i.e.*, siblings) in a class hierarchy tend to share some structural properties. Encouraging such classes to have similar representations could lead to

more efficient learning, *e.g.*, by leveraging common feature detectors. Such priors on the class structure may be especially crucial when dealing with a large taxonomy, as noted by Deng et al. [31].

In this section, we introduce a method to integrate a pre-defined class hierarchy into a classification algorithm. We propose a new distortion-based regularizer for prototypical network [186,23]. This penalty allows the network to learn prototypes organised so that their pairwise distances reflect the error cost defined by a class hierarchy. The key contributions of this chapter are as follows:

- We introduce a scale-independent formulation of the distortion between two metric spaces and an associated smooth regularizer.
- This formulation allows us to incorporate knowledge of the class hierarchy into a neural network at no extra cost in trainable parameters and computation.
- We show on four public datasets (CIFAR100 , NYUDv2, S2-Agri, and iNaturalist-19) that our approach decreases the average cost of the prediction of standard backbones.
- As illustrated in Figure 4.1, we show that our approach can also lead to a better (unweighted) precision, which we attribute to the useful priors contained in the hierarchy.

4.1.2 Related Work

Prototypical Networks. Our approach builds on the growing corpus of work on prototypical networks. These models are deep learning analogues of nearest centroid classifiers [160] and Learning Vector Quantisation networks [137,78], which associate to each class a representation, or prototype, and classify the observations according to the nearest prototype. These networks have been successfully used for few-shot learning [147,33], zero-shot learning [68], and supervised classification [52,186,104,23].

*For a formal definition of scale-free distortion, see Section 4.1.3.2; the distortion is computed with respect to the means of class embeddings for the cross entropy.

In most approaches, the prototypes are directly defined as the centroid of the learnt representations of samples of their classes, and updated at each episode [147] or iteration [52]. In the work of Mettes et al. [104] and Jetley et al. [68], the prototypes are defined prior to learning the embedding function. In this work, we follow the approach of Yang et al. [186] and learn the prototypes simultaneously with the data embedding function.

Hierarchical Priors. The idea of exploiting the latent taxonomic structure of semantic classes to improve the accuracy of a model has been extensively explored [144], from traditional Bayesian modeling [46] to adaptive deep learning architectures [185,128,136,7]. However, for these neural networks, the hierarchy is discovered by the network itself to improve the overall accuracy of the model. In our setting, the hierarchy is defined a priori and serves both to evaluate the quality of the model and to guide the learning process towards a reduced prediction cost.

Srivastava and Salakhutdinov [151] propose to implement Gaussian priors on the weight of neurons according to a fixed hierarchy. Redmon & Farhadi [124] implements an inference scheme based on a tree-shaped graphical model derived from a class taxonomy. Closest to our work, Hou et al. [61] propose a regularisation based on the earth mover distance to penalize errors with high cost.

More recently, Bertinetto et al. [12] highlighted the relative lack of well-suited methods for dealing with hierarchical nomenclatures in the deep learning literature. They advocate for a more widespread use of the AHC for evaluating models, and detail two simple baseline classification modules able to decrease the AHC of deep models: *Soft-Labels* and *Hierarchical Cross-Entropy*. Following this objective, Karthik et al. [74] propose a an inference-time risk minimisation scheme to reduce the AHC of the predictions based on the predicted posteriors.

Hyperbolic Prototypes. Motivated by their capacity to embed hierarchical data structures into low-dimensional spaces [30], hyperbolic spaces are at the center of recent advances in modeling hierar-

chical relations [113,75]. Closer to this work, Liu et al. [95] and Long et al. [96] also propose to embed a class hierarchy into the latent representation space. However, both approaches embed the class hierarchy before training the data embedding network. In contrast, we argue that incorporating the hierarchical structure during the training of the model allows the network and class embeddings to share their respective insights, leading to a better trade-off between AHC and accuracy. In this section, we only explore Euclidean geometry, as this setting allows for the seamless integration of our method without changing the number of bits of precision or the optimizer [30].

Finite Metric Embeddings. Our objective of computing class representations with pairwise distances determined by a cost matrix has links with finding an isometric embedding of the cost matrix—seen as a finite metric. This problem has been extensively studied [65,14] and is at the center of the growing interest for hyperbolic geometry [30]. Here, our goal is simply to influence the learning of prototypes with a metric rather than necessarily seeking the best possible isometry.

4.1.3 Methods

We consider a generic dataset $\mathcal{N}$ of N elements $x \in \mathcal{X}^{\mathcal{N}}$ with ground truth classes $z \in \mathcal{K}^{\mathcal{N}}$. The classes $\mathcal{K}$ are organised along a tree-shape hierarchical structure, allowing us to define a cost matrix D by considering the shortest path between nodes. The matrix thus defined is symmetric, with zero diagonal, strictly positive elsewhere, and respects the triangle inequality: $D[k,l] + D[l,m] \geq D[k,m]$ for all k, l, m in $\mathcal{K}$. In other words, D defines a finite metric. We denote by Ω an *embedding space* which, when equipped with the distance function $d : \Omega \times \Omega \mapsto \mathbb{R}_+$, forms a continuous metric space.

4.1.3.1 Prototypical Networks

A prototypical network is characterised by an embedding function $f : \mathcal{X} \mapsto \Omega$, typically a neural network, and a set $\pi \in \Omega^{\mathcal{K}}$ of K prototypes. π must be chosen such that any sample x_n of true class k has a representation $f(x_n)$ which is *close* to π_k and *far* from other prototypes.

Following the methodology of Snell et al. [147], a prototypical network (f, π) associates to an observation x_n the posterior probability over its class z_n defined as follows:

$$p(z_n = k|x_n) = \frac{\exp\left(-d\left(f(x_n), \pi_k\right)\right)}{\sum_{l \in \mathcal{K}} \exp\left(-d\left(f(x_n), \pi_l\right)\right)}, \forall k \in \mathcal{K} \tag{4.2}$$

We define an associated loss as the normalised negative log-likelihood of the true class:

$$\mathcal{L}_{\text{data}}(f, \pi) = \frac{1}{N} \sum_{n \in \mathcal{N}} \left(d(f(x_n), \pi_{z_n}) + \log \left(\sum_{l \in \mathcal{K}} \exp\left(-d(f(x_n), \pi_l)\right) \right) \right) . \tag{4.3}$$

This loss encourages the representation $f(x_n)$ to be close to the prototype of the class $z_n$and far from the other prototypes. Conversely, the prototype π_k is drawn towards the representations $f(x_n)$ of samples n with true class k, and away from the representations of samples of other classes.

Following the insights of Yang et al. [186], the embedding function f and the prototypes π are learned simultaneously. This differs from many works on prototypical networks which learn prototypes separately or define them as centroids of representations. We take advantage of this joint training to learn prototypes which take into account both the distribution of the data and the relationships between classes, as described in the next section.

4.1.3.2 Metric-Guided Penalisation

We propose to incorporate the cost matrix D into a regularisation term in order to encourage the prototypes' positions in the embedding space Ω to be consistent with the finite metric defined by D. Since

the sample representations are attracted to their respective prototypes in (4.3), such regularisation will also affect the embedding network.

Metric Distortion. As described in De Sa et al. [30], the distortion of a mapping $k \mapsto \pi_k$ between the finite metric space $(\mathcal{K}, D)$ and the continuous metric space (Ω, d) can be defined as the average relative difference between distances in the source and target space:

$$\text{disto}(\pi, D) = \frac{1}{K(K-1)} \sum_{k,l \in \mathcal{K}^2,\, k \neq l} \frac{|d(\pi_k, \pi_l) - D[k,l]|}{D[k,l]} \,. \tag{4.4}$$

We argue that a network (f, π) trained to minimize $\mathcal{L}_{\text{data}}$ and whose prototypes π have a low distortion with respect to D should produce errors with low hierarchical costs. To understand the intuition behind this idea, let us consider a sample x_n of true class k and misclassified as class l. This tells us that the distance between $f(x_n)$ and π_l is small. If k and l have a high cost according to D, and since $k \mapsto \pi_k$ is of low distortion, then $d(\pi_k, \pi_l)$ must be large. The triangular inequality tells us that $d(f(x_n), \pi_k) \geq d(\pi_k, \pi_l) - d(f(x_n), \pi_l)$, and consequently that $d(f(x_n), \pi_k)$ must be large as well, which contradicts that (f, π) minimizes $\mathcal{L}_{\text{data}}$.

Scale-Free Distortion. For a prototype arrangement π to have a small distortion with respect to a finite metric D as defined in Equation 4.4, the distance between prototypes must correspond to the distance between classes. This imposes a specific scale on the distances between prototypes in the embedding space. This scale may conflict with the second term of $\mathcal{L}_{\text{data}}$ which encourages the distance between embeddings and unrelated prototypes to be as large as possible. Therefore, lower distortion may also cause lower precision. To remove this conflicting incentive, we introduce a scale-independent formulation of the distortion (4.5) where $s \cdot \pi$ are the scaled prototypes, whose coordinates in Ω are multiplied by a scalar factor s.

$$\text{disto}^{\text{scale-free}}(\pi, D) = \min_{s \in \mathbb{R}_+} \text{disto}(s \cdot \pi, D) \,, \tag{4.5}$$

Computing the scale-free distortion defined in Equation 4.5 amounts to finding a minimizer of the following function $f : \mathbb{R} \mapsto \mathbb{R}$:

$$f(s) = \sum_{i \in I} |s\alpha_i - 1| \,, \tag{4.6}$$

with $\alpha_{k,l} = d(\pi_k, \pi_l)/D[k, l] \geq 0$, and I an ordering of $\{k, l\}_{k,l \in \mathcal{K}^2}$ such that the sequence $[\alpha_i]_{i \in I}$ is non-decreasing.

Proposition 1. *A global minimizer of f defined in (4.6) is given by $s^\star = 1/\alpha_{k^\star}$ with $k^\star$ defined as:*

$$k^\star = \min \left\{ k \in I \,\middle|\, \sum_{i \leq k} \alpha_i \geq \sum_{i > k} \alpha_i \,. \right\} \tag{4.7}$$

Proof. First, such $k^\star$ exists as it is the smallest member of a discrete, non-empty set. Indeed, since all α_i are nonnegative, the set contains at least $k = |I|$. We now verify that $s^\star = 1/\alpha_k^\star$ is a critical point of f. By definition of $k^\star$ we have that $\sum_{i \leq k^\star} \alpha_i \geq \sum_{i > k^\star} \alpha_i$ and $\sum_{i < k^\star} \alpha_i < \sum_{i \geq k^\star} \alpha_i$. By combining these two inequalities, we have that

$$-\sum_{i < k^\star} \alpha_i + \sum_{i > k^\star} \alpha_i \in [-\alpha_{k^\star}, \alpha_{k^\star}] \,. \tag{4.8}$$

Since I orders the α_i in increasing order, we can write the subgradient of f at $s^\star$ under the following

form:

$$\partial_s f(s^\star) = \sum_{i<k^\star} \partial_s \left|s^\star \alpha_i - 1\right| + \sum_{i>k^\star} \partial_s \left|s^\star \alpha_i - 1\right| + \partial_s \left|s^\star \alpha_{k^\star} - 1\right| \tag{4.9}$$

$$= -\sum_{i<k^\star} \alpha_i + \sum_{i>k^\star} \alpha_i + [-\alpha_{k^\star}, \alpha_{k^\star}]. \tag{4.10}$$

By using the inequality defined in Equation 4.8, we have that $0 \in \partial_s f(s^\star)$ and hence $s^\star$ is a critical point of f. Since f is convex, such $s^\star$ is also a global minimizer of f, *i.e.*, an optimal scaling. ■

This proposition gives us a fast algorithm to obtain an optimal scaling and hence a scale-free distortion: compute the cumulative sum of the $\alpha_{k,l}$ sorted in ascending order until the equality in (4.7) is first verified at index $k^\star$. The resulting optimal scaling is then given by $1/\alpha_{k^\star}$.

Distortion-Based Penalisation. We propose to incorporate the error qualification D into the prototypes' relative arrangement by encouraging a low *scale-free* distortion between π and D. To this end, we define $\mathcal{L}_{\text{disto}}$, a smooth surrogate of $\text{disto}^{\text{scale-free}}$ (4.11).

$$\mathcal{L}_{\text{disto}}(\pi) = \frac{1}{K(K-1)} \min_{s \in \mathbb{R}_+} \sum_{k,l \in \mathcal{K}^2,\, k \neq l} \left(\frac{s d(\pi_k, \pi_l) - D[k,l]}{D[k,l]} \right)^2 . \tag{4.11}$$

The minimisation problem with respect to s defined in Equation 4.11 can be solved in closed form and $\mathcal{L}_{\text{disto}}$ can thus be directly used as a regularizer. :

$$s^\star = \sum \frac{d(\pi_k, \pi_l)}{D[k,l]} \Big/ \sum \frac{d(\pi_k, \pi_l)^2}{D[k,l]^2} . \tag{4.12}$$

4.1.3.3 End-to-end Training

We combine $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{disto}}$ in a single loss $\mathcal{L}$. $\mathcal{L}_{\text{data}}$ allows to jointly learn the embedding function f and the class prototypes π, while $\mathcal{L}_{\text{disto}}$ enforces a metric-consistent prototype arrangement, with

$\lambda \in \mathbb{R}_+$ an hyper-parameter setting the strength of the regularisation:

$$\mathcal{L}(f, \pi) = \mathcal{L}_{\text{data}}(f, \pi) + \lambda \mathcal{L}_{\text{disto}}(\pi) \,. \tag{4.13}$$

4.1.3.4 Choosing a Metric Space

Prototypical networks operating on $\Omega = \mathbb{R}^m$ typically use the squared Euclidean norm in the distance function, motivated by its quality as a Bregman divergence [147]. However, given that the metric penalizers tend to produce prototypes which are further apart than their unguided counterparts, the square norm makes learning less stable. We observe that defining d with the Euclidean norm yields significantly better results.

The non-differentiability can be handled by composing with a Huber-like [62,22] function $d = H(\|\cdot\|)$, with H defined in Equation 4.14 and $\delta \in \mathbb{R}_+$ a (small) hyper-parameter. The resulting metric d is asymptotically equivalent to the Euclidean norm for large distances and behaves like the smooth squared Euclidean norm for small distances. In Section 4.2.4.2, we investigate the effect of this change.

$$H(x) = \delta(\sqrt{\|x\|^2/\delta^2 + 1} - 1) \,, \tag{4.14}$$

4.2 Numerical experiments

4.2.1 Datasets and backbones

We evaluate our approach with different tasks and public datasets with fine-grained class hierarchies: for image classification on CIFAR100 [83] and iNaturalist-19 [169], RGB-D image segmentation on NYUDv2 [110], and parcel-based crop type classification on the dataset introduced in Chapter 1 (S2-Agri)[†]. We define the cost matrix of these class sets as the length of the shortest path between nodes

[†] This work was carried out before the introduction of the PASTIS dataset.

Table 4.1: Datasets. Composition and taxonomies of the four studied datasets. IR stands for the Imbalance Ratio (largest over smallest class count), nodes and leaves denote respectively the total number of classes and leaf-classes in the tree-shape hierarchy, ABF stands for the Average Branching Factor, and $\langle D \rangle$ stands for the average pairwise distance.

Dataset	Data			Hierarchical Tree			
	Volume (Gb)	Samples	IR	Depth	Nodes (leaves)	ABF	$\langle D \rangle$
NYUDv2	2.8	1449	93	3	57 (40)	5.0	4.3
S2-Agri	28.2	189 971	617	4	83 (45)	5.8	6.5
CIFAR100	0.2	60 000	1	5	134 (100)	3.8	7.0
iNat-19	82.0	265 213	31	7	1189 (1010)	6.6	11.0

in the associated tree-shape taxonomies represented Figures 4.2 to 4.5. As shown in Table 4.1, these datasets cover different settings in terms of data distribution and hierarchical structure.

Image Classification on CIFAR100. CIFAR100 is composed of 50 000 training images and 10 000 test images of size 32×32, evenly distributed across 100 classes. We use a super-class system inspired by Krizhevsky & Hinton [83] and form a 5-level hierarchical nomenclature of size: 2, 4, 8, 20, and 100 classes (see Figure 4.2). We use as backbone the established ResNet-18 [57] as embedding network for this dataset.

RGB-D Semantic Segmentation on NYUDv2. We use the standard split of 795 training and 654 testing pairs. We combine the 4 and 40 class nomenclatures of Gupta et al. [53] and the 13 class system defined by Handa et al. [54] to construct a 3-level hierarchy (see Figure 4.3). We use FuseNet [55] as backbone for this dataset.

Parcel-based classification. We use the dataset presented in Chapter 1. We define a 4-level crop type hierarchy of size 4, 12, 19, and 44 classes with the help of experts from a European agricultural monitoring agency (ASP) (see Figure 4.4). We use the PSE+TAE architecture (Chapter 1) as backbone, with the same 5-fold cross-validation scheme for training. Crop mapping in particular bene-

fits from predictions with a low hierarchical cost. Indeed, payment agencies monitor the allocation of agricultural subsidies and whether crop rotations follow best practice recommendations[51]. The monetary and environmental impact of misclassifications are typically reflected in the class hierarchy designed by domain experts[15,17]. By achieving a low AHC, we ensure that these downstream tasks can be meaningfully realised from the predictions.

Fine-Grained Image Classification on iNaturalist-19 (iNat-19). iNat-19[169] contains 1 010 different classes organised into a 7 level hierarchy with respective width 3, 4, 9, 34, 57, 72, and 1 010 (see Figure 4.5). We use ResNet-18 pretrained on ImageNet as backbone. We sample 75% of available images for training, while the rest is evenly split into a validation and test set.

Illustrative Example on MNIST. In Figure 4.1, we illustrate the difference in performance and embedding organisation of the embedding space for different approaches. We use a small 3-layer convolutional net trained on MNIST with random rotations (up to 40 degrees) and affine transformations (up to 1.3 scaling). For plotting convenience, we set the features' dimension to 2.

4.2.2 Implementation details

CIFAR100 ResNet-18 is trained on CIFAR100 using SGD with initial learning rate $l_r = 10^{-1}$, momentum set to 0.9 and weight decay $w_d = 5 \cdot 10^{-4}$. The network is trained for 200 epochs in batches of size 128, and the learning rate is divided by 5 at epochs 60, 120, and 160. The model is evaluated using its weights of the last epoch of training, and the results we report are median values over 5 runs.

NYUDv2 We train FuseNet on NYUDv2 using SGD with momentum set to 0.9. The learning rate is set initially to 10^{-3} and multiplied at each epoch by a factor that exponentially decreases from

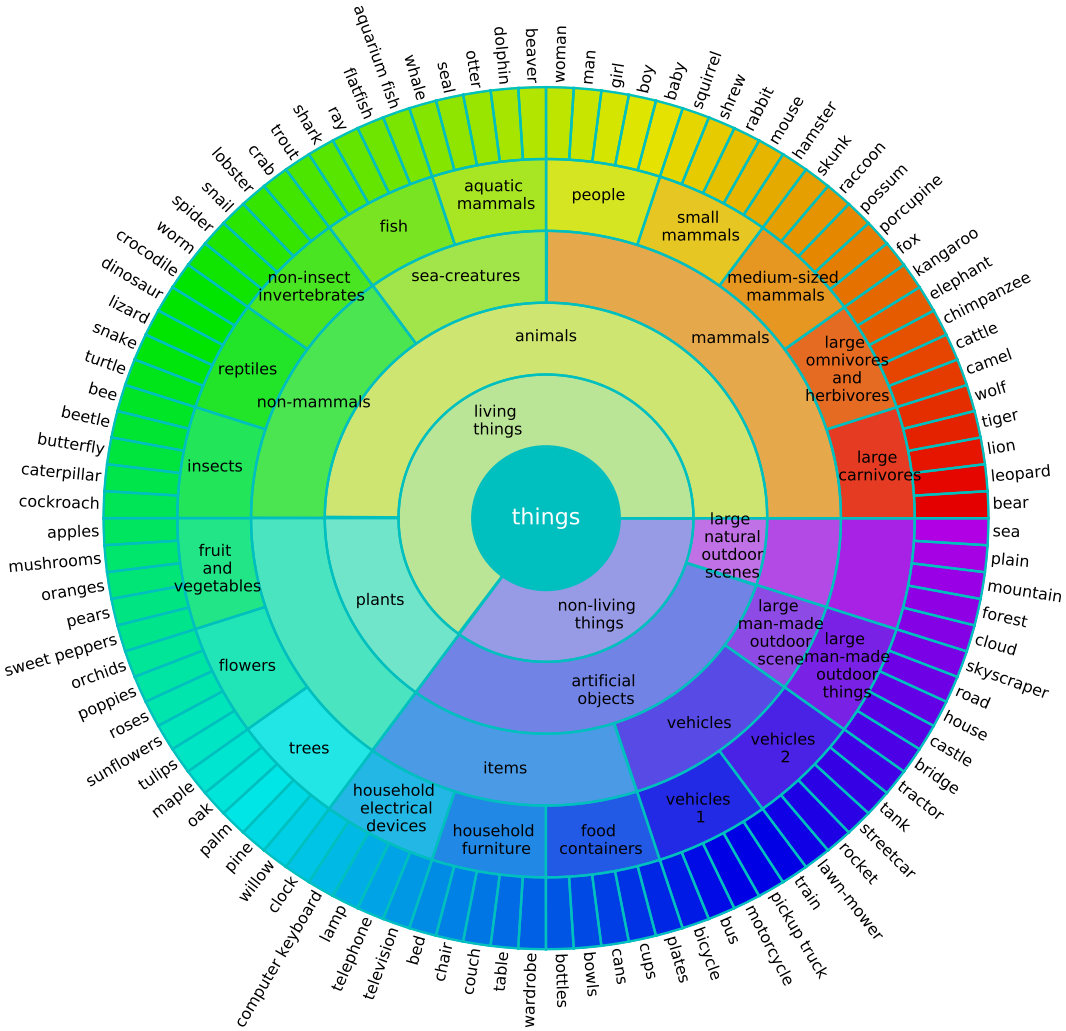


Figure 4.2: CIFAR100 class hierarchy. The arcs at different radii represent the different classes of each level of the hierarchy. Unlabelled arcs share the same name as their parent class.

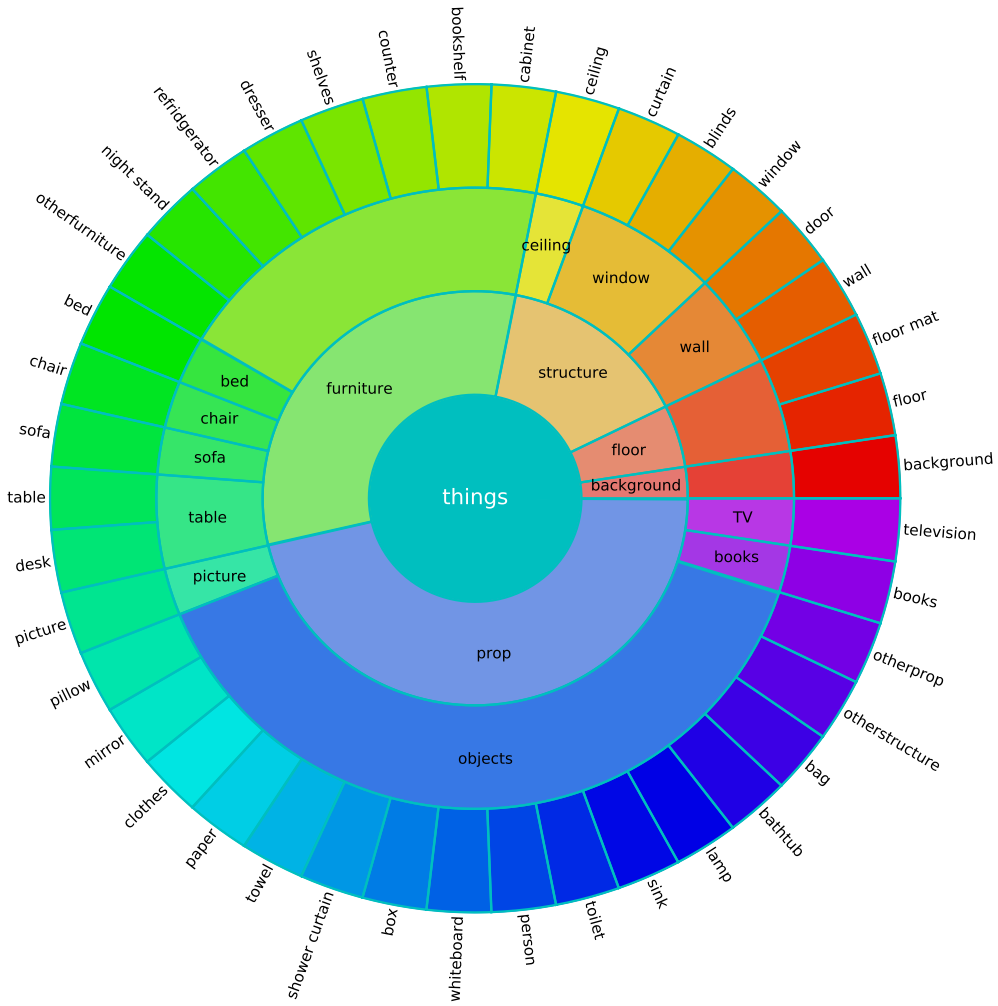


Figure 4.3: NYUv2 class hierarchy.

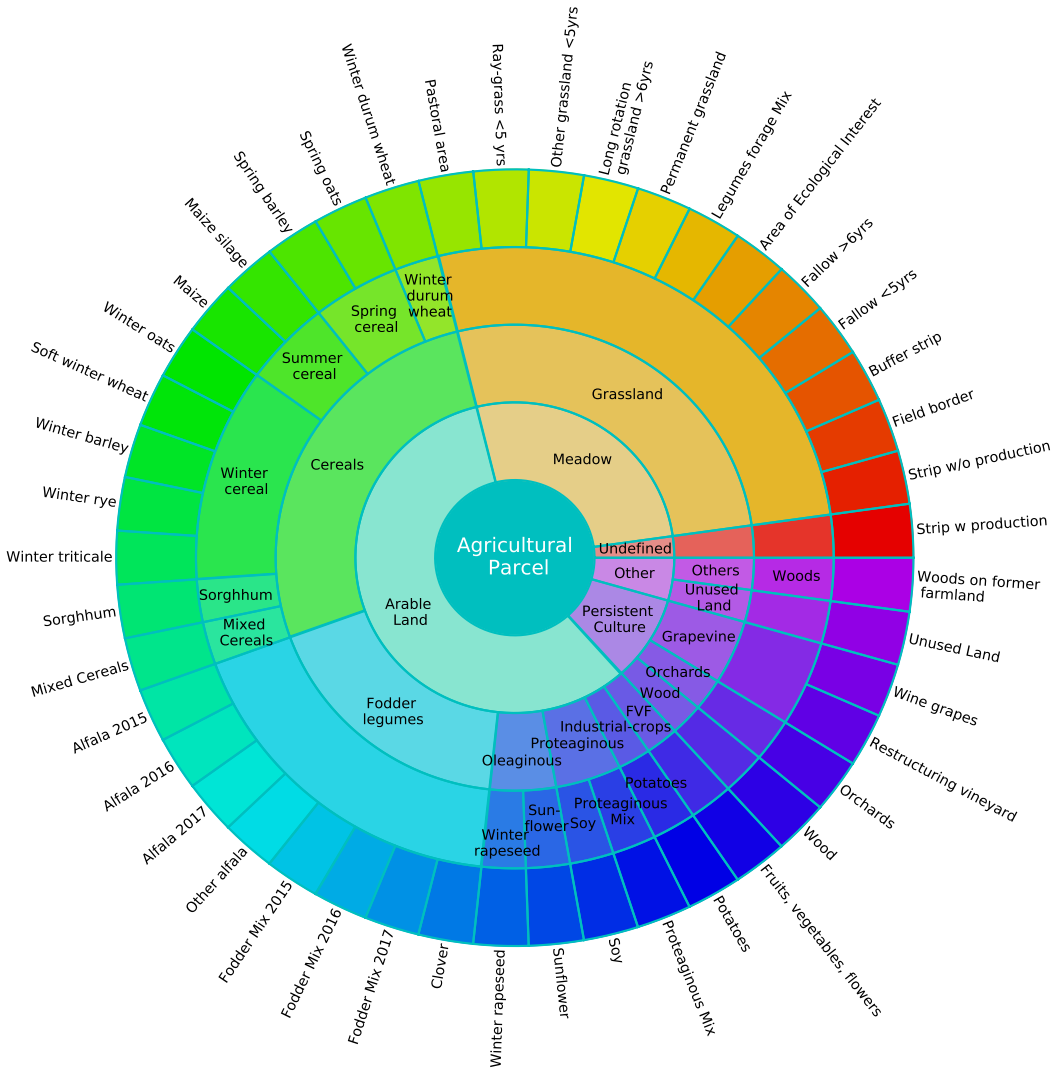


Figure 4.4: S2-Agri class hierarchy.

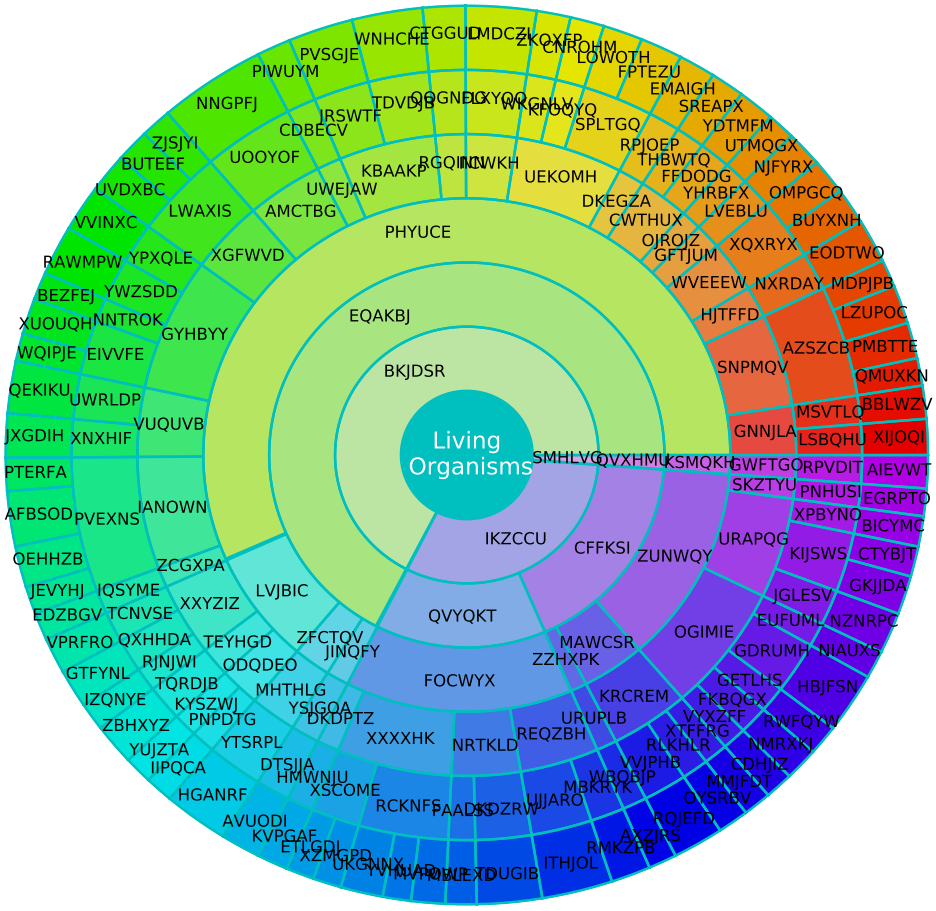


Figure 4.5: iNat-19 class hierarchy. Only the first 6 levels of the hierarchy are represented. At the time of writing, only the classes' obfuscated names were publicly available

1 to 0.9. The network is trained for 300 epochs in batches of 4 with weight decay set to $5 \cdot 10^{-3}$. We report the performance of the best-of-five last testing epochs.

S2-Agri We train PSE+TAE on S2-Agri using Adam with $l_r = 10^{-3}, \beta = (0.9; 0.999)$ and no weight decay. The dataset is randomly separated in five splits. For each of the five folds, 3 splits are used as training data on which the network is trained in batches of 128 samples for 100 epochs. The best epoch is selected based on its performance on the validation set, and we use the last split to measure the final performance of the model. We report the average performance over the five folds.

iNaturalist-19 Given the complexity of the dataset, we follow [12] and use a ResNet-18 pre-trained on ImageNet. The network is trained for 65 epochs in batches of 64 epochs using Adam with $l_r = 10^{-4}, \beta = (0.9; 0.999)$ and no weight decay. The best epoch is selected based on the performance on the validation set, and we report the performance on the held-out test set.

MGP Hyper-Parameterisation. The embedding space Ω is chosen as $\mathbb{R}^{512}$ for iNat-19 and $\mathbb{R}^{64}$ for all other datasets. We chose d as the Euclidean norm. (see 4.2.4.2 for a discussion on this choice). We evaluate our approach (Guided-proto) with $\lambda = 1$ in (4.13) for all datasets. We use the same training schedules and learning rates as the backbone networks in their respective papers. In particular, the class imbalance of S2-Agri is handled with a focal loss [92].

4.2.3 Competing methods

In the paper where they are introduced, all backbone networks presented in Section 4.2.1 use a linear mapping between the sample representation and the class scores, as well as the cross-entropy loss. The resulting performance defines a baseline, denoted as `Cross-Entropy`, and is used to estimate the gains in Average Hierarchical Cost (AHC) and Error Rate (ER) provided by different approaches.

We reimplemented other competing methods:

- **Hierarchical Cross-Entropy** (HXE): Bertinetto et al. [12] model the class structure with a hierarchical loss composed of the sum of the cross-entropies at each level of the class hierarchy. As suggested, a parameter α taken as 0.1 defines exponentially decaying weights for higher levels.

- **Soft Labels** (Soft-labels): Bertinetto et al. [12] propose as second baseline in which the the one-hot target vectors are replaced by soft target vectors in the cross-entropy loss. These target vectors are defined as the softmin of the costs between all labels and the true label, with a temperature $1/\beta$ chosen as 0.1, as recommended in Bertinetto et al. [12].

- **Earth Mover Distance regularisation**(XE+EMD): Hou et al. [61] propose to account for the relationships between classes with a regularisation based on the squared earth mover distance. We use D as the ground distance matrix between the probabilistic prediction p and the true class y. This regularizer is added along the cross-entropy with a weight of 0.5 and an offset μ of 3.

- **Hierarchical Inference** (YOLO): Redmon & Farhadi [124] propose to model the hierarchical structure between classes into a tree-shaped graphical model. First, the conditional probability that a sample belongs to a class given its parent class is obtained with a softmax restricted to the class' co-hyponyms (*i.e.*, siblings). Then, the posterior probability of a leaf class is given by the product of the conditional probability of its ancestors. The loss is defined as the cross-entropy of the resulting probability of the leaf classes.

- **Hyperspherical Prototypes** (Hyperspherical-proto): The method proposed by Mettes et al. [104] is closer to ours, as it relies on embedding class prototypes. They advocate to first position prototypes on the hypersphere using a rank-based loss (see Section 4.2.4.2) combined with a prototype separating term. They then use the squared cosine distance between the image embed-

dings and prototypes to train the embedding network. Note that in our re-implementation, we used the finite metric defined by D instead of Word2Vec [105] embeddings to position prototypes. Lastly, we do not evaluate on S2-Agri as the integration of the focal loss is non-trivial.

- **Deep Mean Classifiers** (Deep-NCM): Guerriero et al. [52] present another prototype-based approach. Here, the prototypes are the cumulative mean of the embeddings of the classes' samples, updated at each iteration. The embedding network is supervised with $\mathcal{L}_{\text{data}}$ with d defined as the squared Euclidean norm.

- **Learnt-Proto**: Lastly, we evaluate simple prototype learning [186] by setting $\lambda = 0$ in (4.13).

4.2.4 Results

Overall Performance. As displayed in Figure 4.6, the benefits provided by our approach can be appreciated on all datasets. Compared to the `Cross-entropy` baseline, our model improves the AHC by 3% on NYUDv2 and S2-Agri, and up to 9% and 14% for CIFAR100, and iNat-19 respectively. The hierarchical inference scheme `YOLO` of Redmon & Farhadi [124] performs on par or better than our methods for NYUDv2 and S2-Agri, while `Soft-labels` perform well on CIFAR100 and NYUDv2. Yet, metric-guided prototypes bring the most consistent reduction of the hierarchical cost across all tasks, datasets, and class hierarchies configurations. This suggest that arranging the embedding space consistently with the cost metric is a robust way of reducing a model's hierarchical error cost. We argue that these results, combined with its ease of implementation, make a strong case for our approach.

While being initially designed to reduce the AHC, our method also provides a relative decrease of the ER by 3 to 4% across all datasets compared to the cross-entropy baseline. This indicates that cost matrices derived from the class hierarchies can indeed help neural networks to learn richer representations.

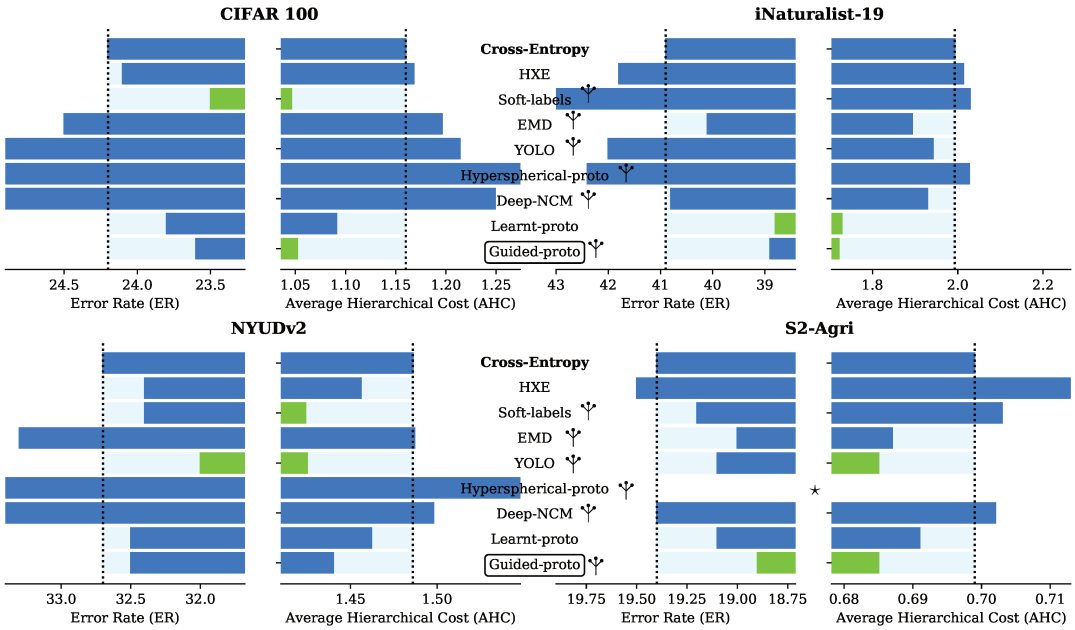
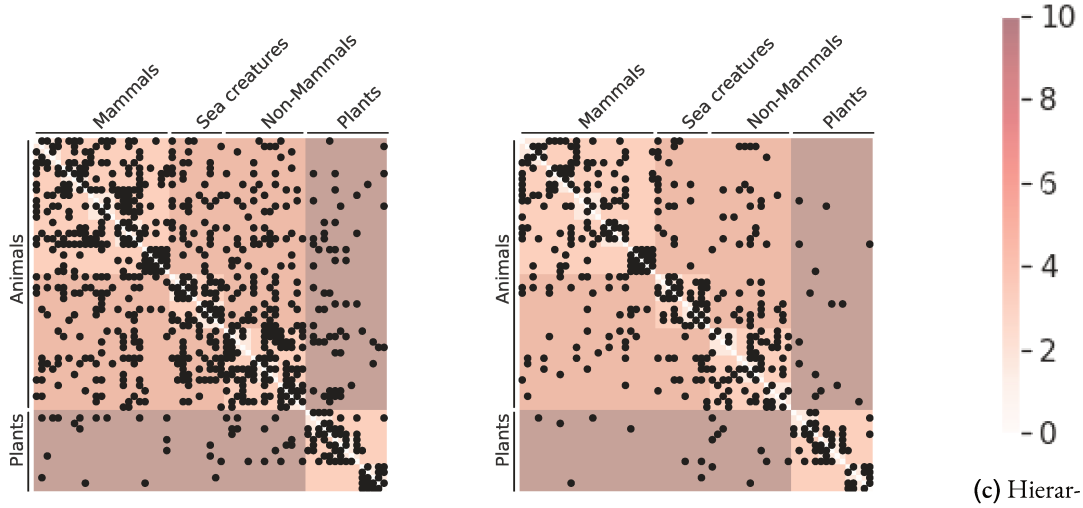


Figure 4.6: Experimental results. Error Rate (ER) in % and Average Hierarchical Cost (AHC) on four datasets for `Guided-proto`, the `Cross-Entropy` baseline (in bold), and competing approaches. Methods that use the hierarchical knowledge are indicated with the symbol ⚜ . The best performances on each dataset are plotted in green. Our guided prototype approach improves both the ER and AHC across the four datasets compared to the baseline. The metrics are computed with the median over 5 runs for CIFAR100, the average over 5 cross-validation folds for S2-Agri, and a single run for NYUDv2 and iNat-19. The numeric values are given in the Table 4.2. ($\star$: not evaluated).

Prototype Learning. We observe that the learnt prototype approach `Learnt-proto` consistently outperforms the `Deep-NCM` method. This suggests that defining prototypes as the centroids of their class representations might actually be disadvantageous. As illustrated on Figure 4.1, the positions of the embeddings tend to follow a Voronoi partition[36] with respect to the learnt prototypes of their true class rather than prototypes being the centroid of their associated representations. A surprising observation for us is that `Learnt-proto` consistently outperforms the `Cross-entropy` baseline, both in terms of AHC and ER.



(a) Cross-Entropy **(b)** Metric-Guided Prototypes **(c)** Hierarchical cost

Figure 4.7: Impact of our method on class confusion. Partial confusion matrix for the "living organism" class subset of CIFAR100 for the Cross-Entropy baseline (a) and our approach (b). For readability, we only display (in black) entries of the matrices with at least one confusion. We also represent the cost of confusing different classes in shades of reds (c). We note that our approach yields fewer confusions between pairs of classes with high costs, such as plants and animals.

Computational Efficiency. Computing distances between representations and prototypes is comparable in terms of complexity than computing a linear mapping. The scaling factor in $\mathcal{L}_{\text{disto}}$ can be efficiently obtained as described in Section 4.1.3.2. In practice, we observed that both training and inference time are identical for `Cross-Entropy` and `Guided-proto`: most of the time is taken by the computation of the embeddings.

Evolution of Optimal Scaling. In Figure 4.8, we represent the evolution of the scaling factor s^* in $\mathcal{L}_{\text{disto}}$ during training of our guided prototype method on the four datasets. Across all four models, s^* presents a decreasing trend overall, which signifies that the average distance between prototypes increases. This is consistent with our analysis of prototypical networks: as the feature learning network and the prototypes are jointly learned, the samples' representations get closer to their true class' prototype. In doing so, they repel the other prototypes, which translate into an *inflation* of the global

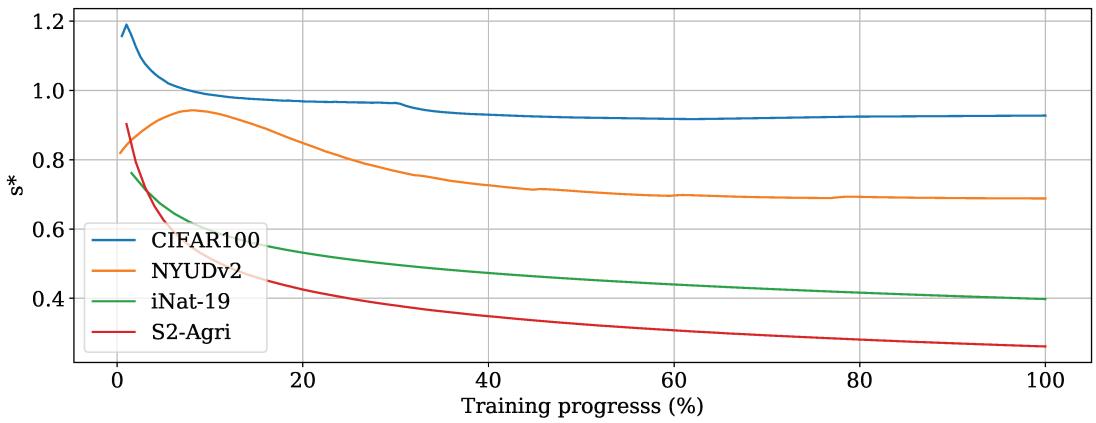


Figure 4.8: Scaling factor. Evolution of the scaling factor s^* in $\mathcal{L}_{\text{disto}}$ along the training iterations of the four networks. We observe that s^* consistently decreases to values smaller than 1, which allow the prototypes to spread apart while respecting the fix distances defined by D.

scale of the problem. Our optimal scaling allows the prototypes' scale to expand accordingly. Without adaptive scaling, the data loss (4.3) and regularizer (4.11) would conflict.

In all our experiments, this scale remained bounded and did not diverge. This can be explained by the fact that for each misclassification $k \to l$ of a sample x_n, the representation $f(x_n)$ is by definition closer to the erroneous prototype π_l than of the true prototype π_k. The first term of $\mathcal{L}_{\text{data}}$ pushes the true prototype π_k towards $f(x_n)$, and by transitivity—towards the erroneous prototype π_l. This phenomenon prevents prototypes from being pushed away from one another indefinitely. However, if the prediction is too precise, *i.e.*, most samples are correctly classified, the prototypes may diverge. This setting, which we have not yet encountered, may necessitate a regularisation such as weight decay on the prototypes' parameters.

Lastly, we remark that the asymptotic optimal scalings are different from one dataset to another. This can be explained foremost by differences in the depth and density of the class hierarchy of each dataset, as presented in Table 4.1. As explained above, the inherent difficulty of the classification tasks may also have an influence on the problem's scale. However, our parameter-free method is able to automatically find an optimal scaling.

4.2.4.1 Restricted Training Data Regime

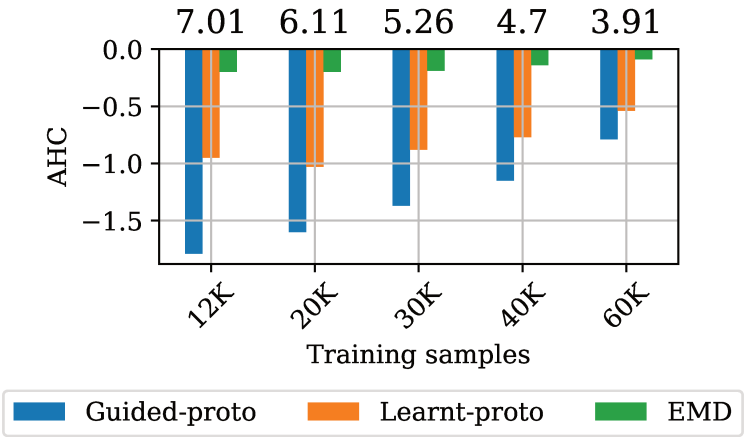


Figure 4.9: Restricted training data experiment. AHC of ResNet-18 trained on restricted training sets of iNaturalist-19 with `Guided-proto`, `Learnt-proto`, and `EMD`. We represent the relative improvement compared to the performance of the `Cross-Entropy` baseline, which is shown on top of the plots.

We observed that the `Learnt-proto` method decreases the AHC across all four datasets even though it does not take the cost matrix into account. This suggests that, given enough data, this simple model can learn an empirical taxonomy through its prototypes' arrangement. Furthermore, this taxonomy can share enough similarity with the one designed by experts to result in a decrease in AHC. To further evaluate the benefit of explicitly using the expert taxonomy with our approach, we train the models `Learnt-proto`, `Guided-proto`, and `EMD` with only part of the 160k images in the training set of iNat-19, and without pretraining on ImageNet. To compensate for the lack of data, we increase the regularisation strength to $\lambda = 20$.

In Figure 4.9, we observe that the two prototype-based approaches consistently improve the performance of the baseline for all training set sizes in terms of AHC. Moreover, the advantages brought by our proposed regularisation are all the more significant when applied to small training sets. This observation reinforces the idea that the `learnt-proto` method requires large amounts of data to learn

a meaningful class hierarchy in an unsupervised way.

4.2.4.2 Ablation Study

Table 4.2: Numerical values of the experimental results. Error Rate (ER) in % and Average Hierarchical Cost (AHC) on three datasets for our proposed method (top) and the competing approaches (bottom). The values are computed with the median over 5 runs for CIFAR100, the average over 5 cross-validation folds for S2-Agri, and a single run for NYUDv2 and iNat-19. (HSP: Hyperspherical Prototypes, GP: Guided Prototypes).

	CIFAR100		NYUDv2		S2-Agri		iNat-19	
	ER	AHC	ER	AHC	ER	AHC	ER	AHC
Cross-Entropy	24.2	1.160	32.7	1.486	19.4	0.699	40.9	1.993
HXE	24.1	1.168	32.4	1.456	19.5	0.731	41.8	2.013
Soft-label	23.5	**1.046**	32.4	**1.424**	19.2	0.703	52.8	2.029
XE+EMD	24.5	1.196	33.3	1.498	19.0	0.687	40.1	1.893
YOLO	26.2	1.214	**32.0**	**1.425**	19.1	**0.685**	42.0	1.942
HSP	29.4	1.472	49.7	2.329	-	-	42.4	2.027
Deep-NCM	25.6	1.249	33.5	1.498	19.4	0.702	40.8	1.929
Free-proto	23.8	1.091	32.5	1.462	19.1	0.691	**38.8**	**1.728**
Fixed-proto	24.7	1.083	33.1	1.462	19.4	0.710	43.9	2.148
GP-rank	**23.3**	**1.056**	32.7	1.445	19.1	0.691	39.3	**1.718**
GP-disto	23.6	**1.052**	32.5	1.440	**18.9**	**0.685**	**38.9**	**1.721**

Scale-Free Distortion. Our method for automatically choosing the best scale in our smooth distortion surrogate leads to an improvement of 0.9 ER on the iNat-19 dataset, which amounts to half the improvement compared to the baseline. In the other datasets, the improvements were more limited. We attribute the impact of our scale-free distortion on iNat-19 in particular to the structure of its class hierarchy: at the lowest level, iNat-19 classes have on average 14 co-hyponyms (siblings), compared to only 2 to 5 for the other datasets. When minimizing the distortion with a fixed scale of 1, the prototypes of hyponyms are incentivised to be close with respect to d since hyponyms have a small hierarchical distance of 2. This clashes with the minimisation of the second part of $\mathcal{L}_{\text{data}}$ as defined in

Table 4.3: Ablation study. Influence of the choice of scaling in $\mathcal{L}_{\text{disto}}$, metric guiding regularizer, and distance function d on the performance of Guided-proto on the four datasets. For d, we compare the performance of the Euclidean norm, the pseudo-Huberised Euclidean norm, and the square Euclidean norm.

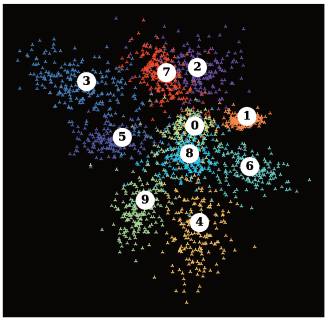
	CIFAR100		NYUDv2		S2-Agri		iNat-19	
	ER	AHC	ER	AHC	ER	AHC	ER	AHC
Guided-proto	23.6	**1.052**	32.5	1.440	**18.9**	**0.685**	**38.9**	1.721
Fixed-scale	+0.1	+0.003	0.0	0.000	+0.2	**+0.001**	+0.9	0.000
Fixed-proto	+1.1	+0.031	+0.6	+0.013	+0.5	+0.025	+5.0	+0.427
Rank-based guiding	**-0.3**	+0.004	+0.2	+0.005	+0.2	+0.006	+0.4	**-0.003**
Pseudo-Huber	+0.1	+0.015	**-0.3**	**-0.017**	+0.4	+0.016	+0.2	+0.003
Squared Norm	+1.0	+0.118	0.0	+0.005	+0.6	+0.022	+2.2	+0.233

(4.3), which mutually repels prototypes of different classes. This conflict, made worse by classes with many hyponyms, is removed by our scale-free distortion.

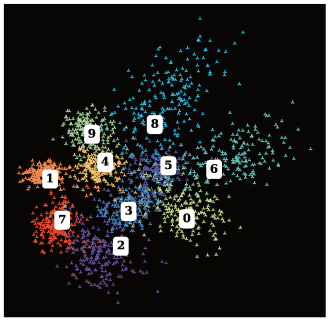
Choice of Metric Space. Prototypical networks operating on $\Omega = \mathbb{R}^m$ typically use the squared Euclidean norm in the distance function, motivated by its quality as a Bregman divergence [147]. However, given the large distance between prototypes induced by our regularisation, this metric can cause stability issues. We observe for all datasets that that defining d as the Euclidean norm yields significantly better results across all datasets.

Guided vs. fixed prototypes. As suggested by the lower performance of `Hyperspherical-proto`, jointly learning the prototypes and the embedding network can be advantageous. To confirm this observation, we altered our `Guided-proto` method to first learn the prototypes and then the embedding network. We observed a significant decrease in performance across the board, up to 5 more points of ER in iNat-19. This suggests that insights from the data distribution can conversely benefit the positioning of prototypes, and that they should be learned conjointly.

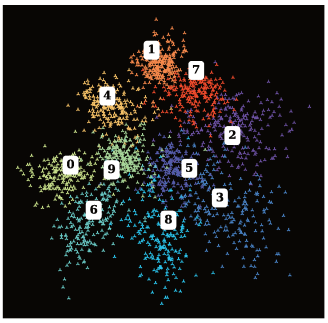
Choice of distance. In Table 4.3, we report the performance of the Guided-proto model on the four datasets when replacing the Euclidean norm with the squared Euclidean norm. Across our ex-



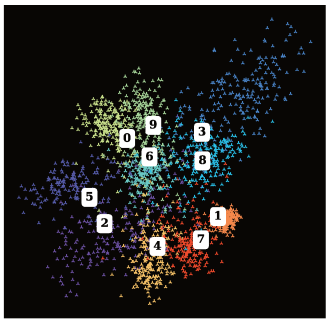
(a) Cross entropy, ER= 15.2%
$\text{disto}_{\text{vis}} = 0.47$, $\text{disto}_{\text{abs}} = 0.61$
$\text{AHC}_{\text{vis}} = 0.81$, $\text{AHC}_{\text{abs}} = 0.65$



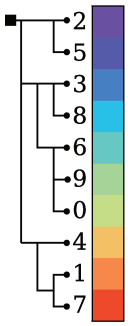
(b) Learnt prototypes, ER= 14.2%
$\text{disto}_{\text{vis}} = 0.42$, $\text{disto}_{\text{abs}} = 0.58$
$\text{AHC}_{\text{vis}} = 0.75$, $\text{AHC}_{\text{abs}} = 0.50$



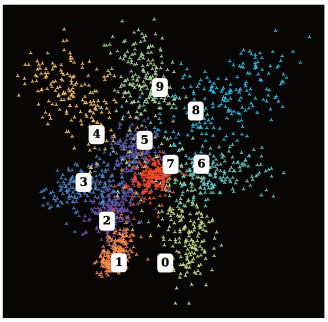
(c) Guided prototypes, ER= 12.8%
$\text{disto}_{\text{vis}} = 0.22$ $\text{AHC}_{\text{vis}} = 0.56$



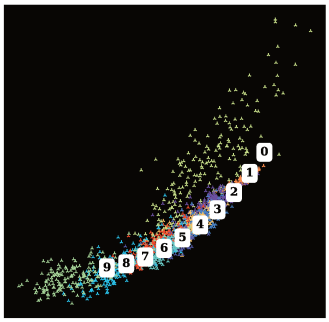
(d) Fixed prototypes, ER= 21.5%
$\text{disto}_{\text{vis}} = 0.17$ $\text{AHC}_{\text{vis}} = 0.82$



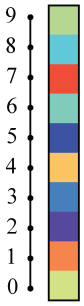
(e) Visual similarity hierarchy.



(f) Guided prototypes, ER= 16.9%
$\text{disto}_{\text{abs}} = 0.24$ $\text{AHC}_{\text{abs}} = 0.54$



(g) Fixed prototypes, ER= 48.8%
$\text{disto}_{\text{abs}} = 0.00$ $\text{AHC}_{\text{abs}} = 0.80$



(h) Numerical order hierarchy.

Figure 4.10: Full illustrative example on MNIST. Mean class representation ○, prototypes □, and 2-dimensional embeddings ⅄ learnt on perturbed MNIST by a 3-layer convolutional net with six different classification modules: (a) cross-entropy, (b) learnt prototypes, (c) learnt prototypes guided by a visual taxonomy, (d) fixed prototypes from a visual taxonomy , (f) learnt prototypes guided by the numbers' values, and (g) fixed prototypes from the numbers' values. The visual hierarchy is represented in (e) and the numerical order in (h). AHC_{vis} corresponds to the cost defined by our proposed visual hierarchy, while AHC_{abs} is defined after the chain-like structure obtained when organizing the digits along their numerical values. While embedding the metric with prototypes prior to learning the representations leads to lower (scale-free) distortion, this translates into worst performance in terms of AHC and ER. Joint learning achieves better performance on both evaluation metrics. We also remark that when the hierarchy is arbitrary (f-g), metric guiding is detrimental to precision.

periments, the squared-norm based model yields a worse performance. This is a notable result as it is the distance commonly used in most prototypical networks [147,52].

Rank-based Regularisation. Mettes et al. [104] use a rank-based loss [19] to encourage prototype mappings whose pairwise distance follows the same order as an external qualification of errors D. Following theirs ideas, we also experiment with a RankNet-inspired loss [19] which encourages the distances between prototypes to follow *the same order* as the costs between their respective classes, without imposing a specific scaling:

$$\mathcal{L}_{\text{rank}}(\pi) = -\frac{1}{|\mathcal{T}|} \sum_{k,l,m \in \mathcal{T}} \bar{\mathbf{R}}_{k,l,m} \cdot \log(R_{k,l,m}) + (1 - \bar{\mathbf{R}}_{k,l,m}) \cdot \log(1 - R_{k,l,m}) \,, \tag{4.15}$$

with $\mathcal{T} = \{(k,l,m) \in \mathcal{K}^3 \mid k \neq l, l \neq m, k \neq m\}$ the set of ordered triplet of $\mathcal{K}$, $\bar{\mathbf{R}}_{k,l,m}$ the hard ranking of the costs between $D_{k,l}$ and $D_{k,m}$, equal to 1 if $D_{k,l} > D_{k,m}$ and 0 otherwise, and $R_{k,l,m} = \text{sigmoid}(d(\pi_k, \pi_l) - d(\pi_k, \pi_m))$ the soft ranking between $d(\pi_k, \pi_l)$ and $d(\pi_k, \pi_m)$. For efficiency reasons, we sample at each iteration only a S-sized subset of $\mathcal{T}$. We use $S = 10$ in our experiments.

We argue that our formulation of $\mathcal{L}_{\text{disto}}$ provides a stronger supervision than only considering the order of distances, and allows the prototypes to find a more profitable arrangement in the embedding space. In Table 4.3, we observe that replacing our distortion-based loss by a rank-based one results in a slight decrease of overall performance.

Robustness. As shown in Table 4.4, our presented method has low sensitivity with respect to regularisation strength: models trained with λ ranging from 0.5 to 3 yield sensibly equivalent performances. Choosing $\lambda = 1$ seems to be the best configuration in terms of AHC.

Table 4.4: Hyperparameter robustness. Robustness assessment of guided prototypes on CIFAR100 (left) and S2-Agri (right). The top line is our chosen hyper-parameter configuration.

	CIFAR100		S2-Agri	
	ER	AHC	ER	AHC
Guided-proto $\lambda = 1$, hidden proto,	23.6	**1.052**	**18.9**	**0.685**
$\lambda = 0.5$	**-0.2**	+0.015	+0.5	+0.019
$\lambda = 2$	+0.3	+0.013	+0.2	+0.010
$\lambda = 3$	+0.1	+0.004	+0.1	+0.010
leaf proto only	+0.2	+0.015	+0.3	+0.011

Hidden prototypes. In cases where the cost matrix D is derived from a tree-shaped class hierarchy, it is possible to also learn prototypes for the internal nodes of this tree, corresponding to super-classes of leaf-level labels. These prototypes do not appear in $\mathcal{L}_{\text{data}}$, but can be used in the prototype penalisation to instill more structure into the embedding space. In Table 4.4, line *leaf-proto*, we note a small but consistent improvement in terms of AHC, resulting in associating prototypes to classes corresponding to the internal nodes of the tree hierarchy as well.

4.3 Conclusion

We introduced a new regularizer modeling the hierarchical relationships between the classes of a nomenclature. This approach can be incorporated into any classification network at no computational cost and with very little added code. We showed that our method consistently decreases the average hierarchical cost of three different backbone networks on different tasks and four datasets. Furthermore, our approach can reduce the rate of errors as well. In contrast to most recent works on hierarchical classification, we showed that this joint training is beneficial compared to the staged strategy of first positioning the prototypes and then training a feature extracting network.

In the context of crop type mapping, our metric guided prototypes[41] can be leveraged to reduce the hierarchical cost of parcel-based or pixel-based crop type classification. We believe that by reducing the severity of erroneous predictions, this can facilitate the adoption of deep learning methods for automated crop type mapping.

There is no real ending. It's just the place where you stop the story.

Frank Herbert

5

Conclusion

In this last chapter, we present our concluding remarks. We wish to reward the reader who made it to here with a refreshing perspective on the work we presented, while also providing a good landing ground to the reader who likes jumping to conclusions.

Table 5.1: Summary of results. We report the performances of all architectures presented in this dissertation, evaluated on the PASTIS dataset. When relevant, we report the semantic performances (OA and mIoU) computed at object level and at pixel level (resp. X_{obj} and X_{pix}), and the panoptic metrics for the panoptic segmentation methods. We also report the performance of the state-of-the-art method prior to this thesis.

	OA_{obj}	$mIoU_{obj}$	OA_{pix}	$mIoU_{pix}$	SQ	RQ	PQ	#Param
Parcel Classification								
LSTM [130]	84.2	56.8	89.5	66.2	-	-	-	1 458k
PSE+TAE (1.4)	91.4	73.0	95.2	82.4	-	-	-	214k
PSE+LTAE (1.5)	91.2	73.9	95.7	85.1	-	-	-	114k
PSE+LTAE + Fusion (3.2)	92.7	77.2	96.7	88.2	-	-	-	287k
Semantic segmentation								
3D-UNet [135]	-	-	81.3	58.4	-	-	-	1 554k
U-TAE (2.1)	-	-	83.2	63.1	-	-	-	1 087k
U-TAE + Fusion (3.2)	-	-	84.2	66.3	-	-	-	1 742k
Panoptic segmentation								
U-TAE + PaPs (2.3)	47.6	34.3	70.7	50.9	81.3	49.2	40.4	1 260k
U-TAE + PaPs + Fusion (3.2)	52.2	35.0	74.8	60.0	82.2	50.6	42.0	1 791k

5.1 Summary

Results. In this dissertation, we developed deep learning methods to learn representations of satellite image time series and predict agricultural maps. Specifically, we addressed crop mapping successively as parcel classification, semantic segmentation, and panoptic segmentation. For each of these settings, our analysis of the specificities of the data and task, and of the recent developments in the deep learning literature, helped us significantly improve the state-of-the-art. We also defined a new state-of-the-art for panoptic segmentation of satellite image time series, a task which was not yet addressed by the crop mapping community. To conclude with a comprehensive view, we report on Table 5.1 the performance of all architectures we introduced, consistently evaluated on PASTIS. For parcel classification, our Pixel-Set Encoder combined with the Lightweight Temporal Attention Encoder improved the state-of-the-art by 16.9pts of mIoU with 10 times fewer trainable parameters. For

semantic segmentation, our U-Net with Temporal Attention Encoder improved the state-of-the-art by 4.7pts of mIoU. With Parcels-as-Points, we set the first milestone for panoptic segmentation of satellite image time series at 40.4 PQ. We also assessed how these performances could be improved by leveraging obstruction-resilient radar acquisitions through modality fusion models.

Tasks difficulty. Beyond our proposed methods, these results illustrate that the three tasks at hand have inherently different levels of difficulty. The knowledge of the parcel boundaries in parcel-based classification makes this problem the simplest of the three. Indeed, knowing the extent of parcels implies that pixels are already segmented into semantically homogeneous groups and dispenses with the classification of background pixels. In practice, the *pixel-level* metrics of the parcel classification models demonstrate how well the task is addressed: only $\sim 3\%$ of pixels are incorrectly classified by our best fusion model. In contrast, addressing crop mapping as a semantic segmentation adds the challenge of making consistent predictions across the entire extent of agricultural parcels, and not to confuse agricultural land with surrounding areas, and vice versa. Expectedly, the performance of semantic segmentation is significantly lower. Lastly, our first exploration of panoptic segmentation of satellite image time series outlined the inherent difficulties of this task. Indeed, a valid prediction requires the model to detect the presence of an agricultural parcel, correctly delineate the parcel's shape, and predicting the true crop type. While the semantic predictions of U-TAE+PaPs are mostly correct, instance segmentation remains challenging: correctly detecting the position and dimensions of a parcel, and subsequently predicting a pixel precise mask proves difficult. We believe that this task requires further research efforts to be considered fully solved. Cityscapes[29], a computer vision benchmark of natural images with a similar number of semantic classes as PASTIS, boasts a state-of-the-art performance of 69.6 PQ at the time of writing[24]. While not directly comparable, this score gives a sense of the progress that can be expected in future works.

5.2 Towards large-scale automated crop mapping.

The application of our methods to real world crop mapping at large-scale raises challenges that were not addressed in this dissertation. In this section, we discuss these issues and outline what we believe is within reach using our methods.

Type of mapping. Given the previous discussion on task difficulty, we believe deep learning methods are mature for real life experimentation if the problem can be addressed as parcel classification or semantic segmentation. Parcel classification corresponds to applications where parcel boundaries are known, for example in regions with well established cadaster. Semantic segmentation would correspond to cases where parcel boundaries are not required in the final agricultural map, *e.g.*, for the inventory of crop production over a region. In such cases, we argue for the use of PSE+L-TAE and U-TAE, respectively. For the more complex problem of retrieving parcel boundaries, some progress is still to be made for real life applications. We hope that the release of PASTIS will encourage further explorations of this challenging problem.

Mapping efforts with accessible annotation. Let us first consider the case of a crop mapping effort in a country for which annotations are accessible. This case is closer to the setting evaluated in this dissertation as the model can be both trained and applied to the same region. Yet, in the operational setting, annotations are usually not available for the on-going agricultural year, which poses the challenge of generalizing from past years to the current year. Quinton & Landrieu [123] showed that PSE-LTAE actually performs better when trained on all available historical years, rather than only the year under consideration, as was done in this thesis. Similar results can be expected for U-TAE as it is also based on the L-TAE architecture, hence this challenge does not seem to prevent real life applications of our methods. However, other challenges remain. First, the models should adapt to the variability of the observed satellite image time series across the entire country. Indeed, each region can

have specific climate, terrain, and cultivating practices. Second, while PASTIS focuses on the 18 most common crop types, real world mapping efforts need to cover the complete set of crop types existing in the region of interest, including rare ones. Such rare types are challenging for learning-based methods as fewer training examples are available. Additionally, increasing the number of classes equally increases the difficulty of the classification problem. Yet, we argue that assembling a multi-year and country-scale training dataset would help mitigating these issues, by enabling training on spatially and temporally diverse data and by increasing the number of samples for rare classes. In addition, we argue that the increased variability of country-scale applications can also be addressed by increasing the size of our models. Our attention-based L-TAE showed, indeed, consistent improvements when scaled up (Figure 1.15). In the NLP literature, drastically increasing the size of attention-based models has proven a valid strategy to address ever more complex problems [16], and it could be similarly valid in our setting to adapt to the increased complexity of *large-scale* crop mapping.

Mapping efforts with scarce or no annotations. A more difficult setting is met when ground truth data for the region of study are scarce or nonexistent. In practice, this setting is quite common as only a limited number of countries produce yearly consolidated agricultural maps. Addressing crop mapping in this setting with learning-based methods can imply training models in a region where annotations are available and applying them to the region of interest. Our experiments did not cover this situation, which raises the challenge of domain shift between the input and target space: both the distribution of the observed satellite time series and of crop types can significantly vary from one location to the other. This problem is at the core of on-going research on out of distribution robustness [182], few-shot learning [134,89], and self-supervised pre-training methods [47,172,188] for geographical generalisation of crop mapping models. In these frameworks, neural architectures such as PSE-LTAE and U-TAE act as backbone encoding networks, while geographical generalisation is tackled with a specific learning procedure, *e.g.*, pre-training or few-shot learning. We believe that the performance

demonstrated by our methods for *within distribution* parcel classification and semantic segmentation makes them solid candidates to be used as backbone networks to test methods addressing geographical generalisation.

5.3 Epilogue

Outcomes. This thesis was supported by the French Mapping Agency (IGN) and the French subsidy allocation authority (ASP). The objectives set by these stakeholders are to develop deep learning methods for large-scale crop mapping from SITS and assess the benefit of using optical radar multi-modal time series. The broader aim of this project is to automatize, at least partially, the production of the French LPIS for subsidy allocation. Our methods for crop type classification both at parcel level with PSE-LTAE and at pixel level with U-TAE achieved significant performance improvements compared to previous approaches. In particular, we showed in Table 1.10 that PSE-LTAE outperforms Random Forest classifier by $\sim$ 20pts of mIoU. This highlights the potential gain associated to a shift from the traditional ML methods used in current automated crop type mapping systems such as iota2[66], and SEN4CAP[10]. Since large amount of annotations are available in France, we advocate for ASP to start evaluating our methods for their potential integration into their production line.

Ethics. The advent of deep learning-based analysis of satellite image time series for crop mapping was not disruptive in the sense that automated analysis of remote sensing data can be traced back to several decades ago[79]. Yet, as we have seen in this dissertation, learning representations from satellite time series to predict agricultural maps brought significant gains in terms of classification performance compared to traditional approaches. Deep learning-based approaches combined with present-day satellite imagery sources thus enable monitoring efforts at an unprecedented spatial scale and accuracy. In this regard, this may affect the balance of power in applications involving farmers such as subsidy allocation. Indeed, the increased performance of automated crop mapping provides additional

monitoring capabilities to the subsidy allocating authority. Yet, an over reliance on such automated tools can be detrimental to farmers who may make honest mistakes or for whom the model made erroneous predictions. Hence, deep learning-based agricultural monitoring systems should provide structures for farmers to voice their potential concerns. This could take the form of an elected or randomly picked assembly of farmers involved in the decision making processes of the monitoring system. Additionally, ensuring transparency by open sourcing the involved code would give further guarantees.

References

[1] (2020). PAC FAQ 2020. https://www.manche.gouv.fr/index.php/content/download/44806/314556/file/2020_FAQ_declaration_v3%20def.pdf. Accessed: 2021-09.

[2] (2021). Comity on earth observation satellites, mission databse. http://database.eohandbook.com/database/missiontable.aspx.

[3] Abramov, S., Rubel, O., Lukin, V., Kozhemiakin, R., Kussul, N., Shelestov, A., & Lavreniuk, M. (2017). Speckle reducing for sentinel-1 sar data. *IGARSS*.

[4] Akhtar, N. & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*.

[5] AlQuraishi, M. (2019). AlphaFold at CASP13. *Bioinformatics*.

[6] Avery, G. (1960). Identifying southern forest types on aerial photographs. *USDA Forest Service, Station Paper SE-112, Southeastern Forest Experiment Station, September 1960*.

[7] Ayub, S. & Saini, J. (2011). ECG classification and abnormality detection using cascade forward neural network. *International Journal of Engineering, Science and Technology*.

[8] Bailly, S., Giordano, S., Landrieu, L., & Chehata, N. (2018). Crop-rotation structured classification using multi-source sentinel images and lpis for crop type mapping. *IGARSS*.

[9] Ballas, N., Yao, L., Pal, C., & Courville, A. (2016). Delving deeper into convolutional networks for learning video representations. *ICLR*.

[10] Bellemans, N., Bontemps, S., Defourny, P., Nicola, L., & Malcorps, P. (2021). *ATBD for L4A crop type mapping*. Technical report, SEN4CAP.

[11] Benedetti, P., Ienco, D., Gaetano, R., Ose, K., Pensa, R. G., & Dupuy, S. (2018). M3Fusion: A deep learning architecture for multiscale multimodal multitemporal satellite data fusion. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

[12] Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., & Lord, N. A. (2020). Making better mistakes: Leveraging class hierarchies with deep networks. *CVPR*.

[13] Bomberger, E. H. & Dill Junior, H. (1960). Photointerpretation in agriculture. *American Society of Photogrammetry. Manual of Photographic Interpretation.*

[14] Bourgain, J. (1985). On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*.

[15] Brankatschk, G. & Finkbeiner, M. (2015). Modeling crop rotation in agricultural LCAs—challenges and potential solutions. *Agricultural Systems*.

[16] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *NeurIPS*.

[17] Bullock, D. G. (1992). Crop rotation. *Critical Reviews in Plant Sciences*.

[18] Bulo, S. R., Neuhold, G., & Kontschieder, P. (2017). Loss max-pooling for semantic image segmentation. *CVPR*.

[19] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. *ICML*.

[20] Campos-Taberner, M., García-Haro, F. J., Martínez, B., Sánchez-Ruíz, S., & Gilabert, M. A. (2019). A Copernicus Sentinel-1 and Sentinel-2 classification framework for the 2020+ european common agricultural policy: A case study in València (Spain). *Agronomy*.

[21] Censi, A. M., Ienco, D., Gbodjo, Y. J. E., Pensa, R. G., Interdonato, R., & Gaetano, R. (2021). Spatial-temporal GraphCNN for land cover mapping. *IEEE Access*.

[22] Charbonnier, P., Blanc-Féraud, L., Aubert, G., & Barlaud, M. (1997). Deterministic edge-preserving regularization in computed imaging. *Transactions on Image Processing*.

[23] Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). This looks like that: deep learning for interpretable image recognition. *NeurIPS*.

[24] Chen, L.-C., Wang, H., & Qiao, S. (2020). Scaling wide residual networks for panoptic segmentation. *arXiv*.

[25] Christophe, E., Inglada, J., & Giros, A. (2008). Orfeo toolbox: a complete solution for mapping from high resolution satellite images. *ISPRS Archive*.

[26] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*.

[27] Cloude, S. R. & Pottier, E. (1996). A review of target decomposition theorems in radar polarimetry. *Transactions on Geoscience and Remote Sensing*.

[28] Congalton, R. G., Balogh, M., Bell, C., Green, K., Milliken, J. A., & Ottman, R. (1998). Mapping and monitoring agricultural crops and other land cover in the lower Colorado river basin. *Photogrammetric Engineering and Remote Sensing*.

[29] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The Cityscapes dataset for semantic urban scene understanding. *CVPR*.

[30] De Sa, C., Gu, A., Ré, C., & Sala, F. (2018). Representation tradeoffs for hyperbolic embeddings. *Proceedings of Machine Learning Research*.

[31] Deng, J., Berg, A. C., Li, K., & Fei-Fei, L. (2010). What does classifying more than 10,000 image categories tell us? *ECCV*.

[32] Derksen, D., Inglada, J., & Michel, J. (2018). Spatially precise contextual features based on superpixel neighborhoods for land cover mapping with high resolution satellite image time series. *IGARSS*.

[33] Dong, N. & Xing, E. (2018). Few-shot semantic segmentation with prototype learning. *BMVC*.

[34] Fippin, E. (1907). Relation of soil surveys to crop surveys. *Agronomy Journal*.

[35] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*.

[36] Fortune, S. (1992). Voronoi diagrams and Delaunay triangulations. In *Computing in Euclidean Geometry*. World Scientific.

[37] Garcia-Pedrero, A., Gonzalo-Martin, C., & Lillo-Saavedra, M. (2017). A machine learning approach for agricultural parcel delineation through agglomerative segmentation. *International Journal of Remote Sensing*.

[38] Garioud, A., Valero, S., Giordano, S., & Mallet, C. (2020). On the joint exploitation of optical and SAR satellite imagery for grassland monitoring. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

[39] Garkusha, I. N., Hnatushenko, V., & V, V. (2017). Research of influence of atmosphere and humidity on the data of radar imaging by sentinel-1.

[40] Garnot, V. S. F. & Landrieu, L. (2020). Lightweight temporal self-attention for classifying satellite images time series. *International Workshop on Advanced Analytics and Learning on Temporal Data*.

[41] Garnot, V. S. F. & Landrieu, L. (2021a). Leveraging class hierarchies with metric-guided prototype learning. *BMVC*.

[42] Garnot, V. S. F. & Landrieu, L. (2021b). Panoptic segmentation of satellite image time series with convolutional temporal attention networks. *ICCV*.

[43] Garnot, V. S. F., Landrieu, L., Giordano, S., & Chehata, N. (2019). Time-space tradeoff in deep learning models for crop classification on satellite multi-spectral image time series. *IGARSS*.

[44] Garnot, V. S. F., Landrieu, L., Giordano, S., & Chehata, N. (2020). Satellite image time series classification with pixel-set encoders and temporal self-attention. *CVPR*.

[45] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR*.

[46] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis*. CRC press.

[47] Ghaffari, H. (2021). An efficient method for the classification of croplands in scarce-label regions. *arXiv*.

[48] Giordano, S., Bailly, S., Landrieu, L., & Chehata, N. (2020). Improved crop classification with rotation knowledge using sentinel-1 and-2 time series. *Photogrammetric Engineering & Remote Sensing*.

[49] Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. *AISTATS*.

[50] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[51] Grant, W. (1997). *The common agricultural policy*. Macmillan International Higher Education.

[52] Guerriero, S., Caputo, B., & Mensink, T. (2018). DeepNCM: deep nearest class mean classifiers. *ICLR Worskhop*.

[53] Gupta, S., Arbelaez, P., & Malik, J. (2013). Perceptual organization and recognition of indoor scenes from RGB-D images. *CVPR*.

[54] Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., & Cipolla, R. (2016). Uunderstanding real world indoor scenes with synthetic data. *CVPR*.

[55] Hazirbas, C., Ma, L., Domokos, C., & Cremers, D. (2016). Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. *ACCV*.

[56] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *ICCV*.

[57] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *CVPR*.

[58] He, W. & Yokoya, N. (2018). Multi-temporal Sentinel-1 and-2 data fusion for optical image simulation. *ISPRS Journal*.

[59] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*.

[60] Hoffman, R., Edwards, D., & Eucker, C. (1976). Identifying and measuring crop type using satellite imagery. *Transactions of the ASAE*.

[61] Hou, L., Yu, C.-P., & Samaras, D. (2016). Squared earth mover's distance-based loss for training deep neural networks. *NeurIPS Workshop*.

[62] Huber, P. (1973). Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*.

[63] Ienco, D., Gaetano, R., Dupaquier, C., & Maurel, P. (2017). Land cover classification via multitemporal spatial data by deep recurrent neural networks. *Geoscience and Remote Sensing Letters*.

[64] Ienco, D., Interdonato, R., Gaetano, R., & Minh, D. H. T. (2019). Combining Sentinel-1 and Sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture. *ISPRS Journal*.

[65] Indyk, P., Matoušek, J., & Sidiropoulos, A. (2017). Low-distortion embeddings of finite metric spaces. In *Handbook of Discrete and Computational Geometry*. Chapman and Hall/CRC.

[66] Inglada, J., Arias, M., Tardy, B., Hagolle, O., Valero, S., Morin, D., Dedieu, G., Sepulcre, G., Bontemps, S., & Defourny, P. (2015). Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery. *Remote Sensing*.

[67] Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*.

[68] Jetley, S., Romera-Paredes, B., Jayasumana, S., & Torr, P. (2015). Prototypical priors: From improving classification to zero-shot learning. *BMVC*.

[69] Ji, S., Zhang, C., Xu, A., Shi, Y., & Duan, Y. (2018). 3d convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote Sensing*.

[70] Johnson, C. W., Browden, L. W., & Pease, R. W. (1969). *A system of regional agricultural land use mapping tested against small scale Apollo 9 color infrared photography of the Imperial Valley (California)*. Technical report, United States Department of the Interior, Geological Survey,.

[71] Joshi, N., Baumann, M., Ehammer, A., Fensholt, R., Grogan, K., Hostert, P., Jepsen, M. R., Kuemmerle, T., Meyfroidt, P., & Mitchard, E. T. (2016). A review of the application of optical and radar remote sensing data fusion to land use mapping and monitoring. *Remote Sensing*.

[72] Kanani, S. S. (1979). *Use of remote sensing for agricultural statistics*. PhD thesis, University of Nairobi.

[73] Kaplan, G., Fine, L., Lukyanov, V., Manivasagam, V., Tanny, J., & Rozenstein, O. (2021). Normalizing the local incidence angle in Sentinel-1 imagery to improve leaf area index, vegetation height, and crop coefficient estimations. *Land*.

[74] Karthik, S., Prabhu, A., Dokania, P. K., & Gandhi, V. (2021). No cost likelihood manipulation at test time for making better mistakes in deep networks. *ICLR*.

[75] Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., & Lempitsky, V. (2020). Hyperbolic image embeddings. *CVPR*.

[76] Kingma, D. P. & Ba, J. (2014). ADAM: A method for stochastic optimization. *ICLR*.

[77] Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019). Panoptic feature pyramid networks. *CVPR*.

[78] Kohonen, T. (1995). Learning vector quantization. In *Self-Organizing Maps*. Springer.

[79] Kolm, K. E. & Case, H. (1984). The identification of irrigated crop types and estimation of acreages from landsat imagery. *Photogrammetric Engineering and Remote Sensing*.

[80] Kondmann, L., Toker, A., Rußwurm, M., Unzueta, A. C., Peressuti, D., Milcinski, G., Mathieu, P.-P., Longépé, N., Davis, T., & Marchisio, G. (2021). DENETHOR: The dynamicearthnet dataset for harmonized, inter-operable, analysis-ready, daily crop monitoring from space. *NeurIPS*.

[81] Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., & Androutsopoulos, I. (2015). Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*.

[82] Kourgli, A., Ouarzeddine, M., Oukil, Y., & Belhadj-Aissa, A. (2010). *Land cover identification using polarimetric SAR images*.

[83] Krizhevsky, A. & Hinton, G. (2009). *Learning multiple layers of features from tiny images*. Technical report, University of Toronto.

[84] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *NeurIPS*.

[85] Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *Geoscience and Remote Sensing Letters*.

[86] Kussul, N., Lemoine, G., Gallego, F. J., Skakun, S. V., Lavreniuk, M., & Shelestov, A. Y. (2016). Parcel-based crop classification in ukraine using landsat-8 data and sentinel-1a data. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

[87] Kussul, N., Lemoine, G., Gallego, J., Skakun, S., & Lavreniuk, M. (2015). Parcel based classification for agricultural mapping and monitoring using multi-temporal satellite image sequences. *IGARSS*.

[88] LeCun, Y. & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*.

[89] Li, Y., Shao, Z., Huang, X., Cai, B., & Peng, S. (2021). Meta-fseo: A meta-learning fast adaptation with self-supervised embedding optimization for few-shot remote sensing scene classification. *Remote Sensing*.

[90] Lin, D., Ji, Y., Lischinski, D., Cohen-Or, D., & Huang, H. (2018). Multi-scale context intertwining for semantic segmentation. *ECCV*.

[91] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature pyramid networks for object detection. *CVPR*.

[92] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. *ICCV*.

[93] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv*.

[94] Liu, J., Gong, M., Qin, K., & Zhang, P. (2016). A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *Transactions on Neural Networks and Learning Systems*.

[95] Liu, S., Chen, J., Pan, L., Ngo, C.-W., Chua, T.-S., & Jiang, Y.-G. (2020). Hyperbolic visual embedding learning for zero-shot recognition. *CVPR*.

[96] Long, T., Mettes, P., Shen, H. T., & Snoek, C. G. (2020). Searching for actions on the hyperbole. *CVPR*.

[97] Marie, M., Bermond, M., Madeline, P., & Coinaud, C. (2013). Quelle cartographie de l'utilisation agricole du sol en france en 2010? les apports du recensement parcellaire graphique. *7èmes Journées de Recherches en Sciences Sociales*.

[98] Martinez, J. A. C., La Rosa, L. E. C., Feitosa, R. Q., Sanches, I. D., & Happ, P. N. (2021). Fully convolutional recurrent networks for multidate crop recognition from multitemporal image sequences. *ISPRS Journal*.

[99] Masoud, K. M., Persello, C., & Tolpekin, V. A. (2020). Delineation of agricultural field boundaries from Sentinel-2 images using a novel super-resolution contour detector based on fully convolutional networks. *Remote Sensing*.

[100] McNairn, H., Kross, A., Lapen, D., Caves, R., & Shang, J. (2014). Early season monitoring of corn and soybeans with terrasar-x and radarsat-2. *International Journal of Applied Earth Observation and Geoinformation*.

[101] Meraner, A., Ebel, P., Zhu, X., & Schmitt, M. (2020). Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion. *ISPRS Journal*.

[102] Mercier, A., Betbeder, J., Rumiano, F., Baudry, J., Gond, V., Blanc, L., Bourgoin, C., Cornu, G., Marchamalo, M., & Poccard-Chapuis, R. (2019). Evaluation of Sentinel-1 and 2 time series for land cover classification of forest-agriculture mosaics in temperate and tropical landscapes. *Remote Sensing*.

[103] Mestre-Quereda, A., Lopez-Sanchez, J. M., Vicente-Guijalba, F., Jacob, A. W., & Engdahl, M. E. (2020). Time-series of sentinel-1 interferometric coherence and backscatter for crop-type mapping. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

[104] Mettes, P., van der Pol, E., & Snoek, C. (2019). Hyperspherical prototype networks. *NeurIPS*.

[105] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ICLR Workshop*.

[106] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*.

[107] Mohan, R. & Valada, A. (2021). EfficientPS: Efficient panoptic segmentation. *International Journal of Computer Vision*.

[108] Monserrat, O., Crosetto, M., & Luzi, G. (2014). A review of ground-based sar interferometry for deformation measurement. *ISPRS Journal*.

[109] Nair, V. & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *ICML*.

[110] Nathan Silberman, Derek Hoiem, P. K. & Fergus, R. (2012). Indoor segmentation and support inference from RGBD images. *ECCV*.

[111] Ndikumana, E., Ho Tong Minh, D., Baghdadi, N., Courault, D., & Hossard, L. (2018). Deep recurrent neural network for agricultural classification using multitemporal sar sentinel-1 for camargue, france. *Remote Sensing*.

[112] Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. *ECCV*.

[113] Nickel, M. & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *NeurIPS*.

[114] Nijhawan, R., Sharma, H., Sahni, H., & Batra, A. (2017). A deep learning hybrid CNN framework approach for vegetation cover mapping using deep features. *Signal Image Technology & Internet Based Systems*.

[115] Ofori-Ampofo, S., Pelletier, C., & Lang, S. (2021). Crop type mapping from optical and radar time series using attention-based deep learning. *Remote Sensing*.

[116] Orynbaikyzy, A., Gessner, U., & Conrad, C. (2019). Crop type classification using a combination of optical and radar remote sensing data: a review. *International Journal of Remote Sensing*.

[117] Orynbaikyzy, A., Gessner, U., Mack, B., & Conrad, C. (2020). Crop type classification using fusion of Sentinel-1 and Sentinel-2 data: Assessing the impact of feature selection, optical data availability, and parcel sizes on the accuracies. *Remote Sensing*.

[118] Papadomanolaki, M., Vakalopoulou, M., & Karantzalos, K. (2021). A deep multi-task learning framework coupling semantic segmentation and fully convolutional LSTM networks for urban change detection. *Transactions on Geoscience and Remote Sensing*.

[119] Pelletier, C., Webb, G. I., & Petitjean, F. (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*.

[120] Petitjean, F., Kurtz, C., Passat, N., & Gançarski, P. (2012). Spatio-temporal reasoning for the classification of satellite image time series. *Pattern Recognition Letters*.

[121] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*.

[122] Qin, Y., Ferraz, A., Mallet, C., & Iovan, C. (2014). Individual tree segmentation over large areas using airborne lidar point cloud and very high resolution optical imagery. *IGARSS*.

[123] Quinton, F. & Landrieu, L. (2021). Crop rotation modeling for deep learning-based parcel classification from satellite time series. *arXiv*.

[124] Redmon, J. & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *CVPR*.

[125] Richards, J. A. et al. (2009). *Remote sensing with imaging radar*, volume 1. Springer.

[126] Rieke, C. (2017). Deep learning for instance segmentation of agricultural fields. https://github.com/chrieke/InstanceSegmentation_Sentinel2.

[127] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *MICCAI*.

[128] Roy, D., Panda, P., & Roy, K. (2020). Tree-CNN: a hierarchical deep convolutional neural network for incremental learning. *Neural Networks*.

[129] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., & Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*.

[130] Rußwurm, M. & Körner, M. (2017). Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. *CVPR Workshop*.

[131] Rußwurm, M. & Körner, M. (2018a). Convolutional LSTMs for cloud-robust segmentation of remote sensing imagery. *NeurIPS Workshop*.

[132] Rußwurm, M. & Körner, M. (2018b). Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS Archive*.

[133] Rußwurm, M. & Körner, M. (2020). Self-attention for raw optical satellite time series classification. *ISPRS Journal*.

[134] Rußwurm, M., Wang, S., Korner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. *CVPR Workshops*.

[135] Rustowicz, R., Cheong, R., Wang, L., Ermon, S., Burke, M., & Lobell, D. (2019). Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. *CVPR Workshops*.

[136] Salakhutdinov, R., Tenenbaum, J. B., & Torralba, A. (2012). Learning with hierarchical-deep models. *Transactions on Pattern Analysis and Machine Intelligence*.

[137] Sato, A. & Yamada, K. (1995). Generalized learning vector quantization. *NeurIPS*.

[138] Sauer, C. O. (1919). Mapping the utilization of the land. *Geographical Review*.

[139] Schuler, D. L., Lee, J.-S., & De Grandi, G. (1996). Measurement of topography using polarimetric sar images. *Transactions on Geoscience and Remote Sensing*.

[140] Segarra, J., Buchaillot, M. L., Araus, J. L., & Kefauver, S. C. (2020). Remote sensing for precision agriculture: Sentinel-2 improved features and applications. *Agronomy*.

[141] Shang, J., Liu, J., Poncos, V., Geng, X., Qian, B., Chen, Q., Dong, T., Macdonald, D., Martin, T., Kovacs, J., et al. (2020). Detection of crop seeding and harvest through analysis of time-series sentinel-1 interferometric sar data. *Remote Sensing*.

[142] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *NeurIPS*.

[143] Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. *CVPR*.

[144] Silla, C. N. & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*.

[145] Simons, M. & Rosen, P. (2007). Interferometric synthetic aperture radar geodesy. *Treatise on Geophysics - Geodesy*.

[146] Singhroy, V. & Saint-Jean, R. (1999). Effects of relief on the selection of radarsat-1 incidence angle for geological applications. *Canadian Journal of Remote Sensing*.

[147] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *NeurIPS*.

[148] Srikanth, P., Ramana, K., Deepika, U., Chakravarthi, P. K., & Sai, M. S. (2016). Comparison of various polarimetric decomposition techniques for crop classification. *Journal of the Indian Society of Remote Sensing*.

[149] Srivastava, H. S., Patel, P., & Navalgund, R. R. (2006). Application potentials of synthetic aperture radar interferometry for land-cover mapping and crop-height estimation. *Current Science*.

[150] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*.

[151] Srivastava, N. & Salakhutdinov, R. R. (2013). Discriminative transfer learning with tree-based priors. *NeurIPS*.

[152] Steinhausen, M. J., Wagner, P. D., Narasimhan, B., & Waske, B. (2018). Combining Sentinel-1 and Sentinel-2 data for improved land use and land cover mapping of monsoon regions. *International Journal of Applied Earth Observation and Geoinformation*.

[153] Stoian, A., Poulain, V., Inglada, J., Poughon, V., & Derksen, D. (2019). Land cover maps production with high resolution satellite image time series and convolutional neural networks: adaptations and limits for operational systems. *Remote Sensing*.

[154] Sudmanns, M., Tiede, D., Augustin, H., & Lang, S. (2020). Assessing global sentinel-2 coverage dynamics and data availability for operational Earth Observation (EO) applications using the eo-compass. *International Journal of Digital Earth*.

[155] Sung, K.-K. (1996). Learning and example selection for object and pattern detection. *MIT A.I.*

[156] Tamm, T., Zalite, K., Voormansik, K., & Talgre, L. (2016). Relating sentinel-1 interferometric coherence to mowing events on grasslands. *Remote Sensing*.

[157] Tarchi, D., Casagli, N., Fanti, R., Leva, D. D., Luzi, G., Pasuto, A., Pieraccini, M., & Silvano, S. (2003). Landslide monitoring by using ground-based sar interferometry: an example of application to the tessina landslide in italy. *Engineering Geology*.

[158] Tarchi, D., Ohlmer, E., & Sieber, A. (1997). Monitoring of structural changes by radar interferometry. *Journal of Research in Nondestructive Evaluation*.

[159] Tarpanelli, A., Santi, E., Tourian, M. J., Filippucci, P., Amarnath, G., & Brocca, L. (2018). Daily river discharge estimates by merging satellite optical sensors and radar altimetry through artificial neural network. *Transactions on Geoscience and Remote Sensing*.

[160] Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*.

[161] Tison, C., Tupin, F., & Maître, H. (2007). A fusion scheme for joint retrieval of urban height map and classification from high-resolution interferometric sar images. *Transactions on Geoscience and Remote Sensing*.

[162] Tokmakov, P., Schmid, C., & Alahari, K. (2019). Learning to segment moving objects. *International Journal of Computer Vision*.

[163] Tom, M. (1997). *Machine Learning*. McGraw Hill.

[164] Tom, M., Jiang, Y., Baltsavias, E., & Schindler, K. (2021). Learning a sensor-invariant embedding of satellite data: A case study for lake ice monitoring. *CoRR*.

[165] Tomás, R., García-Barba, J., Cano, M., Sanabria, M. P., Ivorra, S., Duro, J., & Herrera, G. (2012). Subsidence damage assessment of a gothic church using differential interferometry and field data. *Structural Health Monitoring*.

[166] Trevor, H., Robert, T., & Jerome, F. (2009). *Elements of Statistical Learning*. Springer.

[167] Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*.

[168] Tupin, F., Maitre, H., Mangin, J.-F., Nicolas, J.-M., & Pechersky, E. (1998). Detection of linear features in sar images: Application to road network extraction. *Transactions on Geoscience and Remote Sensing*.

[169] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., & Belongie, S. (2018). The iNaturalist species classification and detection dataset. *CVPR*.

[170] Van Tricht, K., Gobin, A., Gilliams, S., & Piccard, I. (2018). Synergistic use of radar Sentinel-1 and optical Sentinel-2 imagery for crop mapping: a case study forBelgium. *Remote Sensing*.

[171] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *NeurIPS*.

[172] Vincenzi, S., Porrello, A., Buzzega, P., Cipriano, M., Fronte, P., Cuccu, R., Ippoliti, C., Conte, A., & Calderara, S. (2021). The color out of space: learning self-supervised representations for Earth Observation imagery. *International Conference on Pattern Recognition*.

[173] Vrieling, A., Meroni, M., Darvishzadeh, R., Skidmore, A. K., Wang, T., Zurita-Milla, R., Oosterbeek, K., O'Connor, B., & Paganini, M. (2018). Vegetation phenology from Sentinel-2 and field cameras for a dutch barrier island. *Remote Sensing of Environment*.

[174] Vuolo, F., Neuwirth, M., Immitzer, M., Atzberger, C., & Ng, W.-T. (2018). How much does multi-temporal Sentinel-2 data improve crop type classification? *International Journal of Applied Earth Observation and Geoinformation*.

[175] Wagner, F. H., Dalagnol, R., Tarabalka, Y., Segantine, T. Y., Thomé, R., & Hirye, M. (2020). U-net-id, an instance segmentation model for building extraction from satellite images—case study in the Joanópolis city, Brazil. *Remote Sensing*.

[176] Waldner, F. & Diakogiannis, F. I. (2020). Deep learning on edge: extracting field boundaries from satellite images with a convolutional neural network. *Remote Sensing of Environment*.

[177] Wang, C., Zhang, G., & Grosse, R. (2020a). Picking winning tickets before training by preserving gradient flow. *ICLR*.

[178] Wang, Y., Xu, Z., Shen, H., Cheng, B., & Yang, L. (2020b). Centermask: single shot instance segmentation with point representation. *CVPR*.

[179] Wardlow, B. D. & Egbert, S. L. (2008). Large-area crop mapping using time-series modis 250m NDV data: An assessment for the us central great plains. *Remote Sensing of Environment*.

[180] West, N. (1950). Mapping from the air. *Dansk Skovforeningens Tidsskrift*.

[181] Wu, Y. & He, K. (2018). Group normalization. *ECCV*.

[182] Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., & Liang, P. (2021). In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. *ICLR*.

[183] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *NeurIPS*.

[184] Yamaguchi, Y., Moriyama, T., Ishido, M., & Yamada, H. (2005). Four-component scattering model for polarimetric sar image decomposition. *Transactions on Geoscience and Remote Sensing*.

[185] Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W., & Yu, Y. (2015). HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. *ICCV*.

[186] Yang, H.-M., Zhang, X.-Y., Yin, F., & Liu, C.-L. (2018). Robust classification with convolutional prototype learning. *CVPR*.

[187] Yu, F., Wang, D., Shelhamer, E., & Darrell, T. (2018). Deep layer aggregation. *CVPR*.

[188] Yuan, Y. & Lin, L. (2020). Self-supervised pre-training of transformers for satellite image time series classification. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

[189] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. *NeurIPS*.

[190] Zhang, M., Lin, H., Wang, G., Sun, H., & Fu, J. (2018). Mapping paddy rice using a convolutional neural network (CNN) with Landsat 8 datasets in the Dongting lake area, China. *Remote Sensing*.

[191] Zhao, T., Niu, H., de la Rosa, E., Doll, D., Wang, D., & Chen, Y. (2018). Tree canopy differentiation using instance-aware semantic segmentation. *ASABE*.

[192] Zhong, L., Gong, P., & Biging, G. S. (2014). Efficient corn and soybean mapping with temporal extendability: A multi-year experiment using Landsat imagery. *Remote Sensing of Environment*.

[193] Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. *arXiv*.