# Deep Learning for Remote Sensing in Precision Agriculture:

## Large-scale and label-efficient methods for satellite image data

Joachim Nyborg

PhD Dissertation

SOLIDUM PETIT IN PROFUNDIS · UNIVERSITAS ARHUSIENSIS

Department of Computer Science
Aarhus University
Denmark

# Deep Learning for Remote Sensing in Precision Agriculture:

## Large-scale and label-efficient methods for satellite image data

A Dissertation
Presented to the Faculty of Natural Sciences
of Aarhus University
in Partial Fulfillment of the Requirements
for the PhD Degree

by
Joachim Nyborg
March 31, 2022

# Abstract

The growing number of Earth-observing satellites has led to an unprecedented availability of satellite images in the recent decade. The high spectral, spatial, and temporal resolutions of this data have enabled many applications in precision agriculture, including crop monitoring, variable nutrient application, disease and pest management, as well as yield prediction. However, effectively making use of the vast amount of remote sensing data requires automatic, accurate, and scalable methods. Recent deep learning methods are promising because of their strong performance and ability to learn directly from the raw data. However, the data-hungry nature of deep neural networks requires large quantities of labeled training data in order to obtain accurate results. This is particularly a challenge in remote sensing, as obtaining labels for satellite images typically requires expert knowledge or in-situ data collection.

In this thesis, we describe several deep learning methods for remote sensing that are particularly label-efficient. We first present a deep learning method that learns to detect clouds in satellite images with only image-level labels as supervision, which are significantly cheaper to obtain than the typical pixel-level labels. With reduced label costs, deep learning methods become more practical for cloud detection, which is a key preprocessing step for most use of satellite images in precision agriculture. Secondly, we present a method to transfer deep crop type classification models from a region where training labels are available to another where they are not. We introduce a technique to estimate the temporal shift of crop growth between two regions, enabling models to generate accurate pseudo-labels for an unlabeled region. These pseudo-labels are then used as supervision to re-train models for classification in new regions. Lastly, we propose an improved representation of time for crop classification models. Our method is based on accumulated temperatures instead calendar days to better capture the climatic variation affecting crop growth, which enables models to generalize across large geographical areas. We demonstrate that our solutions enable large-scale prediction of crop types in Europe, providing essential information for, *e.g.*, crop monitoring and yield prediction. While our focus is on agricultural applications, our large-scale and label-efficient contributions also hold promise for a variety of other remote sensing tasks ranging from environmental monitoring and climate studies to general classification of satellite image time series.

# Resumé

Det voksende antal jordobservations-satellitter har ført til en hidtil uset tilgængelighed af satellitbilleder i det seneste årti. De høje spektrale, spatiale, og temporale opløsninger af disse data har muliggjort mange anvendelser indenfor præcisionslandbrug, herunder afgrødeovervågning, gradueret tildeling af næringsstoffer, sygdoms- og skadedyrshåndtering samt udbytteforudsigelse. Effektiv brug af den enorme mængde af remote sensing data kræver dog automatiske, nøjagtige og skalerbare metoder. De seneste deep learning-metoder er lovende på grund af deres stærke ydeevne og evne til at lære direkte fra de rå data. Men den datahungrende natur af dybe neurale netværk kræver dog store mængder annoterede træningsdata for at opnå nøjagtige resultater. Dette er især en udfordring indenfor remote sensing, da det typisk kræver ekspertviden eller in situ dataindsamling at få annotering til satellitbilleder.

I denne afhandling beskriver vi flere deep learning-metoder til remote sensing, som er særligt annoteringseffektive. Vi præsenterer først en deep learning-metode, der lærer at detektere skyer i satellitbilleder med kun annoteringer på billedniveau som vejledning, hvilke er væsentligt billigere at anskaffe end de typiske annoteringer på pixelniveau. Med reducerede annoteringsomkostninger bliver deep learning-metoder mere praktiske til skydetektering, hvilket er et vigtigt forbehandlingstrin for de fleste anvendelser af satellitbilleder inden for præcisionslandbrug. For det andet præsenterer vi en metode til at overføre dybe afgrøde-klassificeringsmodeller fra en region, hvor annoteringer er tilgængelige, til en anden, hvor de ikke er. Vi introducerer en teknik til at estimere det temporale skift af afgrødevækst mellem to regioner, hvilket gør det muligt for modeller at generere nøjagtige pseudo-annoteringer for en ikke-annoteret region. Disse pseudo-annoteringer bruges derefter som vejledning til at træne modeller til klassificering i nye regioner. Til sidst foreslår vi en forbedret repræsentation af tid for afgrøde-klassificeringsmodeller. Vores metode er baseret på akkumulerede temperaturer i stedet for kalenderdage for bedre at fange den klimatiske variation, der påvirker afgrødevækst, hvilket gør det muligt for modeller at generalisere på tværs af store geografiske områder. Vi demonstrerer, at vores løsninger muliggør forudsigelse af afgrødetyper i stor skala i Europa, hvilket giver essentiel information til afgrødeovervågning og forudsigelse af udbytte. Mens vores fokus er på landbrugsapplikationer, er vores storskala og annoteringseffektive bidrag også lovende for en række andre anvendelser inden for remote sensing, lige fra miljøovervågning og klimaundersøgelser til generel klassificering af tidsserier af satellitbilleder.

# Acknowledgments

Just as the seed of a crop needs the right nutrients and pest control to grow into a healthy plant, I express my gratitude to all the wonderful people who helped and supported me throughout my PhD journey.

A special thanks to my main supervisor, Ira Assent, for teaching me research skills and helping me navigate the world of academia. Thank you for all your prose feedback in our publications, which I think have greatly improved my writing skills from clumsy to clear and powerful. And thank you for taking on this industrial PhD project, which gave me the opportunity to study a subject I am deeply interested in.

A big thanks also go out to the FieldSense folks for establishing this project in the first place. To John Smedegaard, for serving me a lukewarm beer at Katrinebjerg career day and suggesting this project by chance, and who has since ensured I got immense research freedom during the last 3 years. To my industrial supervisor, Morten Birk, for taking me as your Padawan learner and for being an excellent teacher to GIS, computer vision, and precision agriculture—and for all our excellent idea sparring.

During the last half of my PhD, I've been extremely fortunate to have an online, "bonus" supervisor in France after the coronavirus blocked my chances to go abroad. For this, a huge thanks to Charlotte Pelletier for all our many Zoom meetings. Thank you for your many fruitful ideas and for keeping me on track despite many negative results. And thanks for showing me around beautiful Vannes once I got the chance to go. A big thanks also to Sébastien Lefèvre for our excellent in-depth discussions, and for your expertise and ability to always dig up just the right related work.

Thanks to all my fellow PhD students and colleagues in the Data-Intensive Systems group and FieldSense for going on countless coffee runs with me and for providing a fun and inspiring work environment.

Finally, I thank my friends and family for bearing with me. Thank you, Sille, for your love and support, and for always being there for me through the ups and downs of this roller coaster ride.

*Joachim Nyborg,*
*Aarhus, March 31, 2022.*

# Contents

# Part I

# Overview

# Chapter 1

# Introduction

In modern times, agriculture fulfills one of the most basic needs of humans: food. The history of agriculture started some 100,000 years ago and has since enabled the human population to grow many times larger than was possible by hunting and gathering. New farming techniques have since helped agriculture keep up with the pace of growing food demands, and have simultaneously left many humans free to do other things than to sow fields and harvest crops, such as write computer software for large tech companies or, alternatively, pursue PhDs in computer science.

However, the continuous population growth has put a significant strain on the Earth's limited natural resources. To meet future food demands, the farming process must be optimized to maximize crop yield while minimizing the environmental and economic impact [183]. The field of precision agriculture aims to do just that [9, 25]. Precision agriculture [158] can broadly be defined as the use of emerging technologies, such as remote sensing, the internet of things, big data analysis, and machine learning, to help farmers make informed decisions based on concrete data, rather than their intuition. In particular, the use of satellite images for precision agriculture has rapidly increased in recent years [155]. The open availability of satellite images with high spectral, spatial, and temporal resolutions has enabled many applications for precision agriculture, including crop monitoring [81], determining soil properties [42], variable-rate fertilizer application [83], and yield prediction [190].

In this thesis, we explore the use of deep learning models to analyze satellite images for precision agriculture. Deep learning is a type of machine learning method that in the recent decade has achieved impressive results in for example computer vision (CV) and natural language processing (NLP) tasks [11, 50, 86, 173]. In machine learning, we study algorithms that can acquire their own knowledge about a given task by extracting patterns from raw data, a capability that is known as *learning* [57]. This is typically done by discovering a function that maps a representation of data to some desired output. The performance of machine learning methods depends heavily on the representation of the data they are given. For this reason, most traditional machine learning methods rely on hand-designing the right set of features for a given task.

Deep learning methods differ from traditional machine learning methods in that they learn not only the mapping from representation to output but also the representation itself. Deep learning solves this by using complex models consisting of multiple neural network layers to progressively extract higher-level features from the input. Learning representations means that deep learning methods can operate directly on raw data, removing the need for manual feature design which enables these methods to easily adapt to new tasks. In addition, these learned representations often perform better than hand-designed features [51].

However, a major drawback of deep learning is that large amounts of labeled data must be available for training [162]. This is a central issue when applying these methods to remote sensing data, where unlabeled data is plentiful but collecting high-quality labels is much more challenging [124, 139, 177]. For example, the Sentinel-2 satellites generate about 1.6 terabytes of compressed raw image data every day [29]. But acquiring corresponding labels typically requires experts to manually interpret the images or to physically travel to different areas of the Earth, and *e.g.* survey cultivated crops, which is expensive and not always feasible. Even if extensive efforts are made into gathering high-quality labels, the physical constraint of data collection often results in labels that are localized to specific regions of the Earth. This creates another issue in that training datasets are often not representative of the geographical regions where the deep learning model must be deployed, which causes the model to fail [100, 168]. Consequently, it is important to develop deep learning models for remote sensing which have low labeling costs and which can effectively generalize to new regions after training with few geographically localized labels.

In this thesis, we address these challenges with three machine learning approaches:

1. *Weakly-supervised learning*: How can we create models that can be trained with cheap, low-quality labels but can output high-quality predictions?

2. *Unsupervised domain adaptation*: How can we adapt models trained with data from regions with available labels using unlabeled data from new regions?

3. *Domain generalization*: How can models trained with localized labels generalize to new regions?

Concretely, we consider two tasks in precision agriculture where these issues are particularly prevalent: *cloud detection* and *crop type classification*.

Cloud detection is an essential preprocessing step to detect and mask cloud noise before the satellite images can be used for further analysis in *e.g.* precision agriculture. While deep learning can detect clouds with greater accuracy than other approaches [71], labels are again a significant bottleneck. Obtaining cloud labels requires experts to manually label every pixel of high-resolution satellite images in a large dataset covering different types of clouds and surfaces of the Earth. Moreover, even if labels are obtained, they are specific to only one satellite sensor, meaning this tedious and time-consuming task must be repeated for any future satellites. As a result, weakly-supervised learning is particularly valuable for cloud detection: if we can train accurate deep cloud detection models using cheap, low-quality labels that are easy to acquire, we can benefit from the accuracy of deep learning models without the burden of expensive labels. This leads to our first research questions:

**RQ1:** *How can we train deep cloud detection models with only weak supervision?*

In Chapter 5, we present *Fixed-Point GAN for Cloud Detection*, a weakly-supervised learning method for cloud detection where, instead of requiring labels for every pixel, we only need a label of whether the satellite image contains clouds or not, while still obtaining accurate results at the pixel-level. Our method thus drastically reduces the effort required to obtain cloud labels, as it is much easier to tell if an image is cloudy or not than the exact pixels.

The second task we consider is crop type classification, which is the problem of identifying agricultural parcels and their cultivated crop types in satellite images. Current deep crop classification models use satellite image time series (SITS) to classify crop types based on their unique temporal growth patterns. In particular, crop type maps over wide geographical areas have numerous applications of major financial and environmental importance. For example, knowledge of the cultivated crop types of parcels in Europe is necessary for fair allocation of subsidies to farmers, an endowment of 50 billion euros per year in the European Union [21]. Crop type maps can also help ensure that sustainable crop rotation practices are respected [175] and aid in production forecasts [178]. While large amounts of crop type labels are available for training, such as in some European countries with open data policies [147], the labels are highly localized, which limits the effectiveness of current classifiers on a large scale [80, 100, 168].

We approach this problem with unsupervised domain adaptation, where we adapt a model trained with data from a source region to a different target region using unlabeled target data. This gives rise to our second research question:

**RQ2:** *How can we adapt a trained deep crop classification model to new regions by utilizing unlabeled data?*

In Chapter 6, we present *TimeMatch*, where a crop classifier trained with data from a source region is adapted to a target region by estimating the *temporal shift* between the two regions. This temporal shift is then used to assign pseudo-labels to unlabeled data from the target region and re-train the model. We demonstrate that our approach outperforms existing UDA methods when applied to SITS, which highlights the importance of addressing temporal shifts in the data.

Another approach to large-scale crop classification is domain generalization. The goal of domain generalization is to train a model using data from regions where labels are available that generalize well to regions not seen during training. In comparison to unsupervised domain adaptation, domain generalization does not require any training for the unseen regions, which makes such an approach more practical for large-scale crop classification. Thus, our third research question is:

**RQ3:** *How can we train deep crop classification models that generalize to new regions?*

In Chapter 7, we present *Thermal Positional Encoding*, where we learn crop classification models using *thermal time* [105] to address the problem of temporal shifts discovered in Chapter 6, and thereby improve the generalization. We demonstrate that our approach greatly improves the generalization of crop classifiers on a Europe-wide dataset, thereby enabling practical large-scale crop classification.

## 1.1 Thesis Outline

This thesis is structured into two parts: Part I provides an overview of the research field and an introduction to the papers of this thesis, and Part II contains the full papers.

Part I consists of Chapters 1 to 4. This chapter (Chapter 1) introduces the topic and describes the research questions addressed in the papers. Chapter 2 provides an introduction to satellite images and deep learning, discusses the properties of the data, and provides an overview of the cloud detection pre-processing step. Chapter 3 provides an introduction to large-scale crop classification. We describe current deep learning models, discuss how they address the challenges of the task, and end with a discussion of open questions. Finally, Chapter 4 contains an introduction to the contributions of the papers, including a summary of the results and an assessment of the applied methodologies, and presents directions for future work.

Part II contains publications and manuscripts. Only editing related to formatting has been made to the papers. I was the main contributor to all papers included. In particular, we include the following:

- Chapter 5: **Joachim Nyborg** and Ira Assent (2021). Weakly-Supervised Cloud Detection with Fixed-Point GANs. In *Proceedings of the 2021 IEEE International Conference on Big Data (BigData) Workshops (Machine Learning for Big Data Analytics in Remote Sensing)* [115]. `https://arxiv.org/abs/2111.11879`.

- Chapter 6: **Joachim Nyborg**, Charlotte Pelletier, Sébastien Lefèvre, and Ira Assent (2022). TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation. Revised manuscript submitted to the *ISPRS Journal of Photogrammetry and Remote Sensing* [118]. `https://arxiv.org/abs/2111.02682`.

- Chapter 7: **Joachim Nyborg**, Charlotte Pelletier, and Ira Assent (2022). Generalized Classification of Satellite Image Time Series with Thermal Positional Encoding. Submitted to the *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (EarthVision)* [117]. `https://arxiv.org/abs/2203.09175`.

Other works not included in this thesis:

- **Joachim Nyborg** and Ira Assent (2019). Agricultural Land Cover Classification with Deep Learning. Presented at *Nordic Remote Sensing Conference (NoRSC '19)*.

- Emy Alerskans, **Joachim Nyborg**, Morten Birk, and Eigil Kaas (2021). Prediction of near-surface temperatures using a non-linear machine learning post-processing model (2021). Presented at *EGU General Assembly Conference*.

- Emy Alerskans, **Joachim Nyborg**, Morten Birk, and Eigil Kaas (2021). A Transformer Neural Network for Predicting Near-Surface Temperature. In review for *Meteorological Applications*.

## Chapter 2

# Satellite Image Analysis for Precision Agriculture

In this thesis, we consider the use of remote sensing data to address tasks in precision agriculture. In particular, we focus on *optical* satellite images, that is, satellites that measure the reflectance of light from the Earth's surface, in contrast to *e.g.* radar sensors. In this chapter, we present an introduction to the properties of this type of data and why it is useful in precision agriculture. Then, we give an overview of deep learning and the main challenges with applying existing deep learning models to automate satellite image analysis. Finally, we discuss the application of deep learning for cloud detection to pre-process satellite images.

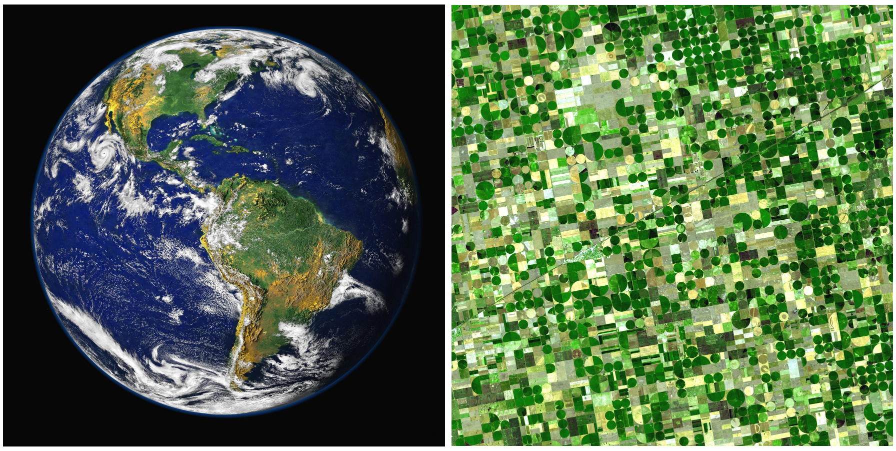


Figure 2.1: Satellite images enable agricultural mapping across the entire Earth. Credit: NASA.

## 2.1 Optical Satellite Image Data

The image quality of optical satellite images has greatly improved since the first image was taken of the Earth in the late 1950s by the Explorer 6 satellite. The improvements in image quality have enabled a broad range of applications, and thanks to the open data policy of many satellite programs, such as Landsat [69] and Sentinel [29], the data is now also available free of charge.

Satellites typically differ based on the spatial, spectral, and temporal resolutions they offer. The spatial resolution is defined as the surface area a single pixel represents. For example, the Landsat-8 and Sentinel-2 satellites, both of which we consider in this thesis, have a relatively coarse spatial resolution of 30m and 10m, respectively, compared to *e.g.* the 2m resolution offered by Planet [123]. In precision agriculture, a high resolution is not always a requirement, as it also means more data to process. A high spectral resolution, however, is key [155]. Compared to regular camera images with three spectral bands (red, green, and blue), satellite images typically provide information beyond the visible spectrum in *e.g.* the near- and short-wave infrared. Near-infrared light is particularly important when studying vegetation. Our eyes see plant leaves as green because the wavelengths in the green region of the spectrum are reflected by plants, while red and blue are absorbed. What we do not see is that near-infrared is reflected even more, and how much is reflected depends on how “healthy” the plant is. If the plant is unhealthy, it instead reflects less near-infrared and more red light, as sick plants typically turn into a brown color.

Because the red and near-infrared bands of satellite images relate to photosynthetic activity, they are highly relevant to analyzing crops in precision agriculture. Thus, a common feature to extract is a ratio between these two bands, also referred to as the normalized difference vegetation index (NDVI) [167]:

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}} \tag{2.1}$$

where Red and NIR are the spectral bands of a satellite image for red and near-infrared, respectively. In precision agriculture, NDVI has been widely used to *e.g.* control variable-rate irrigation and fertilizer systems [155]. In addition to near-infrared, bands in the short-wave infrared region are also highly relevant, as they are connected to the water content in plants and the soil [149].

A high spectral resolution becomes particularly useful for agriculture when combined with a high temporal resolution. The temporal resolution is how frequently a new satellite image is acquired of the same area. For example, the Sentinel-2 program consists of two satellites in the same orbit but phased 180 degrees from each other, which enables a surface area to be revisited every 5 days, and sometimes more often when the images overlap. Combining such satellite images of the same area over time with *satellite image time series* (SITS) makes it possible to study the development of crops over time. This enables the remote sensing study of crop *phenology*, which is the various biological life cycles that characterize each crop type. Computing NDVI from SITS is a way to reveal crop phenologies, as shown in Figure 2.2 for the crop

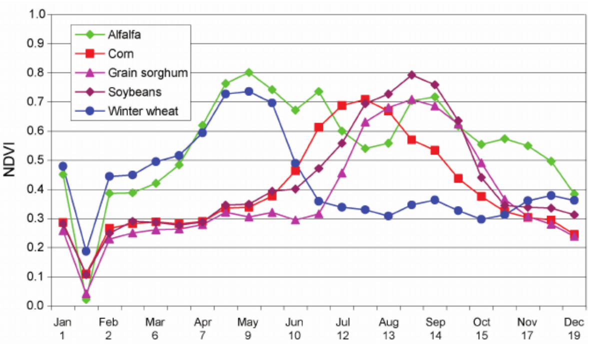


Figure 2.2: Example temporal NDVI profiles of different crop types in Kansas, USA, showing their unique phenology. Image is taken from [103].

types alfalfa, corn, sorghum, soybean, and winter wheat. The figure shows that each crop type has a unique and well-defined profile, as a result of the timing differences of green-up, peak greenness, and senescence. A clear temporal shift separates the peak NDVI of summer crops (corn, sorghum, soybeans) and that of winter crops (alfalfa and winter wheat) in the spring period when the summer crops are yet to be planted. We can also see that winter wheat is the first to be harvested (the NDVI drops), while alfalfa continues to experience "grow and cut" cycles. From this, it is clear that for crop classification from satellite images, we need to understand the unique spectral-temporal profiles of each crop type and the timing differences between them. We discuss the crop classification task further in Chapter 3.

## 2.2 Automatic Satellite Image Analysis

While it is possible to manually interpret satellite images, such as by NDVI, efficiently making use of the vast amount of satellite image data available requires automatic data analysis methods. For this, traditional machine learning methods have been widely applied. Common methods are based on random forests [47] and support vector machines [48]. These methods typically require the extraction of hand-crafted features. For tasks related to precision agriculture, these features include vegetation indices (*e.g*. NDVI) [1, 33, 174, 195], phenological features [55, 77], or additional meteorological data [33]. Although these methods use features that are understandable and robust, the choice of features requires task- or region-specific expert knowledge which limits scalability. In comparison, recent deep learning methods are scalable and often more accurate due to their ability to learn task-specific features directly from the raw satellite image data [125, 138, 142]. Given the scalability and state-of-the-art performance of deep learning at the time of writing this thesis, we focus on these methods for satellite image analysis.

### Overview of Deep Learning

Recent deep learning methods for analyzing satellite images and time series thereof build upon the impressive results of deep learning in computer vision (CV) and natural language processing (NLP).

Following the highly influential AlexNet [86] with groundbreaking results on the ImageNet dataset [26], “deep” neural networks, that is, networks with many neural network layers, have revolutionized many computer vision tasks, including image classification [59, 86, 164], object detection [45, 46, 131], and semantic segmentation [97, 134]. Deep learning models with many layers are computationally expensive, and training is typically made feasible by parallelizing computation using graphics processing units (GPUs). In computer vision, convolutional neural networks (CNNs) [89] are most commonly used, where learnable convolutional filters slide along the height and width of the input image to extract spatial-spectral features in a translation equivariant fashion. Recent advances in this field have been characterized by further increased model sizes, enabled by residual connections to stabilize training with increased depth [59], or neural architecture search to find efficient and large models [164]. Another line of work is Generative Adversarial Networks (GANs) [50], which have shown an impressive ability to generate photo-realistic images [78], transfer the style of one image to another [197], and increasing the resolution of images [90]. A typical GAN consists of two networks: a *generator* and a *discriminator*. The generator learns to generate fake images, while the discriminator learns to distinguish between real and fake images. By training these two networks adversarially, where the two networks contest with each other in a minimax game, the generated images become highly realistic.

Deep learning has also achieved impressive results in many NLP tasks. Compared to CV, the text data of NLP is not spatial but sequential, and this line of work thus focuses on the use of neural network units that process sequences. In NLP, these sequences typically consist of word embeddings [106], obtained by mapping each unique word to a vector representation. Recurrent neural networks (RNNs), such as long short-term memory (LSTM) [62] and gated recurrent units (GRU) [18], have been widely used [18, 52, 163]. Given an input sequence, RNNs output a sequence of hidden states $\boldsymbol{h}^{(t)}$ as a function of the previous hidden state $\boldsymbol{h}^{(t-1)}$ and the input for position $t$. The drawback of RNNs, however, is that this sequential computation limits parallelization and also creates difficulties in modeling long-range time dependencies. The latter is addressed to some degree by bidirectional RNNs [52], but significant benefits were later achieved by the attention mechanism [3], which enables models to compute global dependencies between the inputs and outputs of RNNs. However, the constraint of sequential computation remained. Recently, this problem is addressed with the Transformer model [173], where the recurrent units are completely replaced by a particular attention mechanism called self-attention, where outputs are computed by relating each element in the input sequence itself with all other elements. Self-attention is fully parallelizable, which leads to significantly lower training times compared to RNNs. The efficiency of self-attention has enabled the training of huge

Transformer models to obtain impressive language capabilities, such as the GPT-3 model [11]. Recently, the self-attention mechanism has also been applied to obtain image classification results similar to state-of-the-art convolutional networks [28], indicating that self-attention is not only beneficial for sequences but is a general learning mechanism.

### Differences to Remote Sensing Data

Next, we study how deep learning can be applied to remote sensing data. While we also consider image data and sequences thereof, there are some key differences between the typical data studied in CV and NLP and that in remote sensing which limits the direct use of existing CV+NLP methods in precision agriculture.

For instance, the typical images of CV, such as natural images from ImageNet [26], are hand-held, 3 band camera images (red, green, blue), and the typical objects in the images are large and 3-dimensional. In comparison, Sentinel-2 images are 13 band images, and the typical objects (*e.g.* agricultural fields) are small and 2-dimensional. These differences mean that the best-performing models in CV may not necessarily obtain high performance for remote sensing data. For example, it has been shown that CNNs rely heavily on the texture of objects [44] for image classification, such as the fur texture of cats and dogs. However, this cannot be leveraged for *e.g.* classification of agricultural fields in Sentinel-2 images, as the 10m resolution is not high enough to show any specific plant texture of crops [142]. Another example is that occlusion in natural images happens between objects, whereas in satellite images, occlusion typically only happens between clouds and the objects on the surface (*e.g.* agricultural fields). We leverage this in Chapter 5, where the occlusion difference between clear and cloudy images is used to detect clouds with weak supervision.

Another comparison we make here is between the text sequences of NLP and satellite image time series. However, a crucial difference is the importance of temporal positions. In NLP, the position of a word in a sequence is mostly arbitrary, as words are not more likely to appear at the beginning than the end of the text (to some degree). In contrast, the time position of satellite images is crucial information. For example, the position of an image in SITS refers to a specific instance in time, such as the spring or the summer, and this information is required to model the timing of crop phenology. We leverage this observation in the work we present in Chapter 6, where we use the timing learned by models in one region to adapt the model to another. Furthermore, in Chapter 7, we change the temporal reference from the commonly used calendar time to thermal time, which better accounts for the climatic differences between regions, to improve the generalization of crop classifiers.

### Dealing with Clouds

Making use of the large body of remote sensing data requires the images to be preprocessed to an analysis-ready state, and one of the key steps in that process is *cloud detection*. More than half of the Earth's surface is covered by clouds every day on average [160], and without prior detection of clouds and their accompanying cloud shadows, the availability of satellite images is greatly limited.

Traditionally, cloud detection has been tackled by threshold-based methods [54, 199], where handcrafted thresholds are set to satellite image pixels to separate cloudy pixels from clear pixels. However, it is difficult to scale such an approach to the large variation in cloud types and land surfaces, and much work is required to tackle the many edge cases. For example, with threshold-based methods, it is difficult to detect the semi-transparent cloud-type cirrus without the use of a specialized cirrus band [199]. Another accuracy issue is often with surfaces of high reflectance values, such as mountainous regions, where the use of the elevation maps is needed to accurately detect clouds [127] A challenge is also that threshold-based methods are designed for a specific satellite only, which makes these methods difficult to scale to different satellites.

Recently, deep learning methods based on semantic segmentation have been shown to outperform threshold-based methods [71, 94, 191]. Moreover, as these methods are learnable from data, they can easily be scaled to different satellites as long as a labeled dataset is available to re-train the model for the new image properties. However, if such a dataset is not readily available, manually annotating ground truth for clouds is a highly laborious task. Each pixel of a large and varying satellite image dataset must be manually interpreted as clouds or not, which is time-consuming and can be difficult when dealing with semi-transparent clouds.

In Chapter 5, we present a deep learning method for cloud detection which can be trained with only image-level ground truth labels. That is, instead of labeling every pixel of clouds, our method only requires information about whether the image contains clouds or not, which is much easier to annotate. We demonstrate that our method can predict pixel-level cloud labels with an accuracy similar to existing deep learning methods trained with the expensive pixel-level ground truth, and outperforms threshold-based methods designed for Landsat-8.

## 2.3 Conclusion

In this chapter, we introduce the satellite image data we consider in this thesis and the properties needed for applications in precision agriculture. We describe methods to automatically analyze satellite images. Deep learning methods are considered in this thesis based on their impressive results for image analysis in computer vision and temporal understanding in natural language processing. Additionally, we discuss the main differences between the typical data considered in CV and NLP compared to that in remote sensing which motivates specialized methods. Finally, we provide an overview of cloud detection, a satellite image pre-processing step required for most precision agriculture applications. In the next chapter, we discuss another task for deep learning on satellite images, namely *crop type classification*.

# Chapter 3

# Large-Scale Crop Classification

In this chapter, we introduce the problem of crop classification, which is the task of assigning crop type labels to satellite images. This is one of the core problems in remote sensing that has a large variety of applications—in particular, crop classification over large geographical scales (national, continental, global) is extremely valuable knowledge for many applications, including crop area estimation [10], yield prediction [6, 35], and disaster risk assessment [156]. We present here the different kinds of deep learning models proposed in the literature, and discuss the current state of research, in particular if large-scale crop classification is addressed.

## 3.1 The Crop Classification Problem

Classifying crop types from a single satellite image is not always possible as the appearance of different crops can be very similar at a single time step. Instead, SITS have been the primary data source for crop type classification for decades [119, 132], as temporal data reveals the unique phenology of different crop types (described in Chapter 2).

Formally, we define crop classification as the problem of classifying a sequence of satellite images $\boldsymbol{x}_i = (\boldsymbol{x}_i^{(1)}, \dots, \boldsymbol{x}_i^{(T)})$ of length $T$. The classification can be addressed either parcel-wise or pixel-wise. We illustrate these two approaches in Figure 3.1.

In parcel-wise classification, only pixels from a homogeneous agricultural plot of land, or a *parcel*, are input to the model. In this case, each input $\boldsymbol{x}^{(t)} \in \mathbb{R}^{T \times H \times W \times C}$ contains the pixels within the parcel bounding box of height $H$ and width $W$, which is then mapped to a prediction $y \in \mathbb{R}^K$ to $K$ crop classes. With this approach, the model only has to classify pixels within parcels and does not have to deal with other types of non-agricultural land such as cities or water. However, this approach requires the polygon shapes that outline parcels to be available. While these are typically available for European countries [147], they may not be for many other regions of the world.

In pixel-wise classification, polygon shapes are not required. With deep learning, semantic segmentation is usually performed instead of classifying one pixel at a time [41, 138, 140], where a SITS patch $\boldsymbol{x}^{(t)} \in \mathbb{R}^{T \times H \times W \times C}$ is classified by predicting

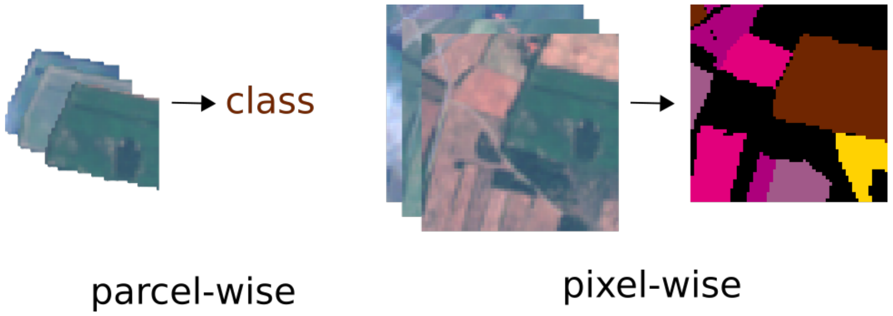


Figure 3.1: Crop classification of SITS with deep learning is either performed parcel-wise, which requires knowledge of the parcel borders, or pixel-wise, which does not. For the latter, we show here a semantic segmentation approach, where a satellite image time series is mapped to a segmentation image containing pixel-wise predictions.

a segmentation map $y \in \mathbb{R}^{H \times W \times K}$ containing predictions for every pixel. But as these pixels can belong to any type of land, the model must be able to handle non-agricultural pixels, such as cities, water, forest, *etc*. This makes it more difficult to study the challenges that affect crop growth on a large scale. Therefore, we focus on parcel-wise classification in our work presented in Chapter 6 and Chapter 7.

## 3.2 Challenges

As we describe in the next section, current deep learning methods are adept at classifying crop types by recognizing their phenological patterns from SITS. However, there are still some challenges that a deep learning model must address, in particular for large-scale classification. We present here a (non-exhaustive) list of such challenges:

- **Cloud occlusions.** The spectral values of crop development can be randomly occluded by clouds or changed because of cloud shadows.

- **Class imbalance.** Farmers do not cultivate an equal amount of each crop type, causing a significant imbalance in the frequency of different classes.

- **Irregular temporal sampling and length.** Depending on the location, SITS are observed a different number of times and intervals. This is a result of the satellite's orbit and filtering observations with high cloud coverage.

- **Spectral variation.** The spectral growth profile of the same crop type can be deformed or scaled depending on the local topography, soil, and farmer practices.

- **Temporal shift variation.** The growth season can be shifted earlier or later in a year depending on the weather that year and the local climate.

A good crop classification model should be invariant to these variations, and at the same time retain sensitivity to the inter-class variation in phenology to separate the classes. These challenges are even more important for large-scale classification since these variations become even greater at *e.g.* continental-scale compared to local regions.

Next, we describe recent deep learning architectures used for crop classification in the context of these challenges. Prior work has shown the importance of modeling the temporal dimension when classifying crop types [40], as this is how the phenological characteristics are learned. Therefore, we categorize existing methods based on the neural component used to model time—in particular, we consider recurrent, convolutional, and self-attention methods.

## 3.3 Recurrent Methods

Similar to how RNNs have been widely used to model sequences in NLP and other fields, they have also been the natural choice to model SITS for crop classification [65, 108, 112, 137]. The work of Rußwurm and Körner [137] uses long short-term memory (LSTM) networks [62] for pixel-wise crop classification using SITS over a year concatenated with the day of acquisition. The latter provides the model with positional information to help account for the challenge of irregular temporal sampling. Similarly, LSTMs have also been used for semantic segmentation by the use of convolutional LSTM networks (ConvLSTM) [138], where the fully-connected layers of the LSTM are replaced with convolutions [152] to handle the spatial dimensions. LSTMs are particularly well-suited to handling cloud occlusions, as shown in [138]. By visualizing the internal cell activations of the LSTM, it was found that certain gates are closed for cloudy pixels, indicating that the network implicitly learns to ignore clouds. This property is highly valuable for deploying crop classification models in real-world scenarios, as the preprocessing step of cloud detection can be skipped. Moreover, instead of selectively choosing only cloud-free observations, partially cloudy observations can be used, which increases the information available to the model and potentially the accuracy. The work of Rustowicz et al. [140] later improve the segmentation performance by combining the ConvLSTM with the U-Net model [134], a popular segmentation model for computer vision.

While these recurrent methods generally achieve high crop classification accuracy, the sequential nature of recurrent models limits their efficiency. This is especially an issue for large-scale crop classification, where the data to process might span continents. For example, a year of Sentinel-2 data for Europe amounts to about 25TB of images [142], and processing data of this volume with non-parallelizable models is a significant performance bottleneck.

## 3.4 Temporal Convolutional Methods

An alternative to recurrent methods is temporal convolutional methods, where convolutions are applied along the temporal dimension [125, 193]. Compared to recurrent networks, convolutions along time can be computed in parallel, and are thus more scaleable.

The use of temporal convolutions (TempCNN) is proposed by Pelletier et al. [125] for pixel-wise classification. To account for irregular temporal sampling, the SITS are interpolated to regular temporal intervals of 2 days. A notable finding of this work is that the use of pooling layers common in computer vision, such as max pooling and global average pooling, are not beneficial for SITS classification. In convolutional networks for image classification, pooling layers are often used to extract features that are invariant to scale and shifts, such that objects can be detected no matter their location. But for SITS, this is not the case, as the temporal location of features is crucial information that is lost with pooling. For example, winter and spring wheat, which we want to distinguish, may have similar features but with a shift in time. Our work presented in Chapter 6 and Chapter 7 exploits this property for domain adaptation and generalization.

Another approach to temporal convolutions is 3D-convolutions [73, 140], where convolutions handle the spatial and temporal dimensions simultaneously. Other work finds that combinations of convolutions and recurrent layers can be beneficial [67].

While temporal convolutional methods have significantly lower processing times than recurrent methods, it is more difficult for convolutions to ignore cloudy observations [136]. Whereas recurrent methods have internal gates that can control the influence of particular time steps, convolutions will always extract features no matter their relevance, making it more difficult to ignore the irrelevant clouds of SITS.

## 3.5 Self-Attention Methods

Today, self-attention is the core of state-of-the-art NLP models [11, 27], building upon the Transformer model by Vaswani et al. [173]. First proposed for SITS in [136], self-attention brings the best of both recurrent and convolutional networks—it enables a model to select, or “attend” to, the most relevant time steps, thus being able to ignore clouds while at the same time being fully parallelizable. Perhaps attention is all you need. Based on these benefits, we build upon self-attention methods in our work in Chapter 6 and Chapter 7, and provide here the details of the inner workings of these models, how they are applied to SITS, and their current limitations in terms of the challenges outlined in Section 3.2.

**Scaled Dot-Product Attention.** The type of self-attention used in the original Transformer model [173], the scaled dot-product attention, works as follows. Given a sequence input $\boldsymbol{x} \in \mathbb{R}^{T \times D}$ where $T$ is the length and $D$ the feature dimension, three linear transformations of each sequence element in $\boldsymbol{x}$ are computed first:

$$\boldsymbol{Q} = \boldsymbol{x}\boldsymbol{W}_Q, \quad \boldsymbol{K} = \boldsymbol{x}\boldsymbol{W}_K, \quad \boldsymbol{V} = \boldsymbol{x}\boldsymbol{W}_V, \tag{3.1}$$

where $\boldsymbol{Q}$ is referred to as the queries, $\boldsymbol{K}$ the keys, and $\boldsymbol{V}$ the values. The projections are learnable parameters $\boldsymbol{W}_Q \in \mathbb{R}^{D\times D_k}, \boldsymbol{W}_K \in \mathbb{R}^{D\times D_k}, \boldsymbol{W}_V \in \mathbb{R}^{D\times D_v}$. Next, an *attention matrix* $\boldsymbol{A} \in \mathbb{R}^{T\times T}$ is computed by:

$$\boldsymbol{A} = \mathrm{softmax}\left(\boldsymbol{Q}\boldsymbol{K}^\top\right), \tag{3.2}$$

where the softmax is applied row-wise such that each row sums to one. Thus, the $i$-th row of $\boldsymbol{A}$ contains the similarity (dot product) of the query $\boldsymbol{Q}^{(i)}$ of time step $i$ and all keys $\boldsymbol{K}^{(j)}, j = 1, \ldots, T$, re-scaled by the softmax function. In other words, each row indicates how "relevant" all other time steps are to a particular time step. Finally, the output of the self-attention is computed by an attention weighted sum of the values,

$$\boldsymbol{H} = \boldsymbol{A}\boldsymbol{V} \tag{3.3}$$

where the result $\boldsymbol{H} \in \mathbb{R}^{T\times D_v}$ contains an output for each time step. As the primary computations of self-attention are dot-products, it can be implemented using highly optimized matrix multiplication, resulting in significant speed-ups compared to recurrent models. The main drawback of self-attention is its memory requirements: computing $\boldsymbol{A}$ consumes $\mathcal{O}(T^2)$ memory, which is problematic for long input sequences.

**Multi-Head Attention.** Instead of only a single linear transformation of each input element to compute the queries, keys, and values, the *multi-head attention* of the Transformer model [173] applies $h$ linear transformations with different parameters. Then, Equation (3.3) is computed $h$ times in parallel, and the final results are concatenated. This allows each head to specialize and attend to information from different representations at different positions. In addition, this further increases parallelization and thus the efficiency of self-attention.

**Positional Encoding.** Since all computations of self-attention are dot-products between linear transformations of input elements, there is no notion of the order of elements in the input sequence, meaning that a random sequence order would give the same result as the original order. But for most sequences tasks, the order matters—in NLP, for example, the sentence "the student finishes the PhD" does not have the same meaning as "the PhD finishes the student". To provide this information to self-attention, *positional encodings* are used, which map positions to unique vectors that can then be added to the elements of the input sequence. The most common positional encoding is *sinusoidal positional encoding* [173]. For an input element $\boldsymbol{x}^{(t)}$ at the position $t$, it is computed by:

$$\boldsymbol{p}^{(t)} = [\sin(\omega_i t), \cos(\omega_i t)]_{i=1}^{D/2} \tag{3.4}$$

where $\omega_i = (1/10000)^{2i/D}$ are different sinusoidal wavelengths for each dimension. Then, the input to self-attention is changed from $\boldsymbol{x}$ to $\boldsymbol{x} + \boldsymbol{p}$ such that each element contains order information.

**Crop Classification with Self-Attention.** The output of self-attention is a sequence $\boldsymbol{H} \in \mathbb{R}^{T \times D_v}$, which is required for many NLP tasks such as translation, where both the input and output are a sequence. However, the classification of SITS requires only a single output and not a sequence. It turns out that most naive approaches to achieving this is not well-suited for SITS classification. For example, we might simply choose one of the vectors in the output sequence, such as the last one. However, the last output of self-attention is computed from the similarity between the query of the last element of the sequence and the keys of all others. But the last image of SITS might not produce a meaningful query—for example, it could be occluded by clouds, or acquired after the crop is harvested. To avoid this, we might instead use all output vectors for classification, which can be done by flattening the output matrix to a vector $\hat{\boldsymbol{H}} \in \mathbb{R}^{TD_v}$ and classifying this combined vector using a linear layer. However, this approach requires the temporal length $T$ to be fixed and can thus not handle variable length SITS.

Instead, the approach of Rußwurm and Körner [136] is to choose the output vector with maximum value. While this often chooses a more meaningful output, a drawback of choosing a particular output vector in self-attention is that the computation of all others becomes unnecessary. That is, we are spending $\mathcal{O}(T^2)$ compute and memory, but only $\mathcal{O}(T)$ is required for the chosen output vector. But the maximum strategy cannot avoid this problem, as all outputs are computed to choose the maximum.

**Temporal Attention Encoder.** The Temporal Attention Encoder (TAE), proposed by Sainte Fare Garnot et al. [142], addresses this by defining a single *master query* $\hat{\boldsymbol{q}} \in \mathbb{R}^{D_k}$ computed as the temporal average of the queries $\hat{\boldsymbol{q}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{Q}^{(t)}$. Then, attention is computed between the master query and the keys $\boldsymbol{K}$, resulting in a single output vector instead of a sequence. Moreover, the attention computation is only $\mathcal{O}(T)$ and thus avoids unnecessary compute and memory.

To help account for the irregular temporal sampling of SITS, the day of the year for observation $t$ is used instead of its index in the positional encoding of Equation (3.4). This provides explicit information to the model about the temporal location of the satellite images, which was found to improve classification within the same region seen during training. We discuss this choice further in the context of large-scale classification in Chapter 7.

**Lightweight Temporal Attention Encoder.** A lightweight version of the TAE (LTAE) was later proposed by Sainte Fare Garnot and Landrieu [141], bringing further computational benefits as well as accuracy improvements. Instead of computing the master query from the temporal average $\boldsymbol{Q}$, which first requires a linear transformation of each input element, the master query is simply set as a learnable vector of parameters $\hat{\boldsymbol{q}} \in \mathbb{R}^{D_k}$. This reduces computation and parameters, as the linear transformation to obtain $\boldsymbol{Q}$ is removed. Furthermore, the sequence-to-vector process is now performed with a vector that is independent of the input SITS instead of a particular time step or the average time step. This reduces variance in the output representation when the input SITS is noisy due to *e.g.* clouds, which is likely beneficial for classification. The LTAE is currently the state-of-the-art temporal neural component for crop classification and has improved results in both parcel-wise classification [141] and semantic segmentation of crops [41].

Parcel-wise classification is performed in combination with the Pixel Set Encoder (PSE) [142], *i.e.*, with the PSE+LTAE model. As agricultural parcels are irregularly sized, processing the spatial dimensions of such SITS with convolutions requires interpolating all polygons to a fixed-sized bounding box, which is both time and memory-consuming. Instead, the PSE encodes a random sample of pixels within the parcel polygon, allowing it to efficiently handle irregularly sized parcels.

Semantic segmentation is performed in combination with U-Net [135], named the U-TAE model [41], by applying the LTAE between the encoder and decoder of the U-Net. The computed attention is subsequently applied across each skip-connection, resulting in a highly efficient segmentation model in memory and compute time.

Given the state of the art performance, computational efficiency, and versatility for different classification tasks, we focus on the LTAE model to address large-scale crop classification.

## 3.6 Open Questions

In this section, we discuss which challenges outlined in Section 3.2 are addressed by the LTAE, and which remain open questions.

**Cloud occlusions.** Similar to RNNs, the attention mechanism enables the LTAE to ignore irrelevant time steps in its prediction, such as those occluded by clouds, as observed in [136]. This brings great practical benefits for crop classification, as satellite data can be used directly for crop classification, removing the need for a computationally heavy pre-processing pipeline including cloud detection.

**Class imbalance.** Imbalanced datasets are generally an issue in supervised classification with deep learning and not just crop classification. To generalize well, a large amount of labeled training data is required, but in practice, some classes, such as crop types, are often more frequent than others. In standard training regimes, such as the one used to train LTAE, models tend to ignore infrequent classes and focus on the dominant classes to achieve maximum accumulated performance across the training dataset. In practical crop classification, however, rare classes are often as important as frequent classes, if not more, so addressing class imbalance is essential.

The LTAE does not address this issue directly, but there exist standard strategies to address the class imbalance. One approach is to reweigh the loss function, where infrequent classes receive a higher weight. However, this tends to bias the model towards the infrequent classes, causing more frequent errors in real-world scenarios when the bias changes. Another option is to resample the training data, either by downsampling frequent classes, or upsampling infrequent classes. Still, removing or repeating samples from a training dataset risks that information is lost or repeated unnecessarily. A promising approach may be hierarchical classification [88, 169], which naturally suits the hierarchical structure of crop classes. Here, a label hierarchy is created where examples of rare classes (*e.g.* strawberries, tomatoes, carrots) are also represented at a coarser level (*e.g.* fruits and vegetables). By training the model to classify inputs at different levels, the model is guided to learn relevant features at the coarse level, where more data is available, which is then re-used to boost classification performance at the fine-grained level for rare classes.

**Irregular temporal sampling and length.** The design choices of the LTAE ensure that there is no limitation on the temporal length $T$, allowing models to predict crop types in SITS of any length. Moreover, incorporating acquisition dates through positional encoding allows the model to account for irregular temporal gaps between acquisitions.

**Spectral variation.** This is not explicitly handled by the LTAE. But given the ability of deep learning models to solve many pattern recognition tasks, a possible approach to address spectral variation might simply be to incorporate more data that captures a larger degree of spectral signatures. If labels are not available for regions with specific topographies or soil conditions that change crop growth, addressing this challenge remains an open question.

**Temporal shift variation.** The timings of crop growth in different regions are not the same but can be shifted earlier or later depending on the local weather conditions. The inputs to the LTAE are always sampled from the same temporal endpoints, for example from January 1 to December 31 of a year. The temporal shift variation between different regions means that the phenological development of crops can be arbitrarily shifted forward or backward in time, and this is not handled by the LTAE.

We might attempt to address this problem by training on data containing a larger degree of temporal shift variation to learn shift-invariant models. However, the temporal location of crop growth is also an indicator of its class. For example, in Denmark, winter barley is sown in September the year before and harvested in early August, while spring barley is sown in March and harvested in late August. This means that the crop classification task itself is variant to shifts, which a shift-invariant model cannot capture. Solutions to this particular challenge are part of the contributions in this thesis, as we describe further in the next chapter.

## 3.7 Conclusion

In this chapter, we present the crop classification task and the different types of temporal neural networks for learning the task from satellite image time series. We present several concrete challenges for large-scale crop classification and discuss these challenges in the context of state-of-the-art models based on the self-attention mechanism. We note that while some challenges are addressed, such as clouds and temporal irregularity, there remain open questions. In particular, current models are not robust to temporal shift variation, which is crucial for large-scale classification.

# Chapter 4

# Contributions and Future Work

In this chapter, we describe our contributions to address the research questions in Chapter 1 regarding cloud detection and crop classification. Our research provides insight into some of the open questions we identified in previous chapters.

This chapter is divided into three sections, one for each of the three included papers. Each section provides a description of the proposed methodology in the papers, a summary of the main results, and concludes with an outline of promising future work.

## 4.1 Weakly-Supervised Cloud Detection with Fixed-Point GANs

In *Weakly-Supervised Cloud Detection with Fixed-Point GANs* [115] (referred to as FCD), presented in Chapter 5, we address the first research question: how can we train deep cloud detection models with only weak supervision? Existing deep learning models for cloud detection require pixel-level cloud labels for a large and diverse dataset of satellite images, which are expensive and time-consuming to acquire. Even though deep learning models can, in principle, learn cloud detection for any satellite sensor, the requirement that pixel-level labels are available for each satellite greatly limits the scalability of these methods.

In FCD, we address this by proposing a *weakly-supervised* approach. The typical approach to training cloud detectors is to divide the satellite image into smaller patches which can be input to deep learning models. For each typical input patch of size $256 \times 256$, existing methods thus require $256 * 256 = 65.536$ pixel-level cloud labels to train, which are generally hand-labeled. In contrast, our approach requires only *one* image-level cloud label, thus significantly reducing the amount of manual work required for labeling. We achieve this by learning Fixed-Point GAN models for image translation between clear (*i.e.*, no clouds) and cloudy satellite image patches, using the image-level labels to divide the dataset. At inference time, we translate an input image to clear, thus removing clouds, and predict accurate pixel-level cloud labels from the difference between the input image and the translated image.

### Background

The problem of learning cloud detection with weak supervision is a weakly-supervised semantic segmentation problem. To obtain pixel-level predictions from image-level labels, a common method in CV is to use class activation maps (CAMs) [64, 179, 182, 194], where rough segmentation predictions are obtained using feature maps from an intermediate layer of a pre-trained convolutional image classifier. As CAMs typically only highlight the most discriminative parts of objects (*e.g.*, the snout of a dog or the beak of an eagle), recent methods propose different methods to "grow" the prediction in order to segment the complete object [2, 64]. This process assumes that objects are non-transparent, which is generally true for *e.g.* ImageNet [26]. However, for cloud detection in satellite images, this assumption of existing CAM-based methods will inevitably lead to problems with detecting semi-transparent clouds. In addition, as CAMs are obtained by up-sampling low-resolution feature maps, they are not able to detect small objects, which is required for accurate clouds detection.

### Contributions

Instead of CAMs, we consider an image translation approach with GANs [70, 197] to address weakly-supervised cloud detection. In particular, we learn image translation between clear and cloudy satellite images, thus requiring only image-level labels for training. To predict pixel-level cloud labels at inference time, we translate inputs images to clear ones and compute a binary cloud mask by setting a threshold to the difference between the original and the translated image. The image difference computed this way essentially corresponds to an alpha map, and thus designates the amount of transparency between the clouds and the generated surface. This enables our approach to more naturally handle transparent clouds compared to CAMs. In addition, we operate on the original image resolution, which enables our approach to detect small clouds.

However, the accuracy of the predicted cloud mask greatly depends on the GANs ability to only affect pixels of clouds, and leave surface pixels unaffected. This is not guaranteed with standard image translation methods, such as CycleGAN [197], which essentially perform a "style transfer" between the two image domains, which may introduce unexpected color shifts affecting the entire image. This shortcoming is addressed in the Fixed-Point GAN by Rahman Siddiquee et al. [130] using an additional fixed-point translation loss, which regularizes the model to change a minimal subset of pixels during translation. Therefore, we use the Fixed-Point GAN to learn image translation between clear and cloudy images as described above and address weakly-supervised cloud detection.

In our results, we demonstrate that FCD outperforms CAMs in cloud detection on the Landsat-8 Biome [34] dataset. But we also observe that the GAN-generated cloud masks contain noisy artifacts. To overcome this limitation, we propose FCD+, where we leverage the label-noise robustness of deep learning models [133] to refine the generated cloud masks. We do so by training a U-Net [134] for cloud detection using the images and FCD-generated cloud masks as pseudo-labels. We demonstrate that FCD+ effectively removes the artifacts of FCD, further increasing accuracy. Finally, we show that FCD+ can reach the performance of existing models, which are trained with 100% of the available pixel-level labels, after fine-tuning FCD+ with just 1% of the available pixel-level labels. Our proposed method thus enables label-efficient training of deep cloud detection methods with little to no loss in accuracy.

### Future Work

With FCD, we learn cloud detection with only weak supervision while achieving near fully-supervised performance. One limitation of FCD is that we focus on cloud detection but ignore cloud shadows. In principle, FCD should be able to handle cloud shadows in addition to clouds, as both always appear together in the cloudy images, which the GAN should be able to capture. However, we were unable to handle this as a dataset with accurate cloud shadow labels is not available to evaluate such an approach.

A possible direction to further reduce labeling costs is to investigate whether the image-level labels to train FCD can be generated automatically. For example, an approach could be to use existing threshold-based methods, such as FMask [199], to assign image-level labels.

We later learned of the similar work by Zou et al. [201], which also trains a GAN to translate between clear and cloudy images. The problem is formulated as a foreground/background separation problem, which enables the GAN to directly predict the difference map, whereas we compute the difference in a separate step. However, their approach only handles translation from cloudy images to clear ones, which limits practicality as clear images must also be handled in real-world scenarios. In comparison, our FCD handles both directions. An interesting direction might therefore be to incorporate the foreground/background separation into the cycle consistency loss of FCD.

Lastly, a concern is described in the work of Chu et al. [20], showing that image translation GANs learn to “hide” information about the original image in the generated image. The model uses this information to ensure the original image can be recovered, in order to satisfy the cycle consistency loss. This phenomenon suggests that the quality of the FCD-generated cloud masks could be improved by preventing the network from hiding information in the generated images. A possible approach might be to add an extra latent vector, in which the model can store this encoded information instead of using the generated images.

## 4.2 TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation

In *TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation* [118] (TimeMatch), presented in Chapter 6, we set out to address the second research question: how can we adapt crop classification models to new regions by utilizing unlabeled data? We answer this question by providing a solution to one of the open questions for large-scale crop classification described in Section 3.2, namely the challenge of temporal shifts between regions in growth patterns.

### Background

As mentioned in the previous chapter, there is a large body of work that propose different neural architectures for crop classification. However, despite the importance of crop classification on a large geographical scale, existing work only reports classification results from the same regions in which the model is trained [136, 138, 140, 142]. As a result, it is not known how well these models work when applied to regions different than those seen during training, without which we cannot provide any guarantees on the large-scale performance of these models.

Phenology Alignment Network (PAN) by Wang et al. [180] are, to the best of our knowledge, the first work which studies cross-region crop classification. On a dataset with three different Chinese regions, it is empirically shown that the performance of existing models drastically drops when trained in one region and evaluated in the others. This failure is attributed to discrepancies in the phenological characteristics of the same crop type in different regions, causing differences in the two data distributions, which violates the assumption of supervised learning that training and test data are identically distributed. To address the problem, an *unsupervised domain adaptation* (UDA) approach is proposed. In this setting, labeled data from one region (the *source* domain) and unlabeled data from another region (the *target* domain) is available. The goal of UDA is to train a model using the labeled source data and unlabeled target data which performs well on the target data. To this end, the authors of [180] propose PAN, which employs an existing image-specific UDA method based on learning domain-invariant features [8]. Here, the model is trained with a standard supervised loss on labeled source data, plus an unsupervised loss which conditions network features from source and target to be domain-invariant, that is, be distributed similarly. By doing so, the features from the target region become similar to the features from the source region, such that the classifier trained with source features also works for target features. In PAN, the maximum mean discrepancy loss [170] is used to learn domain-invariant features which encode both spectral and temporal information. While PAN improves cross-region crop classification results, we demonstrate in TimeMatch (Chapter 6) that simply extending image-specific domain adaptation methods to SITS without explicitly considering the temporal aspect is not sufficient to address cross-region crop classification.

### Overview of Method and Results

In TimeMatch, we instead focus on directly aligning the temporal dimension for cross-region UDA. One of our key observations is that crop phenology between two regions is *temporally shifted* (as also discussed in Section 3.2). While this phenomenon has been known for a long time in agricultural studies [104, 105], TimeMatch is the first to highlight its importance and address the problem of large-scale crop classification. In particular, we observe that when a source-trained model is applied to target data, its performance depends on the timestamps—the day of the year—which are provided along with the SITS. If the timestamps are all shifted by a particular number of days, the performance of the model significantly improves, even achieving higher performance than the domain-invariant method of PAN. Moreover, we observe that the "best" temporal shift is a function of the climate, as it consistently corresponds to how much warmer or colder one region is compared to the other. For example, the temporal shift of a model trained in a Danish region and applied to a French region is about +30 days. This corresponds to the fact that the climate of the French region is warmer than the Danish one, which causes crops to develop earlier in the French region.

We exploit this observation in the methodology of TimeMatch, which consists of two components: temporal shift estimation and TimeMatch learning. From our observation that the performance of source-trained models improves by temporally shifting the target data, our goal is to estimate the temporal shift without using target labels and then apply the temporal shift to automatically assign pseudo-labels for the target data. These pseudo-labels are then used in TimeMatch learning, a self-training routine that re-trains the model for the target data.

Our TimeMatch learning algorithms work by first duplicating a source-trained model into a *teacher* and a *student*. Using the estimated shift, the teacher generates pseudo-labels for the target region to train the student. The knowledge learned by the student is gradually updated back to the teacher during training via an exponential moving average of its parameters, thus adapting both models to the target region and improving the pseudo-labels. As the teacher adapts to the target region, the temporal location of crop growth in the source region is gradually "forgotten" and replaced with that of the target region. This means that temporally shifting the target data gradually becomes unnecessary to generate pseudo-labels with the teacher. We account for this by re-estimating the temporal shift of the teacher every epoch, which ensures that the pseudo-labels remain accurate during TimeMatch learning.

We evaluate our approach on a dataset containing SITS from four different European regions, with one in Austria, one in Denmark, and two in France. On five different adaptation scenarios, TimeMatch consistently outperforms all competing methods, improving results by 11% in average F1 score, thus setting a new state-of-the-art for cross-region UDA. However, we also observe that there is still a gap between the results of TimeMatch and the results achieved by a fully-supervised model. While completely closing the gap is likely unrealistic without any labels, reducing the gap is a direction for future work.

### Future Work

Our work on TimeMatch highlights the importance of addressing the temporal discrepancy for large-scale classification, and we believe there are many exciting approaches for future research in this regard.

One of the limitations of TimeMatch is that our temporal shift estimation is a simple form of temporal alignment which assumes that all crop types are shifted by the same value, which is unlikely to be the case in practice. A possible improvement in this direction could be to consider stronger temporal alignments, such as at the class or example level, or to align the data by time-warping techniques [7].

Another line of work is to account for the change in classes between two regions, such as the techniques applied in open-set domain adaptation [121], to handle the common real-world scenario case where the set of classes in the target domain differs from that of the source domain—for example, sunflowers are a frequent crop type in southern France, but are rarely planted in Denmark.

Another limitation with UDA is that it requires training the model on unlabeled data for every new target region. In large-scale crop classification, this would mean training the model a hundred to a thousand times over for every new region across the world, which would lead to a significant performance issue for scalability. A favorable alternative to UDA would therefore be domain generalization, which avoids this extra training step altogether by improving the generalization of the model itself. We consider such an approach in the next section.

## 4.3 Generalized Satellite Image Time Series Classification with Thermal Positional Encoding

In *Generalized Satellite Image Time Series Classification with Thermal Positional Encoding* [118] (TPE), presented in Chapter 7, we set out to address the last research question: how can we train deep crop classification models that generalize to new regions? Our work on TimeMatch gave us the understanding that accounting for temporal shifts of the growing season is key for classification in different regions. Therefore, in TPE, we build upon our findings in TimeMatch and improve generalization by training models which are robust to temporal shifts but do not need to estimate the shift itself.

### Background

As described in Chapter 3, the current methods in crop classification are based on self-attention, in particular, the current state-of-the-art models are LTAE [141], which modify the self-attention mechanism for crop classification. Our work focuses on improving robustness to temporal shifts in this component since it handles the temporal dimension. As the self-attention computation is position-agnostic, positional encodings are used to provide explicit timing information [173]. Commonly, the timing information provided to crop classifiers is the number of days passed since the first observation (the day of the year if the SITS starts on January 1), and the LTAE also provides these values to the model via positional encoding. The benefit of this type of temporal information, which refer to *as calendar time*, is that the network can tell the order of the input elements and their temporal location in the growing season. This information can be an important clue in separating similar phenological events of different crop types. For example, the spectral values corresponding to the peak of growth for winter and spring barley might be similar, but the two events can easily be classified based on the timing, as winter crops mature earlier than spring crops. However, the problem with calendar time is that the timing which the model learns in one region does not apply to another due to temporal shifts, which causes the accuracy to drop. Our work on TimeMatch gives us the understanding that models rely on the timing information for classification, as shifting the calendar time such that crop growth in the target region aligns with that of the source region, the performance of the model improves.

## Contributions

In TPE, we consider multiple strategies to improve the robustness of models to temporal shifts. A possible approach is to train *shift-invariant* models, which can be achieved by *e.g.* removing positional encoding altogether or applying random temporal shifts to training data. On the same dataset as TimeMatch, our results show that these approaches in fact do improve the generalization of crop classifiers. However, shift-invariant models naturally cannot capture the temporal shifts between classes which we observe is important for classification with shift-variant models.

To address this issue, we propose to forego calendar time in place of different timing information called *thermal time* [104, 105]. In the field of crop phenology, thermal time is measured in units of growing degree days (GDD), which are computed by accumulating daily mean temperatures over a baseline. Crops enter different phenological stages at particular GDD [76], which makes thermal time directly related to the timing of crop growth. As a result, temporal shifts between regions are reduced with thermal time, improving the generalization of models, while simultaneously allowing models to capture the class-wise phenological timings.

In addition to using thermal time in the positional encoding of self-attention models, we also investigate methods for the positional encoding itself. Commonly, the positional encoding used for crop classification stems from the original Transformer model [173], where positions are encoded to a predefined sinusoidal vector, which is then added to the input sequence of word embeddings. This approach is practical for NLP tasks, as word embeddings are typically pre-trained and not trained together with self-attention. In comparison, for crop classification, the input SITS embeddings are learned jointly with self-attention. One of our TPE methods, TPE-Concat, leverages this observation and simply concatenates thermal time with the input SITS to learn the positional encoding and SITS embedding simultaneously. We show that this improves computational efficiency while obtaining similar accuracy as the commonly used sinusoidal positional encoding. Finally, our top-performing TPE method, TPE-Recurrent, shows that a positional encoding of thermal time that captures both the absolute values and the historical rate of crop growth can lead to further improved generalization.

We demonstrate our approach for large-scale crop classification in Europe and show that our method enables Europe-wide classification without requiring the model to be re-trained for each region as with UDA.

### Future Work

Our work on TPE shows that using a time representation that captures the different climatic variations of crop growth can improve the generalization of crop classifiers. Our experimental results show that TPE enables accurate large-scale crop classification in Europe, which is not possible with existing methods. However, it remains an open question how we may achieve crop classification on a global scale.

In our work, the inputs contain satellite images from January 1 to December 31 of the same year, capturing both the early development of crops and their harvest in the summer. However, this range assumes that the SITS are acquired in the Northern Hemisphere, where the winter lasts from December to March and summer from June to September. It would not apply to the Southern Hemisphere where the winter and summer seasons are swapped. We may account for this by acquiring the SITS from the middle of the year instead, *i.e.* from June 1 to May 31. Still, another assumption of this range is that it contains only a single growing season. This would not apply to *e.g.* some regions of Brazil, where the tropical climate allows the planting of one crop during the summer followed by another during the winter. Therefore, we believe a promising future direction could be to dynamically select the start and end dates of the SITS prior to its classification. That is, the SITS should start on the earliest sowing date and end on the latest harvest date of parcels in each region. An automatic selection of these dates could be done from NDVI as in [77, 80]. By normalizing the temporal "window" of SITS in different regions, such an approach would likely also reduce the issue of temporal shifts. We expect that thermal time could still be used in such a model to adjust to the variation of growth rates in this dynamic window, *e.g.* to account for the faster growth rate in Brazil compared to Europe.

While we focus on generalization across space in our work, another possible application of TPE is generalization across time. Similar to that the climatic variation in different regions causes temporal shifts in crop phenology, the variation in the weather for the same region in different years also changes the timing of crop growth, which TPE could help models adapt to. Finally, crop growth is affected by many factors and not just the temperature, and it would be interesting to see the effect of other meteorological variables, such as precipitation, or to account for spectral variation through information about the soil or topography.

## 4.4 Conclusion

This chapter concludes the first part of this thesis. In Chapter 1: *Introduction*, we motivate the use of deep learning methods for agricultural tasks with remote sensing data, based on their recent success on related data in many other fields such as images in computer vision and sequences in natural language processing. But we also highlight that obtaining labeled training datasets is a significant challenge for remote sensing, and therefore describe three research questions to address parts of the challenge.

The first research question concerns the label requirements for cloud detection, a problem particular to remote sensing data described in Chapter 2: *Satellite Image Analysis for Precision Agriculture*. To answer this question, our first paper, presented in Chapter 5: *Weakly-Supervised Cloud Detection with Fixed-Point GANs*, provides a weakly-supervised cloud detection method. The second and third research questions concern the label requirements for the crop classification task. We describe this task in Chapter 3: *Large-Scale Crop Classification*, as well as the deep learning methods that are currently used and their limitations. The second research question concerns unsupervised domain adaptation of crop classifiers, and we provide a solution to this problem in the second paper, presented in Chapter 6: *TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation*. Our method explicitly accounts for the temporal shift between different regions to adapt crop classifiers. Finally, the third research question concerns how the generalization of current crop classifiers can be improved. We provide a solution to this in the third paper by incorporating thermal time to positional encoding, as presented in Chapter 7: *Generalized Satellite Image Time Series Classification with Thermal Positional Encoding*.

The next part of this thesis consists of the three papers. We thank the reader for following along this far and hope that this first part provides a comprehensible overview for the three papers that follow.

# Part II

# Publications

# Chapter 5

# Weakly-Supervised Cloud Detection with Fixed-Point GANs

JOACHIM NYBORG, AARHUS UNIVERSITY, DENMARK
IRA ASSENT, AARHUS UNIVERSITY, DENMARK

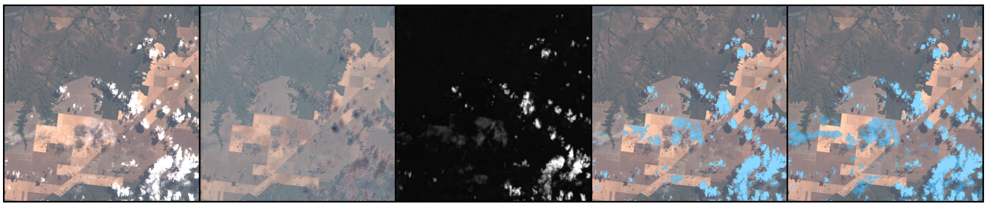
In *Proceedings of the 2021 IEEE International Conference on Big Data* for the *3rd IEEE Workshop on Machine Learning for Big Data Analytics in Remote Sensing*.

**Abstract**

The detection of clouds in satellite images is an essential preprocessing task for big data in remote sensing. Convolutional neural networks (CNNs) have greatly advanced the state-of-the-art in the detection of clouds in satellite images, but existing CNN-based methods are costly as they require large amounts of training images with expensive pixel-level cloud labels. To alleviate this cost, we propose Fixed-Point GAN for Cloud Detection (FCD), a weakly-supervised approach. Training with only image-level labels, we learn fixed-point translation between clear and cloudy images, so only clouds are affected during translation. Doing so enables our approach to predict pixel-level cloud labels by translating satellite images to clear ones and setting a threshold to the difference between the two images. Moreover, we propose FCD+, where we exploit the label-noise robustness of CNNs to refine the prediction of FCD, leading to further improvements. We demonstrate the effectiveness of our approach on the Landsat-8 Biome cloud detection dataset, where we obtain performance close to existing fully-supervised methods that train with expensive pixel-level labels. By fine-tuning our FCD+ with just 1% of the available pixel-level labels, we match the performance of fully-supervised methods. Our source code is publicly available at `https://github.com/jnyborg/fcd`.

## 5.1 Introduction

Clouds are a major issue when analyzing big data in remote sensing, as clouds often partially or entirely obscure a given area of interest. As a result, clouds have a



(a) Input (b) Generated (c) Difference (d) Prediction (e) Ground truth

Figure 5.1: Weakly-supervised cloud detection with FCD. With easily acquired image-level labels of whether satellite images contain clouds or not, we train a generative model for fixed-point translation between clear and cloudy images such that the difference between the input and generated image reveals clouds at the pixel-level.

significant negative impact on a variety of applications that require a clear view of the ground below, such as change detection [198] and cropland monitoring [39]. Thus, detecting and masking clouds is an essential preprocessing step in most satellite image pipelines.

One line of work for cloud detection considers rule-based methods, such as the Fmask algorithm [199] and the MAJA [54] processor. These methods detect clouds by applying thresholds to selected features based on the physical characteristics of clouds. However, as these methods are specific to selected satellites and often contain hand-crafted rules, their direct application to the growing constellation of satellites is difficult. As a result, these methods are not yet available for the majority of high-resolution commercial satellites.

Instead of rule-based methods, supervised learning of Convolutional Neural Networks (CNNs) bring the benefit of leveraging learned features, allowing these methods to automatically adapt to any particular satellite sensor. As in other visual recognition tasks, CNNs have also greatly advanced the state-of-the-art in cloud detection [72, 148, 187] as a result. However, due to the data-hungry nature of CNNs, this approach requires a large number of labeled training images that capture the large variance of clouds and ground surfaces, with each image labeled with ground truth cloud masks typically hand-drawn by experts. Consequently, if no such dataset is available for the satellite at hand, training CNNs for cloud detection is very expensive.

One way to alleviate this issue is by weakly-supervised learning, where weaker but less expensive image-level labels are used to train models that are capable of pixel-level predictions. A popular approach for weakly-supervised learning in both natural images and remote sensing is based on computing class activation maps (CAMs), whereby the feature maps learned by an image-level classifier are used to construct a pixel-level prediction [12, 36, 114, 177].

In this paper, we propose an alternative approach for weakly-supervised cloud detection based on the Fixed-Point GAN [130] (Generative Adversarial Network [50]). Our proposed method, *Fixed-Point GAN for Cloud Detection* (FCD), learns image-to-image translation between *clear* (no clouds) and *cloudy* image patches taken from complete satellite images, thus requiring only image-level labels for training. Due to the fixed-point translation ability of FCD, our approach is able to translate an input image into a clear image while affecting only pixels containing clouds. This enables a pixel-level cloud mask to be predicted by setting a threshold to the difference between the original and translated image, as shown in Figure 5.1.

To further improve weakly-supervised cloud detection performance, we propose FCD+. Here, we first utilize FCD to generate pseudo cloud masks for training images to train existing CNN models. This enables us to refine the cloud masks of FCD by removing generative artifacts for improved performance. Furthermore, we show that our FCD+ is a powerful weakly-supervised pretraining strategy for cloud detection, as, by fine-tuning our model with just 1% of patches with pixel-level ground truth, we match the performance of existing models that train with full supervision from all available pixel-level ground truth. In summary, our contributions are the following:

- We propose FCD, a weakly-supervised cloud detection method based on Fixed-Point GAN.
- We propose FCD+, a training strategy that refines the predictions of FCD and allows for weakly-supervised pretraining of cloud detection CNNs.
- We demonstrate that FCD and FCD+ outperform existing CAM-based methods in weakly-supervised cloud detection on the Landsat-8 Biome dataset [34]. By fine-tuning FCD+ with 1% of available pixel-level labels, we match the performance of models that receive full supervision from all available labels.

## 5.2 Related Work

### Cloud Removal with GANs.

Recent work has applied GANs to cloud removal [30, 53, 146, 154], which aims to remove clouds from satellite images and replace them with a realistic, generated region of the underlying ground surface. In [30, 53, 146], cloud removal is learned based on pix2pix [70], requiring pairs of cloudy and clear images for training, acquired either by synthesis [30, 53] or by satellites with high revisit rates [146]. CloudGAN [154] instead learns unpaired cloud removal with CycleGAN [197], simplifying data acquisition. While we similarly learn a GAN to translate cloudy images to clear ones, our approach differs from this line of work as we do not focus on generating realistic images, but on cloud detection. This requires GANs capable of minimal translation, changing only pixels of clouds, so clouds can be detected by the difference between input and translated image. However, pix2pix [70] or CycleGAN [197] tend to make unnecessary changes and introduce artifacts to translated images [130], thus limiting their use for weakly-supervised cloud detection. Instead, we base our approach on Fixed-Point GAN [130], which enables our model to perform minimal translations for accurate cloud detection.

### Weakly-Supervised Semantic Segmentation.

Weakly-supervised semantic segmentation (WSSS) methods using image-level supervision have been widely used for natural images [2, 64, 84, 179, 182]. These approaches typically [179] use class activation maps (CAMs) [15, 150, 194], where a CNN trained for image-level classification is utilized to roughly localize object areas by drawing attention to discriminative parts of objects based on global average pooling [194] or gradient backpropagation [15, 150]. As CAMs have limited resolution and only cover small parts of objects, most approaches refine CAMs to discover complete object regions, by for instance seeded region growing [64], adversarial erasing [182], or equivariant regularization [179]. These approaches improve upon the standalone CAMs and are typically independent of the specific choice of weakly-supervised localization method, which allows similar improvement to alternative methods that localize objects from image-level labels, such as the Fixed-Point GAN [130] and our FCD. For this reason, we compare CAMs to our FCD in our experiments.

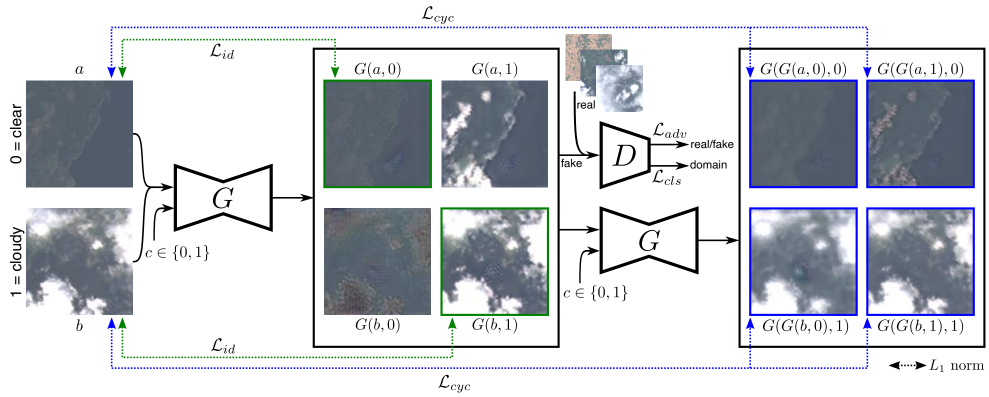


Figure 5.2: Overview of FCD. We learn fixed-point translation between clear and cloudy image patches. During training, the discriminator $D$ learns to distinguish between real and fake images and to classify real images correctly. The generator $G$ is input an image and whether the image should be translated to the clear domain ($c = 0$) or the cloudy domain ($c = 1$), and learns to either remove or add clouds through a cycle-consistency loss and an adversarial loss. If $c$ is the same domain as the input image, $G$ must perform an identity mapping through the identity loss, which regularizes the generator to only affect the clouds. We then detect clouds in an input image $x$ by the difference between the original image and its clear translation with $G$.

In remote sensing, existing WSSS methods have mostly used CAM-based approaches for segmenting satellite images [12, 114] from image-level labels. Fu *et al.* [36] propose WSF-Net, which computes CAMs from fused, multi-level features. Wang *et al.* [177] propose U-CAM, which adapts CAMs for U-Nets [135]. In contrast to these CAM-based approaches, our approach is based on Fixed-Point GAN, which our experimental results suggest is more accurate for weakly-supervised cloud detection.

## 5.3 Weakly-Supervised Cloud Detection Model

### Overview

Our goal is to develop a method for weakly-supervised cloud detection. To achieve this, we base our approach on the Fixed-Point GAN by Siddiquee *et al.* [130], a recent method successful in GAN-based image-to-image translation for weakly-supervised disease localization in medical images. GANs [50] have achieved remarkable results in generating realistic images, and have also significantly improved image-to-image translation [197], where the goal is to translate images from one domain of images to another, such as translating aerial images to maps [70] or changing the season of images from summer to winter [197]. Fixed-Point GAN learns image-to-image translation with the goal of removing objects, if present, from an image, while otherwise preserving the image content. For cloud detection, this objective translates to removing clouds, such that when translating an image, either clear or cloudy, into a clear image, clouds are revealed by subtracting the original image from the generated image. A GAN capable of such a task must satisfy the following four requirements [130]:

- **Req. 1:** The GAN must learn unpaired image-to-image translation, as in general, it is difficult and time-consuming to obtain perfect pairs of clear and cloudy satellite images due to the high temporal variability of the ground surface.
- **Req. 2:** The GAN must require no domain label for the input image, as, at inference time, the GAN must be able to translate any image (both clear and cloudy) into a clear image.
- **Req. 3:** The GAN must perform an identity transformation for same-domain translation. When translating an image from clear to clear, the GAN should simply leave the image intact, injecting neither artifacts nor new information into the image.
- **Req. 4:** The GAN must perform a minimal image transformation for cross-domain translation. When translating cloudy to clear, the GAN should only affect the clouds, while leaving the ground intact.

As Fixed-Point GAN [130] introduces a GAN that satisfies all of these four requirements, so does FCD. However, for such a method to be practical for accurate cloud detection, we require the following as well:

- **Req. 5:** The method must output consistent cloud masks without generative side effects that would lead to decreased cloud detection performance.

- **Req. 6:** It must be possible to fine-tune the method with limited amounts of images labeled with pixel-level ground truth to achieve a cloud detection performance that matches fully-supervised methods.

We observe that the generative objective of FCD leads to artifacts that lower its cloud detection performance. Furthermore, as FCD optimizes an image translation objective, it is not possible to incorporate pixel-level ground truth to increase performance. We address the last two requirements with FCD+: By training a segmentation model with FCD cloud masks as pseudo labels, we show experimentally the side effects of FCD are addressed. Also, as FCD+ is trained for classification instead of image translation, it can be fine-tuned with a few labeled images to match the performance of existing fully-supervised methods.

### Image-to-Image Translation with GANs

In this section, we describe the background literature surrounding Fixed-Point GAN.

A GAN model [50] typically consists of two neural networks: a generator $G$ and a discriminator $D$. The two networks are trained adversarially by optimizing the *adversarial loss*, where the discriminator $D(y) \in [0,1]$ learns to determine whether a given input image $y$ is real or fake, while the generator $G(z) \to y$ learns to transform a random input $z$ into a fake image $y$ indistinguishable from real images. As a result, $G$ is able to generate highly realistic images from a random input. To apply GANs for image-to-image translation, we replace the random input $z$ with an image $x$, so that $G$ learns a mapping between images. In pix2pix [70], image-to-image translation is learned in a supervised manner by combining the adversarial loss with an L1 loss, which requires paired data samples for training. This violates Req. 1, as for cloud detection, pairs are generally not available, as discussed in Section 5.2.

To overcome the issue of requiring pairs, CycleGAN [197] instead combines the adversarial loss with a *cycle consistency loss*, allowing for unpaired image-to-image translation. Specifically, for each pair of domains $(X,Y)$, two generators $G,F$ are trained for each direction of translation, $G : X \to Y$ and $F : Y \to X$. The cycle consistency loss encourages that these two generators are inverses, by constraining that $F(G(x)) \approx x$ and $G(F(y)) \approx y$, thus enforcing that when translating an image $x$ to an image $y$, we should be able to restore the original $x$ from $y$, and vice versa. The effect of the cycle consistency loss is that the resulting output of the generator is constrained so that the translated image appears aligned with the input image, as if paired, but without any requirements for paired training samples. However, as CycleGAN requires two generators for each pair of image domains, it fails to satisfy Req. 2, as selecting the right generator for correct translation requires a domain label for the input image at inference time.

StarGAN [19] overcomes this limitation by learning a single generator for translation between all domain pairs. This is achieved by conditioning $G$ on a target domain label $c_y$ to indicate which domain $G$ must translate to, so that $G(x,c_y) \to y$. During training, $c_y$ is randomly chosen so that $G$ learns translation between all domain pairs.

To control that the generated image correctly classifies to the domain $c_y$, the discriminator is expanded with an auxiliary classifier $D_{cls}$ to enforce the *domain classification loss*. That is, the discriminator now produces two outputs: $D_{adv}$ for the adversarial loss, and $D_{cls}$ for the domain classification loss.

Still, StarGAN does not satisfy Req. 3 and 4: StarGAN fails to handle identity same-domain translations, and also tends to make unnecessary changes during cross-domain translation, as shown by Siddiquee *et al.* [130]. To satisfy Req. 3, Fixed-Point GAN introduces an additional *conditional identity loss*, where, in the case that $G$ is given an input image $x$ and a domain label $c_x$ with the same domain as $x$, $G$ learns to do an identity translation by constraining that $G(x, c_x) \approx x$. As such, when translating clear images to clear, $G$ must output the input $x$ unchanged. To satisfy Req. 4, Fixed-Point GAN revises the adversarial, domain classification, and cycle consistency loss to explicitly learn same-domain translation, in that $G$ must optimize these losses both for cross-domain translation $G(x, c_y) \to y$ similar to StarGAN, but also for same-domain translation $G(x, c_x) \to x$. By doing so, the generator is regularized to find a minimal transformation during cross-domain translation [130], thus satisfying Req. 4, so that when $G$ translates from cloudy to clear, only clouds are changed.

## 5.4 Fixed-Point GAN for Cloud Detection

In the following, we formally describe the loss functions of Fixed-Point GAN for Cloud Detection (FCD). Figure 5.2 shows an overview of the FCD training scheme.

**Adversarial Loss** To ensure that the images generated for both cross- and same-domain translation follow the distribution of training images, an adversarial loss is enforced for each case:

$$\begin{aligned}\mathscr{L}_{adv} = &\mathbb{E}_x[\log D_{adv}(x)] \\ &+ \mathbb{E}_{x,c_x}[\log(1 - D_{adv}(G(x, c_x)))] \\ &+ \mathbb{E}_{x,c_y}[\log(1 - D_{adv}(G(x, c_y)))],\end{aligned} \tag{5.1}$$

where $G$ generates two images, conditioned on an input image $x$ and either its original label $c_x$ for same-domain translation or a uniformly chosen target label $c_y$ for cross-domain translation, while $D$ must distinguish between real and fake images. $G$ tries to minimize this objective, and $D$ tries to maximize it.

**Domain Classification Loss** In addition to generating images that follow the overall distribution of training images, $G$ must also use the given domain label $c_x$ or $c_y$ to generate an image that is properly classified to that domain. This is achieved by the domain classification loss defined via the auxiliary classifier $D_{cls}$, with a term for optimizing both $D$ and $G$. For $D$, we enforce that real images must be correctly classified:

$$\mathscr{L}^r_{cls} = \mathbb{E}_{x,c_x}[-\log D_{cls}(c_x|x)], \tag{5.2}$$

where $D_{cls}(c_x|x)$ represents the conditional probability of $x$ belonging to its original domain $c_x$, as computed by $D$.

Similarly, for $G$, we enforce that generated images must be classified correctly as well,

$$\begin{aligned} \mathscr{L}_{cls}^{f} = & \mathbb{E}_{x,c_y}[-\log D_{cls}(c_y|G(x,c_y))] \\ & + \mathbb{E}_{x,c_x}[-\log D_{cls}(c_x|G(x,c_x))], \end{aligned} \tag{5.3}$$

where we have a case for both same- and cross-domain translation. Overall, the domain classification loss ensures that $G$ correctly conditions on the given domain label, allowing us to explicitly choose whether $G$ should translate to the cloudy or the clear domain with a single generator.

**Cycle Consistency Loss** Minimizing the adversarial loss and the domain classification loss ensures the generator outputs realistic images of the correct domain but does not guarantee that the generated images have any relation to the input image. This is addressed with the cycle-consistency loss:

$$\begin{aligned} \mathscr{L}_{cyc} = & \mathbb{E}_{x,c_x,c_y}[||x - G(G(x,c_y),c_x)||_1] \\ & + \mathbb{E}_{x,c_x}[||x - G(G(x,c_x),c_x)||_1], \end{aligned} \tag{5.4}$$

where $G$ takes an image translated to either the target domain $G(x,c_y)$ or input domain $G(x,c_x)$, as well as the original domain label $c_x$, and in both cases tries to reconstruct the input image $x$. As $G$ must be able to reconstruct the input image from the generated image, $G$ is constrained to preserve a relation to the input image, resulting in translations that change only domain-related parts.

**Conditional Identity Loss** To avoid false positives, $G$ should not attempt to remove clouds from clear images, and instead output the input without any change. To this end, we enforce that $G$ acts as an identity function when performing same-domain translations:

$$\mathscr{L}_{id} = \mathbb{E}_{x,c_x}[||x - G(x,c_x)||_1], \tag{5.5}$$

where, in the case that $G$ is given an input image $x$ and its original domain label $c_x$, it must return the input $x$ without introducing any changes.

**Full Objective** In combination, the Fixed-Point GAN objective functions to train $D$ and $G$, respectively, are

$$\mathscr{L}_D = -\mathscr{L}_{adv} + \lambda_{cls}\mathscr{L}_{cls}^{r}, \tag{5.6}$$

$$\mathscr{L}_G = \mathscr{L}_{adv} + \lambda_{cls}\mathscr{L}_{cls}^{f} + \lambda_{cyc}\mathscr{L}_{cyc} + \lambda_{id}\mathscr{L}_{id}, \tag{5.7}$$

where $\lambda_{cls}, \lambda_{cyc}, \lambda_{id}$ are hyper-parameters for tuning the relative importance of the domain classification, cycle consistency, and conditional identity loss.

**Detecting Clouds** Optimizing the Fixed-Point GAN objective functions in FCD results in a generator $G$ capable of translation between clear and cloudy images. To generate cloud masks with FCD, we translate input images to clear, and threshold the difference between the translated and original image. Specifically, given an input patch $x$ of either clear or cloudy, we first compute $y = G(x, 0)$, resulting in $y$, a clear version of $x$. This is followed by computing the absolute difference of $x$ and $y$, followed by the mean across channels. The result is a gray-scale image, which we refer to as the *difference map*. Finally, we produce a binary cloud mask by setting a threshold to the difference map, where pixels with high change are labeled as clouds, and otherwise as clear.

### FCD+: Refining Generative Side Effects

Although FCD enables the generation of pixel-level cloud labels from image-level supervision, we observe that the generative goal causes two types of side effects that lower its performance: Generative artifacts and patch-shaped “holes".

- Generative artifacts: we find that the cloud mask of FCD often contains noise, typically around the edges of clouds. These artifacts likely arise from thresholding the difference map for transparent clouds. As clouds typically become more transparent closer to their edges, whereas their centers are bright white, a lower brightness change is required by the generator to translate pixels of transparent clouds to land surface compared to the center of clouds that are often bright white. This can result in noisy artifacts when the difference value is close to the threshold.

- Patch-shaped holes: when combining cloud masks for multiple patches, FCD in some cases ignores clouds in one patch, even though neighboring patches contain overlapping clouds, which results in patch-shaped holes in the final cloud mask. This issue likely stems from the conditional identity loss of Eq. 5.5. This loss implicitly enforces $G$ to classify input patches as clear or cloudy, as $G$ must act as an identity function only for clear patches. If $G$ wrongly classifies a cloudy patch as clear, it likely outputs an empty cloud mask for the entire patch, resulting in patch-shaped holes in the final image.

To refine these generative side effects, we propose FCD+, where we train a standard fully-supervised cloud detection model with the noisy cloud masks of FCD as pseudo-labels. By doing so, we utilize the label noise robustness of CNNs [109, 133] to refine the FCD cloud masks, thereby addressing Req. 5. Additionally, as FCD+ is trained for classification directly, it can be fine-tuned, thus satisfying Req. 6.

Following existing cloud detection architectures [72], we use a standard U-Net [135] network architecture for FCD+, a fully-convolutional segmentation network with skip connections between the encoder and decoder.

## 5.5 Experimental Setup

We demonstrate the effectiveness of our method on the Landsat-8 Biome dataset [34], where we apply our weakly-supervised FCD to generate pixel-level pseudo cloud masks for training images and train our segmentation model FCD+ with the images and their pseudo masks. We evaluate the quality of our FCD-generated pseudo masks as well as the test set performance of FCD+ when trained with them. Our source code is publicly available at `https://github.com/jnyborg/fcd`.

### Dataset

The Landsat-8 Biome dataset [34] is a cloud detection dataset with 96 Landsat-8 scenes of 8 different biomes with various proportions of cloud cover. Each scene is hand-labeled with pixel-level class labels for clear, thin cloud, cloud, and cloud shadow. We combine "thin cloud" and "cloud" into one class for clouds, and combine "clear" with "cloud shadow" as one class for clear. To ensure an even distribution of biomes in our training, validation, and test sets, we randomly assign the 12 scenes in each biome by a 6:2:4 ratio, totaling 48 images for training, 16 images for validation, and 32 images for testing. Our test set contains 43% clear to 57% cloudy pixels. We input Landsat-8 scenes to CNNs by dividing scenes into patches of size $128 \times 128$. We input all 10 available 30m bands. We use the provided pixel-level labels to decide image-level labels, and label a patch as cloudy if there is at least one cloudy pixel in the corresponding ground truth, otherwise clear.

### Comparisons

We compare FCD to the weakly-supervised methods CAM [194], GradCAM [150], and GradCAM++ [15], as most existing methods in weakly-supervised semantic segmentation are based on CAMs (see Section 5.2). CAM compute cloud masks based on the global average pooling layer of a classifier trained for binary cloud classification of images, whereas GradCAM and GradCAM++ compute the cloud mask based on gradient backpropagation.

### Implementation.

We implement FCD following the original implementation of Fixed-Point GAN [130], and update $G$ and $D$ for 10-channel images. We use the default model hyper-parameters, and train for 200,000 iterations with a batch size of 16, setting $\lambda_{cls} = 1$, $\lambda_{cyc} = 10$, and $\lambda_{id} = 10$. We compute CAMs from ImageNet-pretrained ResNet-50 models [59] trained with Landsat-8 Biome image-level labels. For both FCD and CAM models, we use the validation set to select the best model weights and choose the best threshold value to create binary cloud masks. We note that generating pseudo masks for patches with a clear label is unnecessary, as we know their cloud mask contains only background. Hence, we evaluate only methods in their ability to generate pseudo masks for cloudy patches.

Table 5.1: Cloud mask generation performance (%) for Landsat-8 Biome with existing CAM methods and our FCD.

| Method | F1-score | Accuracy |
|---|---|---|
| CAM [194] | 75.9±0.5 | 82.9±0.7 |
| GradCAM [150] | 70.5±1.5 | 78.6±0.5 |
| GradCAM++ [15] | 72.2±3.8 | 79.9±1.9 |
| FCD (ours) | **83.9±0.8** | **87.6±0.5** |

We implement our FCD+ based on the U-Net [135] using the library in [188]. FCD+ trains for 30 epochs with a batch size of 64 using Adam [82] with the default settings. We save the model that gives the best F1-score on the validation set. We use a learning rate of 1e-4, dropped by 10 if after 3 epochs the validation F1 does not increase. In addition to optimizing a pixel-level cross-entropy loss, we use the available image-level cloud labels to optimize a binary cross-entropy loss by attaching a classifier to the encoder. When fine-tuning, we change our initial learning rate to 1e-5 and freeze the encoder weights.

## 5.6 Results

### Weakly-Supervised Cloud Mask Generation

To verify the effectiveness of FCD, we evaluate its ability to generate pixel-level pseudo masks from image-level labels for our Landsat-8 Biome *train* set, which we then use in our final stage for training FCD+.

**Quantitative Results** Table 5.1 shows the overall cloud detection results for FCD in comparison with CAM [194], GradCAM [150], and GradCAM++ [15]. We find that our FCD greatly outperforms all CAM variants in generating pseudo cloud masks from just image-level labels, increasing F1-score by 8.0% compared to the best CAM variant. This strongly indicates a Fixed-Point GAN approach for weakly-supervised cloud detection is more accurate than CAM-based ones.

**Qualitative Results** We illustrate examples of cloud masks generated by FCD in Figure 5.4, showing views of various Landsat-8 scenes. FCD generates accurate cloud masks with high similarity to the ground truth, but we observe issues of generative side-effects. For the Shrubland and Wetlands examples, we observe generative artifacts particularly for areas with semi-transparent clouds. Patch-shaped holes appear mostly in areas where FCD likely confuses cloudy patches with clear, such as in the center of clouds in SnowIce and Water biomes (where clouds can be confused with snow), as well as areas in the Shrubland biome which contains mostly clear ground with a few transparent clouds. Next, we show how we refine these errors with FCD+.

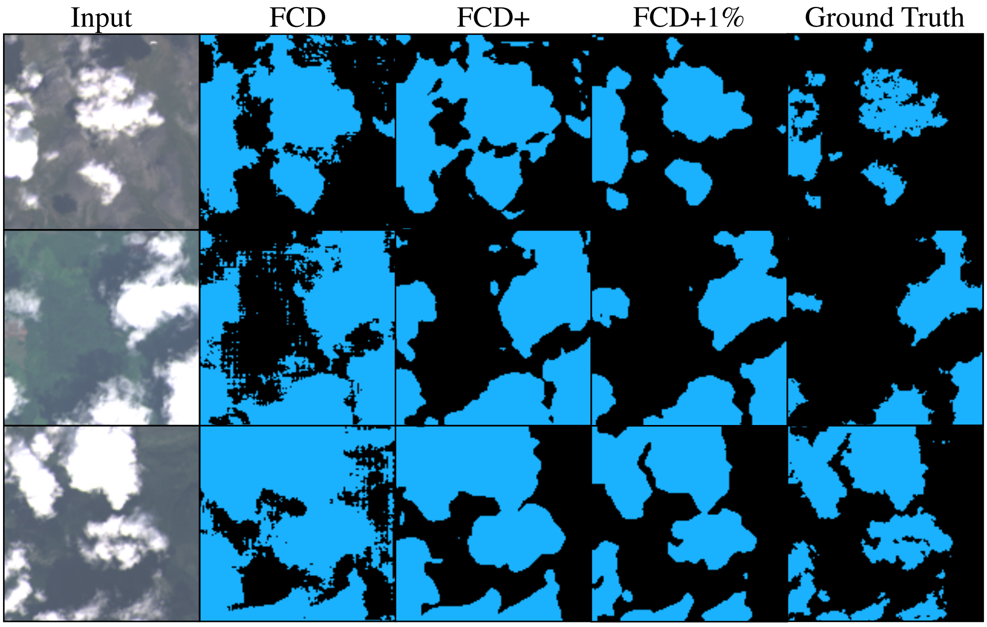


Figure 5.3: FCD, FCD+, and FCD+1% cloud masks for example patches. FCD+ predicts a smoother cloud mask, that removes generative artifacts of FCD. FCD+1% predict a more precise cloud mask after fine-tuning with very few ground truth cloud masks.

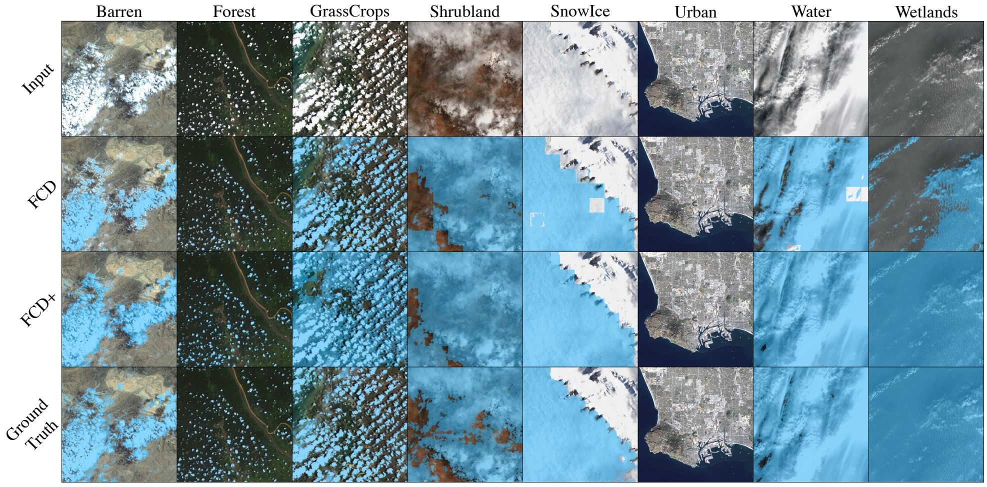


Figure 5.4: Example cloud mask results of FCD and FCD+ on Landsat-8 images from each biome.

Table 5.2: Cloud detection results for Landsat-8 Biome. The supervision type indicates: image-level labels $\mathscr{I}$, all available pixel-level labels $\mathscr{F}$, and 1% of available pixel-level labels $\mathscr{F}^{1\%}$. Our proposed method achieves the highest performance with the least expensive labels.

| Method | Supervision | F1-score | Accuracy |
|---|---|---|---|
| CFmask [34] | - | 87.2 | 87.7 |
| Random initialization | $\mathscr{F}^{1\%}$ | 89.4±0.5 | 89.5±0.5 |
| Pretrained | $\mathscr{I} + \mathscr{F}^{1\%}$ | 91.5±0.4 | 91.6±0.4 |
| Fully supervised [72] | $\mathscr{F}$ | 93.9±0.5 | 94.0±0.5 |
| FCD+ (ours) | $\mathscr{I}$ | **91.5±0.6** | **91.7±0.6** |
| FCD+1% (ours) | $\mathscr{I} + \mathscr{F}^{1\%}$ | **93.4±0.5** | **93.5±0.4** |

### Results of FCD+

We train FCD+ with the pseudo masks generated by FCD for the training data and evaluate its performance on the Landsat-8 Biome *test* set. We compare FCD+ with the rule-based algorithm CFMask [34], whose cloud masks are currently distributed with Landsat-8 images in the QA layer. Furthermore, we compare FCD+ with the same U-Net model but trained in a fully-supervised manner with actual pixel-level labels, which gives us an upper bound on the best performance achievable by the chosen network architecture.

Finally, we evaluate the performance of FCD+ as a weakly-supervised pretrained model in a semi-supervised setting, where 1% of data in our training set is labeled with pixel-level labels. We refer to this model as FCD+1%. This result shows how FCD+ applies to a real-world scenario, where one might allow an increased annotation effort for improved cloud detection performance. We compare FCD+1% against two baselines for the same underlying model: A model with randomly initialized weights is trained with the 1%, and a model with pre-trained weights using the available image-level labels to train for cloud classification and then fine-tuned with the 1%.

**Quantitative Results** Table 5.2 shows our results for the Landsat-8 Biome test set. Compared to FCD, FCD+ greatly increases F1 scores, which shows its ability to refine FCD-generated cloud masks. Moreover, our weakly-supervised FCD+, requiring *only* image-level labels for training, outperforms the existing rule-based method CFMask [34] by $+4.2\%$ in F1-score, only $-2.4\%$ below what is achievable by existing fully-supervised methods that use 100% of available pixel-level labels for supervision.

By fine-tuning FCD+ with 1% of available labels (FCD+1%), we reduce the gap to only $-0.5\%$ of fully-supervised performance. In comparison to the existing two pre-training strategies “random initialization" and “pre-trained", pre-training models with FCD-generated cloud masks are highly beneficial as FCD+1% outperforms both.

This shows that even though FCD-generated cloud masks contain artifacts, using them to pre-train existing supervised models enables them to achieve higher cloud detection performance than what is previously possible when only image-level labels and 1% pixel-level labels are available.

**Qualitative Results** Figure 5.4 illustrates the cloud masks of FCD+ in comparison to FCD for each type of biome. We find that the issues of FCD are resolved, removing the generative artifacts in the Shrubland and Wetlands examples, as well as the patch-shaped holes in the Shrubland, SnowIce, and Water examples. Figure 5.3 further illustrates the improvements for input patches. Compared to the FCD cloud masks, the outputs of FCD+ are less noisy and better resemble the ground truth. We also show the result of fine-tuning with 1% of pixel-level labels: FCD+1% better separates individual clouds, further improving the results.

## 5.7 Conclusion

In this work, we proposed FCD and FCD+ for weakly-supervised cloud detection. Existing supervised CNN-based cloud detection methods require large amounts of training images with pixel-level cloud labels, which brings significant labeling costs. As a result, applying existing CNN-based methods to detect clouds in the growing number of Earth observation satellites is highly expensive when pixel-level labels are not available. To alleviate this issue, we propose FCD, a weakly-supervised cloud detection method that requires only image-level labels, which are significantly cheaper to acquire. FCD applies a Fixed-Point GAN to learn image-to-image translation between clear and cloudy images while ensuring only clouds are affected during translation. By translating images to clear, thus removing any clouds, we are able to detect clouds at the pixel level from the difference between the original image and the translated image. As FCD is a generative model, we additionally propose FCD+ to refine the generated cloud masks of FCD, leading to further improvements by removing generative side effects. On the large Landsat-8 Biome dataset with satellite images from various biomes around the globe, we demonstrate our method outperforms existing rule-based methods as well as weakly-supervised methods based on class activation maps in cloud detection. Furthermore, FCD+ achieves near fully-supervised performance after fine-tuning with only 1% of available pixel-level labels. Our proposed method thus enables a drastic reduction in labeling efforts for training CNN-based cloud detectors with minimal performance loss.

## 5.8 Acknowledgements

The work of Joachim Nyborg was funded by the *Innovation Fund Denmark* under reference *8053-00240*.

# Chapter 6

# TimeMatch: Unsupervised Cross-Region Adaptation by Temporal Shift Estimation

JOACHIM NYBORG, AARHUS UNIVERSITY, DENMARK
CHARLOTTE PELLETIER, UNIVERSITÉ BRETAGNE SUD, FRANCE
SÉBASTIEN LEFÈVRE, UNIVERSITÉ BRETAGNE SUD, FRANCE
IRA ASSENT, AARHUS UNIVERSITY, DENMARK

Revised manuscript submitted to the *ISPRS Journal of Photogrammetry and Remote Sensing*.

**Abstract**

The recent developments of deep learning models that capture complex temporal patterns of crop phenology have greatly advanced crop classification from Satellite Image Time Series (SITS). However, when applied to target regions spatially different from the training region, these models perform poorly without any target labels due to the temporal shift of crop phenology between regions. Although various unsupervised domain adaptation techniques have been proposed in recent years, no method explicitly learns the temporal shift of SITS and thus provides only limited benefits for crop classification. To address this, we propose TimeMatch, which explicitly accounts for the temporal shift for improved SITS-based domain adaptation. In TimeMatch, we first estimate the temporal shift from the target to the source region using the predictions of a source-trained model. Then, we re-train the model for the target region by an iterative algorithm where the estimated shift is used to generate accurate target pseudo-labels. Additionally, we introduce an open-access dataset for cross-region adaptation from SITS in four different regions in Europe. On our dataset, we demonstrate that TimeMatch outperforms all competing methods by 11% in average F1-score across five different adaptation scenarios, setting a new state-of-the-art in cross-region adaptation. Our source code and dataset are available at `https://github.com/jnyborg/timematch`.

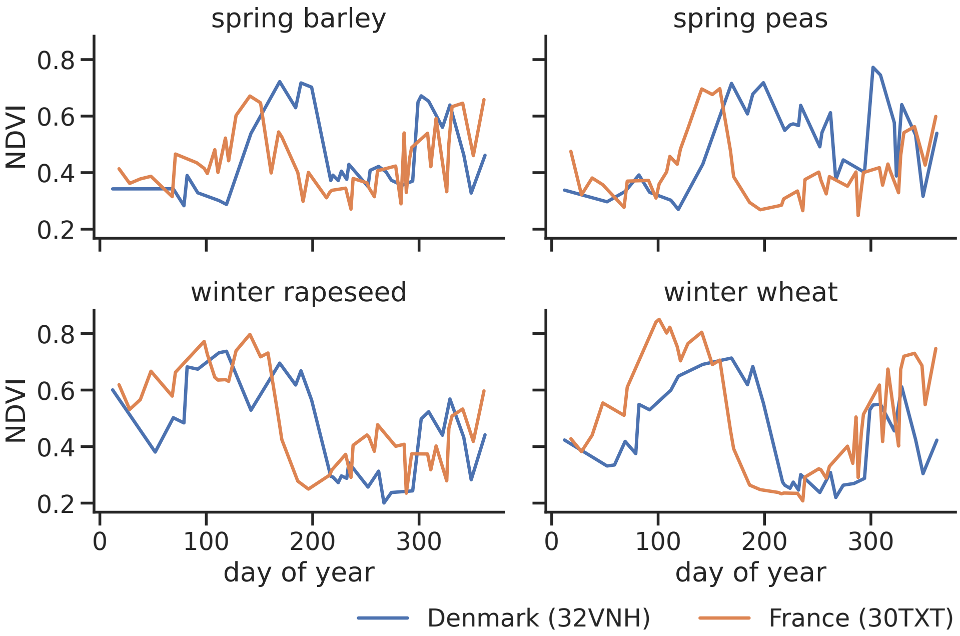


Figure 6.1: Normalized difference vegetation index (NDVI) time series for crops from two different Sentinel-2 tiles in Europe, indicating the growth of four crop types. Crops develop similarly in different regions, but the patterns are temporally shifted, *e.g.* if crops ripen at different times of the year.

## 6.1 Introduction

Today, the availability of satellite image time series (SITS) data is rapidly increasing. For instance, the twin Sentinel-2 satellites provide imagery of the entire Earth every two to five days [29]. A frequent acquisition of images is crucial for vegetation-related remote sensing applications such as crop type classification [132, 174]. Multi-temporal data enables capturing the phenological development of crops (*i.e.*, the progressions of crop growth), a key dimension to discriminate each crop type [119]. Recently, the increasing availability of SITS along with advances in deep learning has led to crop classifiers with temporal neural architectures using convolutions [125, 193], recurrent units [65, 108, 112, 137], self-attention [136, 142], or combinations thereof [67, 138].

These crop classification models achieve impressive performance by capturing the temporal structure of the problem but rely on the existence of a large amount of labeled training data. While unlabeled SITS are plenty, access to labels in the region of interest (the *target* domain) is often either costly or otherwise unavailable. A possible solution is to train a model in a region with labels available (the *source* domain) and apply it to the unlabeled target region. However, when the two regions are geographically different, the dissimilarity between the source and target data distributions can cause a source-trained model to perform poorly when applied to the target region [85, 101, 168].

Addressing the distributional shift problem to adapt a source-trained model to an unlabeled target domain is in machine learning known as unsupervised domain adaptation (UDA) [79, 120, 168]. Here, we consider the cross-region UDA problem for SITS [180], where we are provided with labeled data from a source region and unlabeled data from a target region. In this setting, the source and target data distributions differ due to changes in local conditions, such as the soil, climate, and farmer practices, which cause spectral and temporal shifts [168].

Addressing the temporal shift is of particular importance when adapting crop classifiers to new regions, as we illustrate in Figure 6.1. While crops in different regions have similar growth patterns, the timing of key growth stages, such as the peak of greenness, is shifted along the temporal axis. As crops are classified primarily by their unique growth patterns, the temporal shift may cause misclassifications when a source-trained model is applied to a target region. For example, the shift in time could cause the phenology of spring barley to appear similar to that of winter barley in the target. Thus, accounting for the temporal shift is a key factor in cross-region adaptation.

A possible approach could be to train models that are invariant to temporal shifts, such as by applying random temporal shifts to the training data. However, as the temporal shift could be the main feature that separates two crop types, shift-invariant models have reduced classification ability compared to shift-variant models.

Another approach is to apply existing UDA methods. Typically, these methods address domain adaptation by constraining the classifier to operate on domain-invariant features [8]. This is achieved by training the classifier to perform well on the source domain while minimizing a divergence measure between features extracted from the source and the target domains [37, 170, 180]. While these methods have been successfully applied in various applications [79, 184], they do not directly account for the temporal shift in SITS and have thus been reported to provide limited benefits in cross-region UDA [100]. More recently, self-training methods have emerged as a promising alternative to domain-invariant methods [16, 110, 143, 153, 200]. Self-training iteratively generates pseudo-labels [91] for the target domain and then uses them to retrain the model with target data. To account for noisy pseudo-labels caused by the domain shift, these methods typically incorporate a refinement step where the noise is reduced in various ways, such as with generative models [110] or learned confusion matrices [16]. Still, no method considers the particular case of SITS where the pseudo-label noise is caused by a temporal shift.

In this paper, we propose *TimeMatch*, a self-training method for cross-region UDA where we directly account for the temporal shift. TimeMatch consists of two components: (i) the temporal shift estimation and (ii) the TimeMatch learning algorithm.

Estimating the temporal shift directly from the target data is difficult, as the lack of labels hinders *e.g.* the comparison of class-wise vegetation indices as in Figure 6.1. To address this, we propose an unsupervised method where we estimate the temporal shift from target to source with a source-trained model. First, we obtain the softmax predictions of the model when input target data with different temporal shifts. Then, we choose the temporal shift with high prediction confidences across a diverse set of classes. We show that this approach corresponds well to the actual climatic differences between the two regions. Moreover, as correctly classified examples tend to have higher prediction confidence [61], the estimated shift enables us to generate more accurate pseudo-labels in the target domain for self-training.

In TimeMatch learning, we therefore use self-training to adapt a model to the target domain. We propose an iterative algorithm where we alternate between temporal shift estimation and re-training the model for the target domain by learning from both labeled source data and pseudo-labeled target data. By doing so, the model learns discriminative target features for accurate crop classification in the target region.

Lastly, we present the TimeMatch dataset, a challenging new open-access dataset for training and evaluating cross-region models on SITS with over 300.000 annotated parcels from four different regions in Europe. Evaluated on this dataset, our approach outperforms all competing methods by 11% F1-score on average across five different cross-region UDA experiments.

In summary, our contributions are as follows:

- We propose a method for estimating the temporal shift between a labeled source region and an unlabeled target region to reduce their temporal discrepancy.
- We propose *TimeMatch*, a novel UDA method designed for the cross-region problem of SITS, where crop classification models are adapted to an unlabeled target region by self-training on temporally shifted data for improved performance compared to existing methods. Our source code is available at `https://github.com/jnyborg/timematch`.
- We release the TimeMatch dataset [116], a new dataset for training and evaluating cross-region UDA models on SITS from four different European regions.

This paper is organized as follows. Section 6.2 describes the existing literature related to our work. Section 6.3 describes the proposed method for temporal shift estimation and the TimeMatch learning algorithm. Section 6.4 presents our dataset and the experimental setup, and Section 6.5 the experimental results. Lastly, Section 6.6 concludes this work.

## 6.2 Related Work

TimeMatch is related to the existing work in unsupervised domain adaptation of learning domain-invariant features, time-series adaptation, cross-region adaptation, and self-training.

### Domain-Invariant Methods

The predominant approach in UDA is to train the classifier to rely only on domain-invariant features [8, 184]. To this end, several works consider adversarial training [37, 38, 98]. In domain adversarial neural networks (DANN) [37, 38], the feature extractor is adversarially trained to produce domain-invariant features that are indistinguishable by a domain discriminator. Conditional domain adversarial networks (CDAN) [98] improves upon DANN by conditioning the domain discriminator on classifier predictions in addition to features to enable the alignment of multimodal data distributions.

Another approach is to align the feature distributions directly by minimizing a divergence measure. Choices for divergence measure include maximum mean discrepancy (MMD) [170], correlation alignment [161], or optimal transport [24, 31]. Recently, JUMBOT [31] achieves state-of-the-art UDA results by using mini-batch unbalanced optimal transport to minimize the domain discrepancy of joint deep feature and label distributions.

While domain-invariant methods achieve strong results on computer vision datasets, they do not explicitly handle the temporal dimensions of SITS data and time series in general.

### Time-Series Unsupervised Domain Adaptation

Few methods tackle the challenge of time series UDA. Current methods for time series are typically also based on learning domain-invariant features [4, 126, 185]. Recurrent domain adversarial neural network (R-DANN) and variational recurrent adversarial deep domain adaptation (VRADA) explore long short-term memory and variational recurrent neural networks as feature extractors, respectively, and learn domain-invariant features using the DANN method [126]. Likewise, the convolutional deep domain adaptation model for time series data (CoDATS) learns domain-invariant features with a temporal convolutional network with the DANN method [185]. However, while these methods are effective at learning domain-invariant features for time series, they are not designed to learn the temporal shift present in SITS.

### Cross-Region Crop Classification

Lucas *et al.* [100] reports that existing UDA methods, including existing domain-invariant methods [32, 49], perform poorly when applied to cross-region UDA of SITS due to the temporal shift problem and the change in class distribution between the two regions. Recently, Wang *et al.* [180] proposed the phenology alignment network (PAN) as the first method for cross-region UDA of SITS. PAN learns domain-invariant features with MMD [170] and a feature extractor consisting of gated recurrent units and self-attention. Still, as PAN learns domain-invariant features, the temporal shift problem is not directly addressed.

### Self-Training Methods

Semi-supervised learning (SSL) is a similar task to domain adaptation, but where the labeled and unlabeled data are sampled from the same data distribution [14].

Many SSL methods are based on pseudo-labeling [91] (also called self-training [157]), where the model's own high-confidence predictions are used as labels for the unlabeled samples. In Mean Teacher [165], the model assumes a dual role as *teacher* and *student*. The student is updated by gradient descent with pseudo-labels generated by the teacher, whereas the teacher is updated by an exponential moving average (EMA) of student parameters to reduce pseudo-label noise. FixMatch [157] generates pseudo-labels for weakly-augmented inputs, and uses confident pseudo-labels to self-train the model on strongly-augmented inputs, regularizing the model to output consistent pseudo-labels for random augmentations of the input.

Recently, self-training has emerged for UDA as an alternative to domain-invariant methods [16, 110, 143, 153, 200]. By learning from both labeled source data and pseudo-labeled target data, self-training methods implicitly encourage feature alignment for each class without restricting the model to operate on domain-invariant features. However, since the domain shift often results in increased pseudo-label noise compared to SSL, existing methods introduce various refinement methods to reduce the noise, such as co-training [17], tri-training [143], conditional generative models [110], or confidence regularization [200]. Recently, Adversarial-Learned Loss for Domain Adaptation (ALDA) [16] proposes to refine the pseudo-labels with a noise-correcting domain discriminator.

Similar to this line of work, our approach is based on self-training. By directly accounting for the temporal shift, we can temporally align the target SITS with that of the source, which enables the generation of more accurate pseudo-labels compared to existing self-training methods that do not.

## 6.3 TimeMatch

In this section, we describe our proposed method TimeMatch for cross-region UDA. We begin by formally defining the problem setting, followed by an overview of how TimeMatch addresses it. We then give the details of the two TimeMatch components: temporal shift estimation and TimeMatch learning.

### Problem Setting

In crop classification, the input is a sequence of satellite images $\boldsymbol{x}_i = (\boldsymbol{x}_i^{(1)}, \dots, \boldsymbol{x}_i^{(T_i)})$ of length $T_i$ to be classified into one of the $K$ crop classes. In object-based classification, which we focus on in this work, each $\boldsymbol{x}_i \in \mathbb{R}^{T_i \times N_i \times C}$ contains a sequence of $N_i$ pixels of $C$ spectral bands within a homogeneous, agricultural plot of land which we refer to as a *parcel*.

Each $\boldsymbol{x}_i$ is accompanied by a sequence $\boldsymbol{\tau}_i = (\tau_i^{(1)}, \dots, \tau_i^{(T_i)})$ indicating the time $\tau_i^{(j)}$ at which each observation $\boldsymbol{x}_i^{(j)}$ is sampled. In practice, $\tau_i^{(j)}$ is typically represented by the day-of-year [137, 142], and makes it possible to account for the irregular temporal sampling of most satellites. The goal of the crop classification task is to learn a

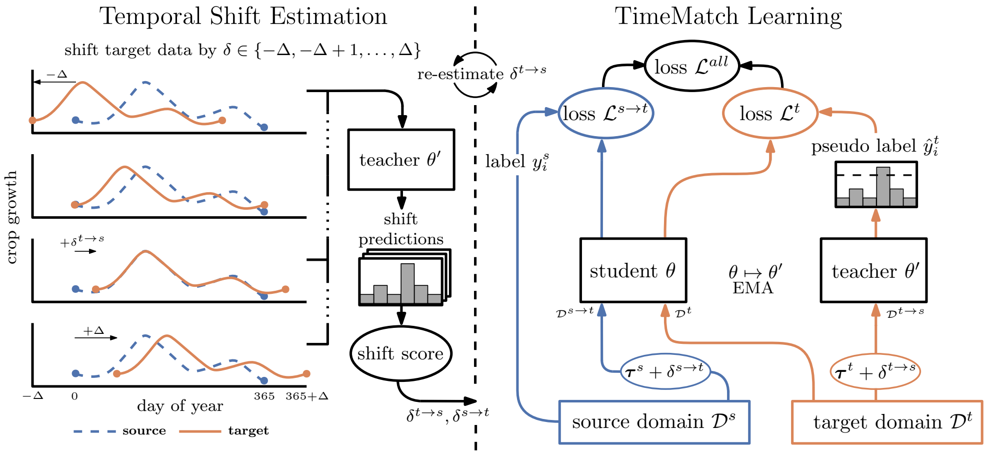

Temporal Shift Estimation
shift target data by $\delta \in \{-\Delta, -\Delta+1, \dots, \Delta\}$
$-\Delta$
$+\delta^{t\to s}$
$+\Delta$
crop growth
$-\Delta$
0
365
365+$\Delta$
day of year
source
target
teacher $\theta'$
shift predictions
shift score
$\delta^{t\to s}, \delta^{s\to t}$
re-estimate $\delta^{t\to s}$
label $y_i^s$
TimeMatch Learning
loss $\mathcal{L}^{all}$
loss $\mathcal{L}^{s\to t}$
loss $\mathcal{L}^{t}$
pseudo label $\hat{y}_i^t$
student $\theta$
$\theta \mapsto \theta'$
EMA
teacher $\theta'$
$\mathcal{D}^{s\to t}$
$\mathcal{D}^{t}$
$\mathcal{D}^{t\to s}$
$\boldsymbol{\tau}^s + \delta^{s\to t}$
$\boldsymbol{\tau}^t + \delta^{t\to s}$
source domain $\mathcal{D}^s$
target domain $\mathcal{D}^t$

Figure 6.2: Overview of TimeMatch. Both the student and teacher are pre-trained on the source domain. *Temporal Shift Estimation*: We input shifted target data to the teacher model and obtain its predictions for each shift. We then score each shift by the confidence and diversity of the teacher predictions, and the shift with the best score is output as the temporal shift estimate $\delta^{t\to s}$ and $\delta^{s\to t} = -\delta^{t\to s}$. *TimeMatch Learning*: The teacher generates pseudo-labels for unlabeled target data shifted by $\delta^{t\to s}$. Then, the student is updated for (non-shifted) target data using the pseudo-labels, and for source data shifted by $\delta^{s\to t}$ using the available source labels. As a result, the student is adapted to the target domain with both generated target labels and actual source labels. After the student parameters have been updated with gradient descent, the teacher parameters are updated as an exponential moving average (EMA) of the student parameters. As both models adapt to the temporal shift of the target domain, the best shift for pseudo-labeling with the teacher changes and must be re-estimated. The EMA ensures the teacher adapts slowly which enables $\delta^{t\to s}$ to be re-estimated each epoch only for improved training efficiency and pseudo-label accuracy.

model which predicts class probabilities $p(y|(\boldsymbol{x}_i, \boldsymbol{\tau}_i)) \in \mathbb{R}^K$, typically learned with supervision from labels $y \in \{1, \dots, K\}$.

In this work, we consider the problem of cross-region UDA. We are given a source domain $\mathscr{D}^s = \{(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s, y_i^s)\}_{i=1}^{n^s}$ of $n^s$ labeled SITS and a target domain $\mathscr{D}^t = \{\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t\}_{i=1}^{n^t}$ of $n^t$ unlabeled SITS. We assume both the source and target domains consist of SITS acquired over a single year (January 1 to December 31) and in geographically different locations. The two domains can be associated with different data distributions, as changes in local conditions, *e.g.* soil, weather, climate, or farmer practices, cause domain discrepancies [168]. In this work, we focus on the domain discrepancies caused by temporal shifts (Section 6.1). Although not explicitly addressed in this work, there are other sources of discrepancies that might occur. For example, the local topography or soil conditions could impact not only the temporal development of crop growth, but also the spectral values, which could change the spectral signature of the same crop type in different regions.

Because of these data discrepancies, models which are trained with the labeled source domain can fail when applied to the unlabeled target domain [101], thus hindering the large-scale application of crop classifiers. To this end, our goal is to adapt a model trained on $\mathscr{D}^s$ to make accurate predictions on $\mathscr{D}^t$. We do so by explicitly estimating the temporal shift between the two regions to generate accurate pseudo-labels for $\mathscr{D}^t$. Then, we re-train the model with target data using the pseudo-labels, thereby adapting the model to the spectral and temporal properties of the target region. We note that the classes in the source may not be exactly the same as the classes in the target. This complicates UDA, which typically assumes a closed-set setting [121], where the set of classes in the source and target domains are equal. For simplicity, we focus on a closed-set setting by adapting a classifier trained for the main $K-1$ crop types in the source region, plus an "unknown" class containing all remaining source data. This ensures that all target examples can be classified to either one of the $K-1$ crop classes or "unknown".

## Approach Overview

Here we give an overview of how TimeMatch addresses the cross-region UDA problem before describing the full details. A visual presentation of TimeMatch is given in Figure 6.2. TimeMatch consists of two components (i) temporal shift estimation and (ii) TimeMatch learning.

We aim to estimate the temporal shift between the source and target regions to reduce their domain discrepancy (see Figure 6.1). We represent the temporal shift by a scalar $\delta^{t\rightarrow s} \in \mathbb{Z}$ (as the number of days), here in the direction from target to source. Note that the shift in the opposite direction is obtained by $\delta^{s\rightarrow t} = -\delta^{t\rightarrow s}$, so we only have to estimate one shift. To shift the target domain by $\delta^{t\rightarrow s}$, we write $\boldsymbol{\tau}^t + \delta^{t\rightarrow s}$, meaning $\delta^{t\rightarrow s}$ is added element-wise to each target day-of-year. With our proposed method for temporal shift estimation (Section 6.3), we obtain estimates for $\delta^{t\rightarrow s}$ and $\delta^{s\rightarrow t}$.

In TimeMatch learning (Section 7), we use $\delta^{s\rightarrow t}$ to construct a target-shifted source domain $\mathscr{D}^{s\rightarrow t} = \{(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s + \delta^{s\rightarrow t}), y_i^s\}_{i=1}^{n^s}$, which has reduced domain discrepancy to the unlabeled target domain $\mathscr{D}^t$ due to the temporal alignment. We therefore use self-training to learn from the labeled $\mathscr{D}^{s\rightarrow t}$ and unlabeled $\mathscr{D}^t$. To do so, TimeMatch learning unifies temporal shift estimation with the loss function of FixMatch [157] and the exponential moving average (EMA) training of Mean Teacher [165], as we explain next.

We first obtain source-trained parameters by training a crop classifier with $\mathscr{D}^s$. We then duplicate the trained classifier into two models: the *teacher* and the *student*. Our TimeMatch learning algorithm aims to adapt both the teacher and the student to the new target region with self-training. The teacher generates pseudo-labels for the target domain to train the student, and the knowledge learned by the student is then updated back to the teacher, thus the pseudo-labels used to train the student itself are improved. We generate pseudo-labels by using $\delta^{t\rightarrow s}$ to create an adapted target domain $\mathscr{D}^{t\rightarrow s} = \{\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t + \delta^{t\rightarrow s}\}_{i=1}^{n^t}$. As $\mathscr{D}^{t\rightarrow s}$ is temporally aligned with $\mathscr{D}^s$, the source-initialized teacher generates more accurate pseudo-labels for $\mathscr{D}^{t\to s}$ than $\mathscr{D}^t$. The student is then trained with labeled $\mathscr{D}^{s\to t}$ and pseudo-labeled $\mathscr{D}^t$ via the FixMatch loss [157], thereby leveraging both the available source labels and the target pseudo-labels to adapt the student to the target domain.

After updating the student, the teacher is updated via an EMA of the student parameters. As the two models adjust to the temporal shift of the target domain, the best shift $\delta^{t\to s}$ for pseudo-labeling with the teacher gradually moves to zero during TimeMatch learning. To adjust to the changing shift and ensure the pseudo-labels are consistently accurate, it is necessary to re-estimate the temporal shift of the teacher as it learns. However, repeating temporal shift estimation is computationally expensive, and drastically increases training time if done each training iteration. Therefore, in Section 16, we discuss how EMA training alleviates this issue by enabling the re-estimation to be done only once per epoch.

Next, we first describe our method for estimating the temporal shift before describing the loss function and learning algorithm of TimeMatch learning.

### Temporal Shift Estimation

Estimating the temporal shift directly from the data is difficult, as labels are not available in the target domain. Without labels, we cannot separate the target data into each crop type, which prevents the computation of *e.g.* vegetation indices to compare the source and target phenology of each crop type directly.

Instead, we propose to estimate the temporal shift by calculating statistics on the predictions of a source-trained model when input temporally shifted target data. By doing so, we estimate the shift that aligns the target data with the source crop phenology learned by a model, leveraging the classification ability of the trained model to estimate the shift from unlabeled data. Another benefit of this approach is that it enables re-estimation of the best temporal shift for pseudo-labeling as the learned phenology of the model changes from source to target in TimeMatch learning.

One possible value to measure is the confidence of the model predictions. Intuitively, when a source-trained model is applied to correctly shifted target data, it should output more confident predictions than for incorrectly shifted target data. As correctly classified examples tend to have more confident predictions than wrongly classified or out-of-distribution examples [61], we argue that a confident temporal shift indicates a better alignment of the target domain with the source which results in accurate pseudo-labels and reduced domain discrepancy.

The confidence of a model for a particular shift $\delta^{t\to s}$ can be measured by the expected entropy:

$$\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)\sim\mathscr{D}^t}\left[H\big(p_\theta\big(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\big)\big)\right], \tag{6.1}$$

where $H$ denotes the entropy, here computed over the predictions of the model $\theta$ when input temporally shifted target data sampled from $\mathscr{D}^t$.

To estimate a temporal shift with entropy, Equation 6.1 should be computed for each possible shift $\delta^{t\to s}\in\{-\Delta,-\Delta+1,\ldots,\Delta\}$, and the estimated shift is then the one with lowest entropy. Here, $\Delta$ defines the maximum possible shift (in days) to estimate between the source and target regions.

However, due to the class imbalance of SITS, relying on expected entropy alone could result in choosing a shift where the model outputs confident predictions for only the most frequent classes while ignoring the less frequent classes. This would hinder the adaptation of the model for the less frequent target classes. To address this problem, the diversity of the predicted marginal distribution should also be considered in the estimation. The marginal is given by:

$$p_\theta(y) = \mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)\sim\mathscr{D}^t}\left[p_\theta\left(y|\left(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s}\right)\right)\right], \tag{6.2}$$

that is, the expected predictions of the model (parameterized by $\theta$) when input shifted target data.

Ideally, the marginal distribution should match the class distribution of the target domain, as this indicates a shift where the model predicts a diverse set of classes according to their actual frequency. But since target labels are unavailable, so is the target class distribution. Instead, inspired by metrics for evaluating image generative models, we consider two options to address this: the Inception score [145] (IS), and the activation maximization score [196] (AM). Both metrics consider the entropy and marginal of a pre-trained model, but IS scores the marginal distribution by its similarity to a uniform distribution, whereas AM uses the actual class distribution.

As these metrics were originally proposed to evaluate the quality of generated images, we describe next how we repurpose them for temporal shift estimation. Finally, we describe an algorithm where IS is used to initialize the temporal shift for estimating the target class distribution with pseudo-labels and enable a better temporal shift estimate with AM.

### Inception Score

IS is computed for a temporal shift $\delta$ by:

$$\begin{aligned}
&\mathrm{IS}(\delta^{t\to s},\theta)\\
&=\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)}\left[D_{\mathrm{KL}}\left(p_\theta\left(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\right)\parallel p_\theta(y)\right)\right] &\quad (6.3)\\
&=H(p_\theta(y))-\mathbb{E}_{(\boldsymbol{x}^t,\boldsymbol{\tau}^t)}\left[H\left(p_\theta\left(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta^{t\to s})\right)\right)\right] &\quad (6.4)
\end{aligned}$$

where $D_{\mathrm{KL}}(\cdot\parallel\cdot)$ is the KL-divergence between two distributions, here the conditional distribution $p_\theta\left(y|(\boldsymbol{x}^t,\boldsymbol{\tau}^t+\delta)\right)$ and marginal distribution $p_\theta(y)$ predicted with model parameters $\theta$. Higher values of IS indicate a better $\delta$, as when the conditional and marginal distributions are different, this corresponds to a temporal shift where the former has low entropy (*i.e.*, the model is confident), and the latter has high entropy (*i.e.*, the model predicts a diverse set of classes). Hence, the temporal shift $\delta^{t\to s}$ is estimated by:

$$\delta_{IS}^{t\to s}(\theta^s) = \underset{\delta^{t\to s}\in\{-\Delta,\ldots,\Delta\}}{\operatorname{argmax}} \mathrm{IS}(\delta^{t\to s},\theta^s), \tag{6.5}$$

where the estimated temporal shift maximizes IS for a source-trained model parameterized by $\theta^s$ when applied to target data.

**Algorithm 1:** ESTIMATETEMPORALSHIFT

```
1 Input: Source-trained parameters $\theta^s$, target domain $\mathscr{D}^t$, target class
    distribution estimate $\hat{C}^t$
2 if $\hat{C}^t = \mathbf{0}$ then
3     Estimate temporal shift $\delta^{t\to s} \leftarrow \delta_{IS}^{t\to s}(\theta^s)$ (Eq. 6.5)
4     Compute pseudo labels for each $(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t) \in \mathscr{D}^t$:
        $\hat{y}_i^t \leftarrow \text{argmax}_y \left(p_{\theta^s}(y|\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t + \delta^{t\to s})\right)$
5     Estimate class distribution $\hat{C}_y^t \leftarrow \frac{1}{n^t}\sum_{i=1}^{n^t} \mathbf{1}_{\hat{y}_i^t = y}$ for $y \in \{1,\dots,K\}$
6 Estimate temporal shift $\delta^{t\to s} \leftarrow \delta_{AM}^{t\to s}(\theta^s, \hat{C}^t)$ (Eq. 6.7)
7 Output: Temporal shift $\delta^{t\to s}$
```

### AM Score

A shortcoming of IS is that the highest score is achieved when $p_\theta(y)$ is uniform [5], which corresponds to an even distribution of classes in the target domain. For SITS, where the class distribution is often highly imbalanced, this may cause IS to estimate a suboptimal shift. AM [196] addresses this issue by taking the actual target class distribution $C^t$ into account:

$$\begin{aligned} \text{AM}(\delta^{t\to s}, \theta, C^t) = \mathbb{E}_{(\boldsymbol{x}^t, \boldsymbol{\tau}^t)} \left[ H\left(p_\theta(y|(\boldsymbol{x}^t, \boldsymbol{\tau}^t + \delta^{t\to s}))\right)\right] \\ + D_{\text{KL}}(C^t \parallel p_\theta(y)). \end{aligned} \tag{6.6}$$

AM consists of two terms: the first term is an entropy term on the conditional distribution, and the second is the KL-divergence between the underlying class distribution $C^t$ and the marginal distribution. Lower values of AM indicate a better $\delta$, as the model is confident in its predictions, and the actual class distribution of the data matches the predicted distribution of classes. The temporal shift $\delta^{t\to s}$ is estimated by:

$$\delta_{AM}^{t\to s}(\theta^s, C^t) = \underset{\delta^{t\to s} \in \{-\Delta, \dots, \Delta\}}{\text{argmin}} \text{AM}(\delta^{t\to s}, \theta^s, C^t). \tag{6.7}$$

where the estimated temporal shift minimizes AM.

### Algorithm for Estimating Temporal Shift

While AM is more accurate at estimating the temporal shift, it requires knowledge of the target class distribution $C^t$, which is not available. To address this, we propose to approximate the target class distribution for AM by pseudo-labels obtained with IS. We show our approach in Algorithm 1. First, we use IS (Equation 6.5) to estimate an initial shift $\delta^{t\to s}$ (line 3). This initial estimate allows us to shift the target domain so that more accurate pseudo-labels can be generated with a source-trained model. We then use the pseudo-labels to estimate the target class distribution $\hat{C}^t$ (lines 4-5). Finally, we re-estimate the temporal shift more accurately with AM and $\hat{C}^t$ (line 6).

### TimeMatch Learning

**Algorithm 2:** TIMEMATCH

```
Input: Labeled source domain $\mathscr{D}^s$, unlabeled target domain $\mathscr{D}^t$,
  source-trained parameters $\theta^s$, total epochs $n$ and iterations $m$, pseudo label
  threshold $\varepsilon$, trade-off value $\lambda$, EMA decay rate $\alpha$, learning rate $\eta$
Initialize student parameters $\theta \leftarrow \theta^s$ and teacher parameters $\theta' \leftarrow \theta^s$
Initialize estimated target class distribution $\hat{C}^t = \mathbf{0}$
for epoch = 1 to $n$ do
    Estimate temporal shift with teacher: $\delta^{t \to s} \leftarrow$
      ESTIMATETEMPORALSHIFT($\theta'$, $\mathscr{D}^t$, $\hat{C}^t$)
    if epoch = 1 then
        Initialize $\delta^{s \to t} \leftarrow -\delta^{t \to s}$
    for iteration = 1 to $m$ do
        Sample mini-batches of size $B$ from source $\mathscr{S} = \{(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s, y_i^s)\}_{i=1}^B$ and
          target $\mathscr{T} = \{(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t)\}_{i=1}^B$
        With $\mathscr{S}$ shifted by $\delta^{s \to t}$, compute source loss $\mathscr{L}^{s \to t}$ (Eq. 6.9)
        For each example in $\mathscr{T}$ shifted by $\delta^{t \to s}$, generate teacher prediction
          $\boldsymbol{q}_i^t$ and pseudo labels $\hat{y}_i^t$ (Eq. 6.10 and 6.11)
        With $\mathscr{T}$ and confident pseudo labels $\hat{y}_i^t$ with $\max(\boldsymbol{q}_i^t) > \varepsilon$, compute
          target loss $\mathscr{L}^t$ (Eq. 6.12)
        Update student by gradient: $\theta \leftarrow \theta - \gamma \nabla_\theta (\mathscr{L}^{s \to t} + \lambda \mathscr{L}^t)$
        Update teacher by EMA: $\theta' \leftarrow (1-\alpha)\theta + \alpha\theta'$
    Re-estimate class distribution: $\hat{C}_y^t \leftarrow \frac{1}{mB} \sum_i \mathbf{1}_{\hat{y}_i^t = y}$ for $y \in \{1, \ldots, K\}$ (using
      all pseudo labels from epoch)
Output: Student parameters $\theta$
```

With our method for estimating the temporal shift, we can reduce the domain discrepancy between the source and target domains. The TimeMatch learning algorithm uses the temporal shift to train the student model for the target domain from teacher-generated pseudo-labels via the FixMatch loss [157] and EMA training [165]. We present the complete TimeMatch algorithm in Algorithm 2, and describe the details of each step in the following.

### Pre-training on the Source Domain

As we rely on the teacher to generate pseudo-labels to train the student, it is important to obtain a good initialization for both models. Additionally, temporal shift estimation requires a source-trained model. Thus, we first use the labeled source domain to obtain source-trained model parameters $\theta^s$. Given a batch of labeled source data from

$\mathscr{D}^s$, we optimize the following loss function:

$$\mathscr{L}^s = \frac{1}{B}\sum_{i=1}^{B} L\big(p_{\theta^s}\big(y|(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s)\big), y_i^s\big), \tag{6.8}$$

where $L(\cdot,\cdot)$ is a classification loss (*e.g.* cross-entropy or focal loss [95]) and $B$ the batch size. After pre-training, we initialize the parameters of the student $\theta$ and teacher $\theta'$ from $\theta^s$ (line 2).

### TimeMatch Loss

The TimeMatch loss consists of two terms: a supervised loss $\mathscr{L}^{s\to t}$ applied to the adapted source domain $\mathscr{D}^{s\to t}$ and an unsupervised loss $\mathscr{L}^t$ applied to the unlabeled target domain $\mathscr{D}^t$. Our loss is based on the FixMatch loss [157]. To regularize the model to predict consistent pseudo-labels on randomly augmented versions of the same inputs, FixMatch applies two types of augmentation functions: *weakly*-augmented $a(\cdot)$ and *strongly*-augmented $A(\cdot)$, corresponding to simple and extensive augmentations of the input. We describe the form of augmentations we use for $a(\cdot)$ and $A(\cdot)$ in Section 6.4.

Let $\delta^{s\to t}$ and $\delta^{t\to s}$ be temporal shifts estimated given by Algorithm 1 using the teacher (line 5-7). To compute the supervised loss on the source domain, we use $\delta^{s\to t}$ to align the source domain with the target domain and optimize:

$$\mathscr{L}^{s\to t} = \frac{1}{B}\sum_{i=1}^{B} L\big(p_\theta\big(y|A(\boldsymbol{x}_i^s, \boldsymbol{\tau}_i^s + \delta^{s\to t})\big), y_i^s\big), \tag{6.9}$$

using source labels $y_i^s$ to update the student $\theta$ on strongly augmented source data shifted by $\delta^{s\to t}$. This loss makes it possible for the student to learn the target phenology from shifted source data (line 10).

To generate pseudo-labels for the target domain, we obtain the predicted class distribution from the teacher when input source-shifted target data:

$$\boldsymbol{q}_i^t = p_{\theta'}\big(y|a\big(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t + \delta^{t\to s}\big)\big), \tag{6.10}$$

where the teacher $\theta'$ is input a weakly-augmented target sample, shifted by $\delta^{t\to s}$. Then, we use

$$\hat{y}_i^t = \operatorname{argmax}(\boldsymbol{q}_i^t) \tag{6.11}$$

as pseudo-label (line 11). The student $\theta$ is then updated on strongly-augmented target data for confident pseudo-labels (line 10):

$$\mathscr{L}^t = \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}_{\max(\boldsymbol{q}_i^t)>\varepsilon} L\big(p_\theta\big(y|A\big(\boldsymbol{x}_i^t, \boldsymbol{\tau}_i^t\big)\big), \hat{y}_i^t\big), \tag{6.12}$$

where $\mathbf{1}$ is the indicator function, and $\varepsilon$ is the confidence threshold for using a pseudo-label. With this loss, the student is trained with target data using pseudo-labels. The total loss minimized by the student in TimeMatch is:

$$\mathscr{L}^{all} = \mathscr{L}^{s\to t} + \lambda \mathscr{L}^t, \tag{6.13}$$

where $\lambda$ is a scalar hyperparameter to control the trade-off between the supervised and the unsupervised loss (line 13).

### EMA training and re-estimating temporal shift

By optimizing $\mathscr{L}^{all}$, the student and teacher are trained only for the target phenology, as $\mathscr{L}^{s \rightarrow t}$ shifts the time of the source to the target, while $\mathscr{L}^{t}$ keeps the target in its original time. This loss enables a source-trained model to adapt to the crop phenology of the target domain.

However, by doing so, the source domain is gradually “forgotten", and as a result, it becomes unnecessary to apply the temporal shift $\delta^{t \rightarrow s}$ for pseudo-labeling the target domain with the teacher. This causes $\delta^{t \rightarrow s}$ to gradually move to zero during TimeMatch learning. Thus, if $\delta^{t \rightarrow s}$ is fixed to the same shift, the target samples will be wrongly shifted, which results in incorrect pseudo-labels. To address this, we re-estimate the temporal shift for the teacher during TimeMatch learning. As Algorithm 1 chooses the shift based on the confidence and diversity of model predictions, re-estimating the temporal shift with the teacher ensures the generated pseudo-labels remain accurate during training.

However, if the teacher is a direct copy of the student, the model will rapidly adapt to the target domain, which requires the temporal shift to be re-estimated every few iterations. But doing so drastically increases training time, as Equation 6.7 requires forwarding a large sample of target data for each possible temporal shift. We address this by introducing EMA training, where the teacher is slowly updated via an EMA of the student parameters (line 14):

$$\theta' \leftarrow (1-\alpha)\theta + \alpha\theta', \tag{6.14}$$

where $\alpha$ is a decay rate. By choosing $\alpha$ close to 1, we reduce the rate at which the teacher adapts to the target domain, enabling the re-estimation of $\delta^{t \rightarrow s}$ to be done only once each epoch (line 5). Moreover, by averaging model weights via EMA, we also obtain less noisy pseudo-labels [165].

By re-estimating the temporal shift, the teacher and the shift can both evolve jointly during training, resulting in better pseudo-labels for improved cross-region adaptation. Note that $\delta^{s \rightarrow t}$ is not re-estimated (line 7). The first shift estimate represents the shift of the data, whereas the re-estimated shift represents the shift of the teacher. By fixing $\delta^{s \rightarrow t}$ to the initial estimate, the source domain is kept aligned with the target domains during training, which enables semi-supervised learning.

## 6.4 Dataset and Materials

This section presents the TimeMatch dataset [116] and the materials for our experiments. We first introduce the crop classification model we use, followed by a description of the dataset and its pre-processing. Then, we describe the competitors and our implementation. Our source code is publicly available, and contains the

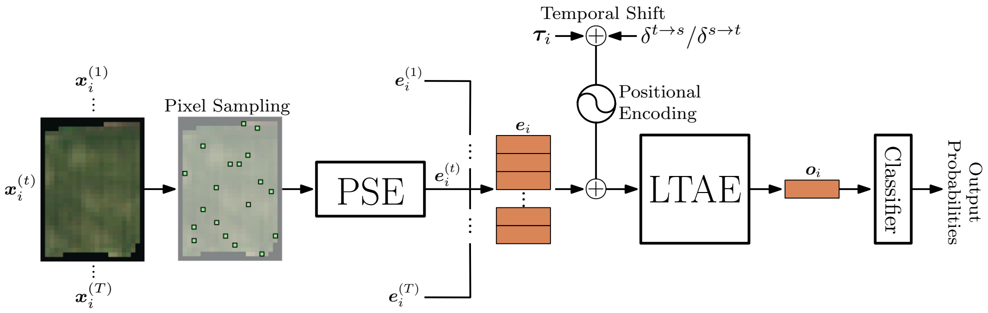


Figure 6.3: Overview of the PSE+LTAE model [141, 142]. Given SITS of an agricultural parcel, the PSE module process each time step independently by embedding a random sample of pixels. The results are then concatenated into a sequence of embeddings $\boldsymbol{e}_i$. The observation dates $\boldsymbol{\tau}_i$, which we add temporal shifts to, are input to the model by adding their positional encoding to $\boldsymbol{e}_i$. The result is temporally processed by LTAE to a single embedding $\boldsymbol{o}_i$ which is then passed to the classifier.

implementation of TimeMatch and the competitors, a link to download our dataset, and the full experimental results: `https://github.com/jnyborg/timematch`.

## Network Architecture

As model, we use the existing object-based crop classifier PSE+LTAE introduced by Sainte Fare Garnot *et al.* [141, 142]. The network consists of two modules: the pixel-set encoder (PSE) and the lightweight temporal attention encoder (LTAE). See Figure 6.3 for an overview.

The PSE module handles the spatial and spectral context of SITS. Given SITS of an agricultural parcel, PSE samples a random pixel-set of size $S$ among the $N_i$ available pixels within the parcel. The PSE is efficient compared to *e.g.* convolutions, which are time and memory-consuming when applied to irregularly sized parcels. As spatial information is lost by doing so, the PSE supports an optional extra input with various geometrical properties of the given parcel, such as its area. We do not input this extra feature to avoid biasing the model towards the shapes of parcels in the source region, which typically change depending on the local farmer practices. Thus, we only input the sequence $\boldsymbol{x}_i \in \mathbb{R}^{T_i \times N_i \times C}$, which is then embedded by the PSE for each time step independently.

The LTAE module [141] handles the temporal context by applying self-attention [173] with modifications to output a single embedding. It improves the accuracy and computational efficiency compared to the original TAE [142] by a channel grouping strategy and a learnable master query. The additional input $\boldsymbol{\tau}_i$ is input to LTAE by encoding the days with sinusoidal positional encoding [173] and adding the result to the output of PSE. As the positional encoding does not support negative inputs, we input negative temporal shifts by offsetting each $\boldsymbol{\tau}_i$ by the maximum temporal shift $\Delta$. Given the sequence of PSE-embeddings and the encoded $\boldsymbol{\tau}_i$, LTAE outputs a single embedding

$\boldsymbol{o}_i$, which is then classified by a multi-layer perceptron to produce class probabilities $p(y|(\boldsymbol{x}_i, \boldsymbol{\tau}_i)) \in \mathbb{R}^K$.

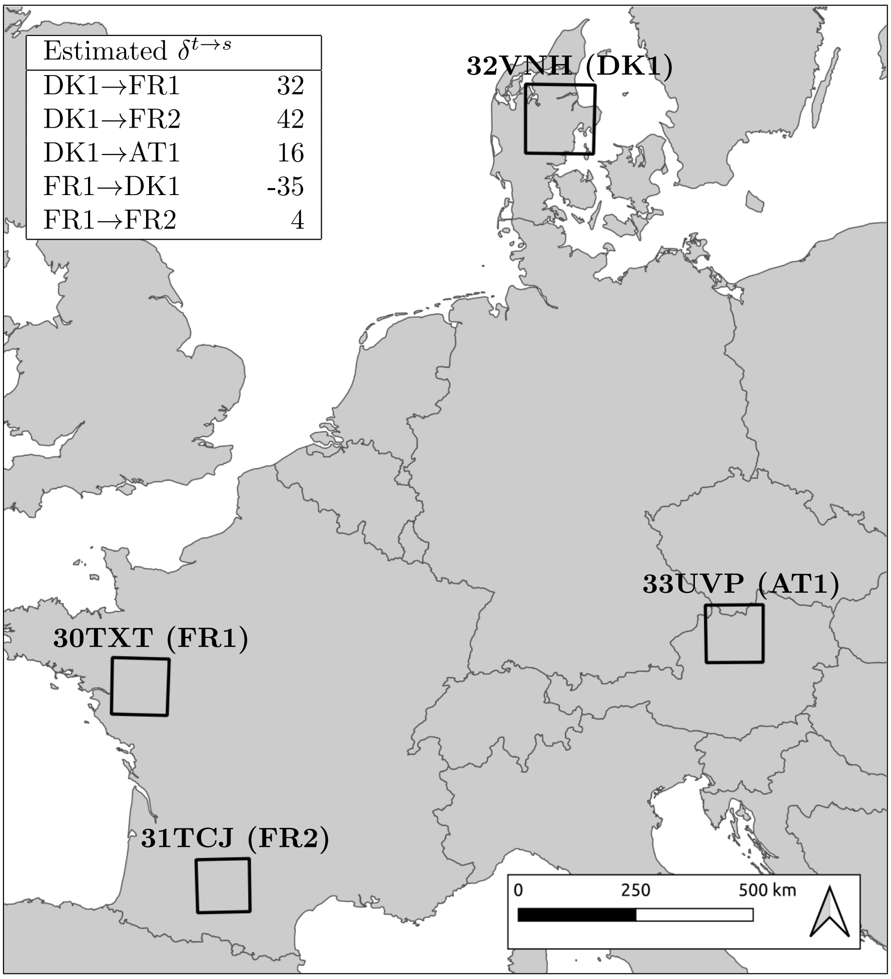
### The TimeMatch Dataset

The TimeMatch dataset [116] contains SITS from Sentinel-2 Level-1C products in top-of-atmosphere reflectance. Four Sentinel-2 tiles are chosen in various climates: 33UVP (Austria), 32VNH (Denmark), 30TXT (mid-west France), and 31TCJ (southern France), abbreviated as AT1, DK1, FR1, and FR2, respectively. A map of the tiles is shown in Figure 6.4. We use all available observations with cloud coverage $\leq 80\%$ and coverage $\geq 50\%$ between January 2017 and December 2017. Figure 6.5 shows the resulting acquisition dates for the four tiles. We leave out the atmospheric bands (1, 9, and 10), keeping $C = 10$ spectral bands. The 20m bands are bilinearly interpolated to 10m.

For ground truth data, we retrieve geo-referenced parcel shapes and their crop type labels from the openly available Land Parcel Identification System (LPIS) records in Denmark[1], France[2], and Austria[3]. We select 15 major crop classes in Europe and label any remaining parcels as unknown. Figure 6.6 shows the selected classes and their frequency in each tile.

We pre-process the parcels by applying 20m erosion and removing all parcels with an area of less than 1 hectare. This reduces label noise by removing pixels near the border of parcels, which are often less representative of the given crop class compared to the pixels in the middle, and also by removing small or thin polygons, which are typically miscellaneous classes such as field borders. The SITS are pre-processed for object-based classification by cropping the pixels within each parcel to input sequences $\boldsymbol{x}_i \in \mathbb{R}^{T_i \times N_i \times 10}$. Each input is then randomly assigned to the train/validation/test sets of each Sentinel-2 tile by a 70%/10%/20% ratio. Note that this process assumes knowledge of parcel shapes in the target region. If this is not available, TimeMatch may instead be applied for pixel-based classification by inputting single pixels ($S = 1$) to PSE+LTAE. We choose five different cross-region tasks (written as “source→target"): DK1→FR1, DK1→FR2, DK1→AT1, FR1→DK1, and FR1→FR2. When a Sentinel-2 tile is chosen as source, all labels of the train and validation sets are available for training. When a tile is the target region, no labels are available, except for the final evaluation on the test set. In contrast, many existing UDA methods assume that a labeled validation set is available for the target domain, and use it during training *e.g.* to select the best model [16, 38, 98, 170]. However, this assumption is unrealistic, as if labels were available in real-world scenarios, they would be better used for training the model. Instead, we report all cross-region UDA test results with the model output at the end of training. Still, hyperparameters must be chosen with a labeled validation set. Thus, we tune hyperparameters with the target validation set

[1] https://kortdata.fvm.dk/download (“Marker")
[2] http://professionnels.ign.fr/rpg (“RPG")
[3] https://www.data.gv.at (“INVEKOS Schläge")



| Estimated $\delta^{t \rightarrow s}$ | |
|---|---|
| DK1→FR1 | 32 |
| DK1→FR2 | 42 |
| DK1→AT1 | 16 |
| FR1→DK1 | -35 |
| FR1→FR2 | 4 |

Figure 6.4: Locations of the four European Sentinel-2 tiles in the TimeMatch dataset. In the upper left corner, we show the temporal shifts $\delta^{t \rightarrow s}$ estimated by Algorithm 1 with a source-trained model.

for only one task, DK1→FR1, and apply the found hyperparameters to all remaining tasks (as done in [31]).

The class distributions between regions differ significantly, and there may not be enough examples of a crop type in the source region for a model to learn their classification. Thus, when pre-training models on source data, we only use a subset of the available crop types with at least 200 examples in the source region (as indicated by the dashed line in Figure 6.6). The remaining classes are set as “unknown". When evaluating on the target data, we report results on the same selection of source classes no matter their frequency in the target.

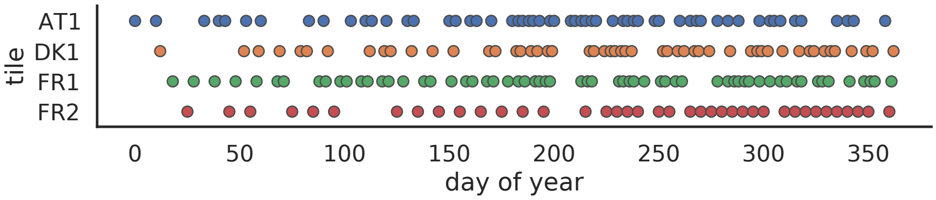


Figure 6.5: Acquisition dates for each Sentinel-2 tile in our dataset. The inputs are irregularly sampled with variable temporal length.

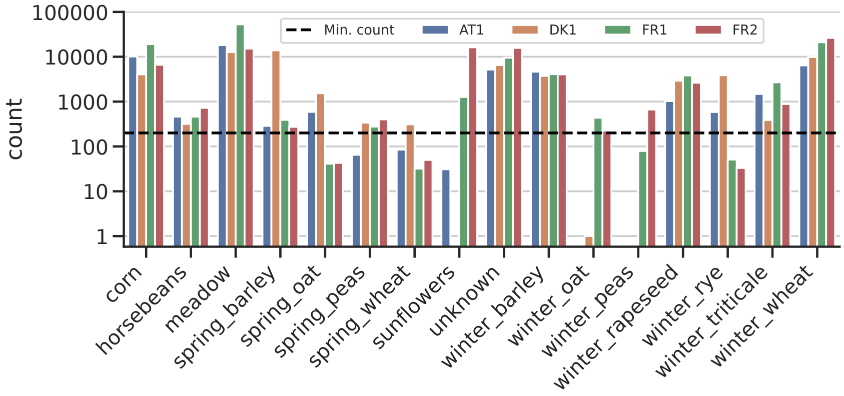


Figure 6.6: Class frequencies (log scale) for each Sentinel-2 tile in the TimeMatch dataset. The dashed line indicates the threshold for the source region when selecting a class as part of the $K$ classes.

## Comparisons

**Baselines** We consider the following baseline methods:

- *Source-Trained* is PSE+LTAE trained on the source domain and applied to the target domain without domain adaptation. This result represents the lower bound cross-region performance of the model.

- *Target-Trained* is PSE+LTAE trained with labeled target data using the same classes as the source-trained. We note that by training with the source classes, which is required for comparison, infrequent classes may not be learned properly which increases the variance of the results. This result can be seen as the upper bound cross-region performance if labels were available in the target region.

**Competing UDA Methods** We compare TimeMatch to five existing UDA methods. We reproduce these methods for SITS by replacing the original feature extractor with PSE+LTAE. For domain-invariant methods, we align the LTAE feature vector input to the final classifier (*i.e.*, $\boldsymbol{o}_i$ in Figure 6.3), similar to the original approach in these methods.

We compare to the following methods:

- *FixMatch* [157] is TimeMatch without the temporal shift estimation. As this method is semi-supervised learning, it shows whether UDA or SSL is more beneficial for cross-region adaptation.

- *MMD* [170] learns domain-invariant features by minimizing the maximum mean discrepancy metric.

- *DANN* [37] uses a domain classifier to learn domain-invariant features with adversarial training.

- *CDAN+E* [98] improves upon DANN by conditioning the domain classifier on the classification output and minimizing an entropy loss on target data.

- *ALDA* [16] is a self-training method where pseudo-labels are refined by a noise-correcting domain discriminator. This method is in essence the most similar to TimeMatch.

- *JUMBOT* [31] learns domain-invariant features by a discrepancy measure based on optimal transport.

We note that time-series domain adaptation methods R-DANN, VRADA [126] and CoDATS [185] also employ DANN to align the features extracted by temporal network architectures. Thus, the only difference between VRADA, CoDATS, and the DANN approach mentioned here is the backbone architecture, which in our case is the temporal model PSE+LTAE.

PAN [180], a UDA method for SITS, learns domain-invariant features by minimizing the MMD loss for a temporal crop classification network. Unfortunately, we were unable to gain access to the source code of PAN for comparison. As an alternative, we include the MMD comparison, which is similar to PAN, except the crop classifier is changed to PSE+LTAE.

**ShiftAug** To verify the benefits of estimating the temporal shift compared to training models that are invariant to temporal shifts, we implement a simple data augmentation technique to train shift-invariant models that we name *ShiftAug*. During training, ShiftAug uniformly samples $\delta \sim \mathscr{U}(-\Delta, \Delta)$ for each training example and shifts the example by $(\boldsymbol{x}_i, \boldsymbol{\tau}_i + \delta)$. By extending the training data to contain all valid temporal shifts with uniform probability, ShiftAug enables training models with invariance towards shifts. Note that ShiftAug is incompatible with the temporal shift estimation presented in Algorithm 1, which requires a shift-variant model.

We implement all competing methods with and without ShiftAug. This reveals the degree at which existing methods can implicitly learn shift-invariance.

### Implementation Details

All experiments are implemented in PyTorch [122] and trains on a single NVIDIA 1080 Ti GPU. Our implementation is based on the source code of PSE+LTAE [141].

**Source-training** To initialize models on the labeled source domain, we follow the original training approach of PSE+LTAE [142]. We train for 100 epochs with the Adam [82] optimizer with an initial learning rate of 0.001 and we decay the learning rate using a cosine annealing schedule [99]. We use weight decay of 0.0001, batch size 128, and focal loss $\gamma = 1$. Inputs are normalized to $[0,1]$ by dividing by the max 16-bit pixel value $2^{16} - 1$. The best source-trained model is selected using the source validation set. We augment the inputs by randomly sub-sampling 30 time steps. The pixel-set size of PSE is set to $S = 64$ during training. The same setup is used for the target-trained model. For the final evaluation, we do not sample time steps or pixels, and instead input all available time steps ($T = T_i$) and pixels ($S = S_i$) for each example to the model. This ensures deterministic test results, and we also observe slightly improved results by doing so.

**ShiftAug** When training with ShiftAug, all training data (both source and target) are randomly shifted during training as described in Section 6.4. ShiftAug is disabled during evaluation.

| Method | ShiftAug | DK1→FR1 | DK1→FR2 | DK1→AT1 | FR1→DK1 | FR1→FR2 | Avg. |
|---|---|---|---|---|---|---|---|
| Source-trained | ✗ | 28.3±1.9 | 29.0±5.2 | 43.4±4.0 | 24.9±2.0 | 70.3±1.9 | 39.2±3.0 |
| | ✓ | 40.9±0.8 | 37.4±2.3 | 48.9±2.8 | 47.3±1.9 | 70.5±1.1 | 49.0±1.8 |
| FixMatch [157] | ✗ | 24.2±4.0 | 28.2±6.9 | 37.4±5.6 | 26.2±1.8 | 70.4±0.9 | 37.3±3.8 |
| | ✓ | 48.2±1.3 | 44.2±3.2 | 57.4±2.2 | 51.3±1.6 | 67.7±0.2 | 53.7±1.7 |
| MMD [170] | ✗ | 36.6±0.7 | 35.5±0.6 | 49.7±2.0 | 32.5±2.0 | 61.6±2.6 | 43.2±1.6 |
| | ✓ | 42.2±0.4 | 39.5±0.8 | 48.9±2.4 | 42.8±2.3 | 59.0±2.7 | 46.5±1.7 |
| DANN [37] | ✗ | 38.7±0.7 | 37.3±0.6 | 52.0±1.4 | 34.0±1.8 | 71.0±0.2 | 46.6±0.9 |
| | ✓ | 45.3±2.2 | 44.1±1.4 | 52.4±1.4 | 42.9±2.5 | 68.7±0.5 | 50.7±1.6 |
| CDAN+E [98] | ✗ | 39.3±0.6 | 37.9±0.3 | 51.5±2.9 | 36.5±1.3 | 71.7±0.6 | 47.4±1.1 |
| | ✓ | 46.5±2.3 | 45.2±1.3 | 55.0±1.3 | 46.9±0.5 | 70.7±1.3 | 52.9±1.3 |
| ALDA [16] | ✗ | 36.9±0.2 | 33.1±1.9 | 47.2±3.9 | 35.0±1.0 | 55.3±3.1 | 41.5±2.0 |
| | ✓ | 42.8±2.1 | 36.2±0.6 | 51.5±2.2 | 40.7±1.3 | 53.8±3.9 | 45.0±2.0 |
| JUMBOT [31] | ✗ | 36.8±0.2 | 33.6±1.3 | 50.5±0.6 | 35.6±3.0 | 63.7±3.0 | 44.0±1.6 |
| | ✓ | 42.7±0.1 | 38.3±1.2 | 49.7±4.2 | 41.5±0.5 | 62.2±1.2 | 46.9±1.4 |
| **TimeMatch** | ✗ | **57.4±1.5** | **47.0±0.9** | **61.7±4.9** | **52.1±1.4** | **73.0±0.5** | **58.2±1.8** |
| Target-trained | ✗ | 74.6±0.6 | 72.4±1.4 | 86.9±2.7 | 90.6±4.3 | 85.7±0.7 | 82.0±1.9 |

Table 6.1: Macro F1-score (%) results on our dataset for unsupervised cross-region adaptation. We consider five adaptation tasks across four Sentinel-2 tiles: DK1=32VNH (Denmark), FR1=30TXT (mid-west France), FR2=31TCJ (southern France), and AT1=33UVP (Austria).

**TimeMatch** We use the same training setup as the source-trained model but instead train for 20 epochs with a lower initial learning rate of 0.0001. We define an epoch as 500 iterations to fix the frequency in which the temporal shift is re-estimated. We use maximum temporal shift $\Delta = 60$ days, as we did not observe shifts greater than 2 months for our dataset in Europe. We set the trade-off hyperparameter $\lambda = 2.0$ in Equation 6.13, EMA keep-rate $\alpha = 0.9999$, and pseudo-label threshold $\tau = 0.9$. A sensitivity analysis of these hyperparameters is provided in Section 6.5. For the FixMatch [157] augmentations, we use the identity function for the weak $a(\cdot)$ in Equation 6.10 and randomly sub-sample time steps for the strong $A(\cdot)$ in Equations 6.9 and 6.12. These are used for simplicity, and we leave the use of more advanced augmentations for SITS to future work. At each iteration, we sample two mini-batches of size 128, one from the source and one from the target, in order to calculate the TimeMatch objective in Equation 6.13. We use a class-balanced mini-batch sampler for the source domain to ensure each source mini-batch contains roughly the same number of samples for each class. This reduces the class imbalance problem for the source domain for improved performance. Additionally, we apply domain-specific batch normalization [13, 93, 144] by forwarding the source and target mini-batches separately instead of concatenated. This ensures the batch normalization [68] statistics are calculated separately for each domain, for improved adaptation.

**Competing Methods** We re-implement the competitors MMD, DANN and CDAN+E following the domain adaptation library in [75], and use the original source codes for ALDA [16] and JUMBOT [31]. FixMatch [157] follows our re-implementation for TimeMatch with an EMA teacher and the student as the final model. All methods are initialized from a source-trained model. ShiftAug versions are initialized from the corresponding ShiftAug source-trained model, and we continue to use ShiftAug during training. As in TimeMatch, we train for 20 epochs and tune the hyper-parameters of these methods on the task DK1→FR1. The full details can be found in our source code.

## 6.5 Experimental Results

### Main Results

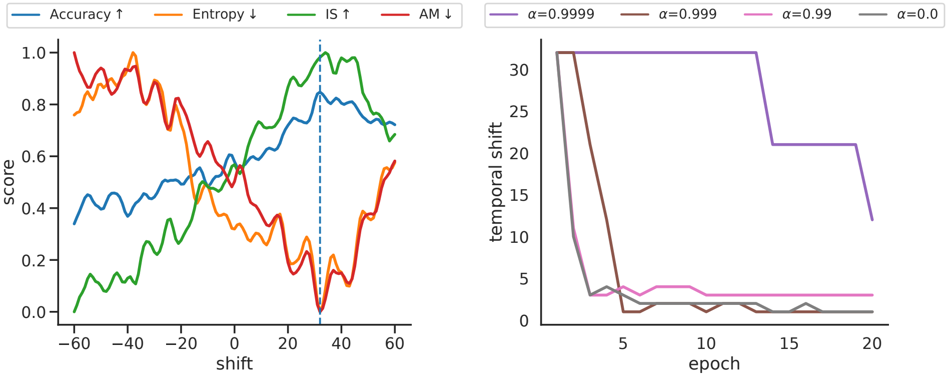
Table 6.1 shows the performance obtained with our approach and the re-implemented baselines and competitors. We report the mean and standard deviation of macro F1 scores, calculated from the results of three runs with different dataset splits.

We observe that source-trained models transfer very poorly to new target regions, with an average F1-score of 39% on target data. In comparison, target-trained models on the same classes achieve 82% on average. We observe that training shift-invariant models with ShiftAug improves domain generalization, leading to an increased average score of 49%. This greatly motivates addressing the temporal shift in UDA.

Existing UDA methods, however, only slightly increase the performance of source-trained models, with the best result obtained by CDAN+E [98] with 47%. By incorporating our ShiftAug, we observe a performance boost across all evaluated methods, indicating that existing methods are unable to implicitly handle the temporal shift.

Our approach TimeMatch, where we explicitly estimate the temporal shift, outperforms all competing methods by 11% on average and 5% for their ShiftAug variants. This shows that accounting for the temporal shift is a key component for the cross-region adaptation problem of SITS. Moreover, the shift-variant approach of TimeMatch outperforms the shift-invariance strategy. We hypothesize that training for shift-invariance may complicate crop classification, as the classification of certain crop types is shift-variant. For example, spring barley and winter barley develop similarly over time but shifted, as also discussed in Section 6.1.

Comparing TimeMatch to the results of the target-trained model, we observe that our approach—without any target labels—recovers a significant part of the highest achievable performance if target labels were available, but we also find that there is room for improvement. From our results, we see that methods which explicitly account for the temporal shift, such as TimeMatch and the ShiftAug variants of competing methods, generally outperform methods which do not. We therefore believe that further improvements can be gained by considering stronger forms of temporal alignment than shifts, such as class-wise alignments or time warping. We leave this interesting direction to future work.



(a) Score of temporal shifts.

(b) Temporal shift during training.

Figure 6.7: (a) Overall accuracy, entropy, IS, and AM scores of a source-trained model when applied to the target domain with different shifts. The dashed line indicate the most accurate shift. (b) The re-estimated temporal shifts of the teacher model during TimeMatch learning with different EMA decay rates.

Lastly, we highlight the results of the semi-supervised learning method FixMatch [157]. This method is similar to TimeMatch, but without temporal shift estimation. We observe that without ShiftAug, FixMatch obtains results worse than the source-trained model. This indicates semi-supervised learning cannot address the cross-region task alone. With ShiftAug, however, the results are greatly improved on average. Interestingly, the performance is worse without ShiftAug for all tasks *except* FR1→FR2. Here, the source and target regions are the most geographically close, and as result, the temporal shift is also closer to zero (see the top-left table in Figure 6.4). This indicates that ShiftAug (controlled by $\Delta$) is a trade-off between better long-range classification results and worse short-range results. In contrast, by estimating the temporal shift directly, TimeMatch does not have this issue and outperforms shift-invariance at both short and long distances.

### Analysis of Temporal Shift Estimation

In Figure 6.7a, we show the change in the overall accuracy of a source-trained model when applied to target data with different temporal shifts for DK1→FR1. We also show the change in entropy, IS, and AM scores of the model. We observe a significant increase in accuracy by temporally shifting the target data. Calculating the statistics of entropy, IS, and AM from the predictions of the model works well as an unlabeled proxy to accuracy. We aim to estimate the shift with the highest accuracy (dashed blue line) for the highest quality pseudo-labels. For the shown example, the minimum of both entropy and AM correspond to the best shift. However, we find AM to be the most consistent across different adaptation tasks.

In Figure 6.7b, we show the rate at which the estimated temporal shift for the teacher goes to zero in TimeMatch learning when training with different EMA decay rates. When the shift changes, the previous estimate becomes sub-optimal for generating accurate pseudo-labels. We address this by re-estimating the temporal shift during training. We observe that low decay rates (*e.g.* 0.99) require the shift to be re-estimated after a few iterations, which is inefficient. In comparison, a decay rate of 0.9999 allows us to only re-estimate the shift only once every epoch.

| Ablation | DK1→FR1 |
|---|---|
| No EMA ($\alpha = 0.0$) | 49.9±3.7 |
| No source temporal shift ($\delta^{s\to t} = 0$) | 51.9±1.9 |
| No balanced batch sampler for source | 53.3±3.6 |
| IS instead of AM | 56.3±2.6 |
| Entropy instead of AM | 56.9±1.8 |
| No domain-specific batch norm. | 56.9±4.1 |
| **TimeMatch** | 57.4±1.5 |

Table 6.2: Ablation study of TimeMatch components, sorted by increasing F1-score (%).

The table in the upper left corner of Figure 6.4 shows the initial temporal shifts estimated by our method. We find the estimated shifts are connected to the climatic differences between regions. For example, the temporal shift ($\delta^{t\to s}$) from the warmer FR1 (mid-west France) to the colder DK1 (Denmark) is estimated as 32 days. Due to the warmer climate, crops in FR1 mature earlier than in DK1, and a positive shift is required to align the former with the latter. In the other direction, the opposite is true, and indeed, we estimate a negative temporal shift of $-35$ days. Note that these are off by 3 days due to estimation variance. Here, the two source-trained models used to estimate the temporal shift in each direction are trained with two completely separate source region, yet their estimated shifts are still roughly inverses. This indicates that the temporal shift learned by these models is connected to the phenological properties of their respective source regions.

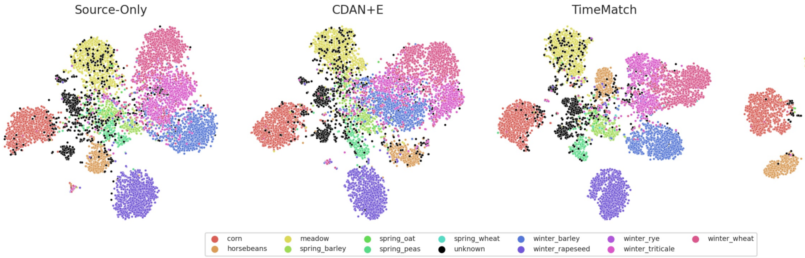
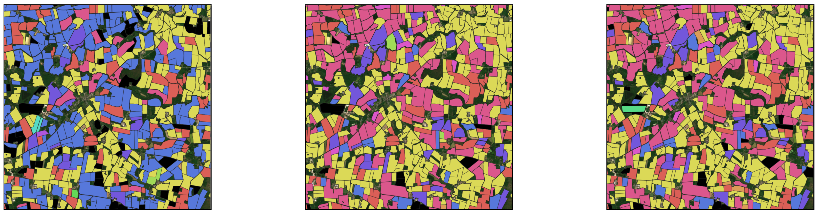


Figure 6.8: Visualization with t-SNE [172] of target features for the DK1→FR1 task. TimeMatch shows improved clustering of target features compared to existing approaches.



(a) Source-Only (b) TimeMatch (c) Ground Truth

Figure 6.9: Parcel predictions for an example target area (6 km$^2$) from the DK1→FR1 task, comparing (a) Source-Only, (b) TimeMatch, and (c) the corresponding ground truth. The figure shows the combination of multiple individual parcel predictions in the target region. The colors map to the classes in Figure 6.8.

## Ablation Study

To better understand how TimeMatch is able to obtain state-of-the-art results, we perform an ablation study on the different components for the task DK1→FR1. We report the results in Table 6.2. We first study the impact of the EMA training. Instead of EMA, we set the teacher as a direct copy of the student (No EMA). We observe that training without EMA introduces a significant drop in F1-score. This shows that EMA is important to ensure high pseudo-label accuracy. Setting $\delta^{s\to t} = 0$ disables the temporal shift of the source domain, and the student is trained with datasets with different temporal shifts. We observe a significant decrease in F1-score as a result. Disabling the balanced mini-batch sampler for the source domain also leads to a degradation of the performance. If the model is trained with class imbalanced source data, the teacher will make biased pseudo-labels for the samples from the target domain [58]. This hinders the TimeMatch learning process, as pseudo-labels for infrequent classes in the source domain are less likely to be generated for the target. By applying a balanced mini-batch sampler for the source, we address this problem by ensuring each source batch contains roughly the same number of samples for each category. Estimating the temporal shift with IS or entropy instead of AM results in a slight performance drop. Domain-specific batch normalization is simple to implement, as it just requires forwarding source and target batches separately instead of concatenated. Disabling this component results in a small average performance loss with notably higher variance.

### Sensitivity Analysis

Here we study the sensitivity of the TimeMatch hyperparameters. The results are shown in Figure 6.10. Higher values of $\alpha$ lead to better results, with a decay rate of 0.9999 being the best. However, increasing it to 1.0, so the teacher is not updated, results in a drop in F1-score, as the teacher cannot benefit from the knowledge learned by the student. The confidence threshold $\varepsilon$ controls the trade-off between the quality and quantity of pseudo-labels. A threshold of 0.9 gives the best F1-score and further increasing the threshold to 0.95 drops performance as a result of too few pseudo-labels, which particularly decreases performance for the less frequent classes. Finally, the trade-off parameter $\lambda$ controls the importance of the source domain loss $\mathscr{L}^{s\to t}$ with respect to the target domain loss $\mathscr{L}^{t}$. We observe that this hyperparameter is less important than the other two, but setting $\lambda = 2.0$ gives the best results.

### Visual Analysis

Finally, we visualize the ability of TimeMatch in learning discriminative features for the target domain. In Figure 6.8, we visualize t-SNE [172] embeddings of target domain features from source-trained, CDAN+E (the best competitor on average), TimeMatch, and target-trained models on the task DK1→FR1. The colors of the points represent their class (black is the unknown class). With TimeMatch, the target features are better clustered into their respective classes compared CDAN+E, which does not result in much better feature separation than the source-trained model. The target-trained plot shows the best possible learned features when training with all available target labels. Even with labels, the classes are not perfectly separated, *e.g.* for unknown/meadow or winter triticale/winter wheat.

Figure 6.9 shows example parcel predictions in a small area for the source-trained and TimeMatch models compared to the ground truth. The colors represent the same classes as before. We observe a large class confusion for the source-trained model, in particular between winter barley (blue) and winter wheat (dark pink), which are also not separated well in Figure 6.8. Without using any target labels, TimeMatch resolves this issue, resulting in clusters that better resemble the ground truth.

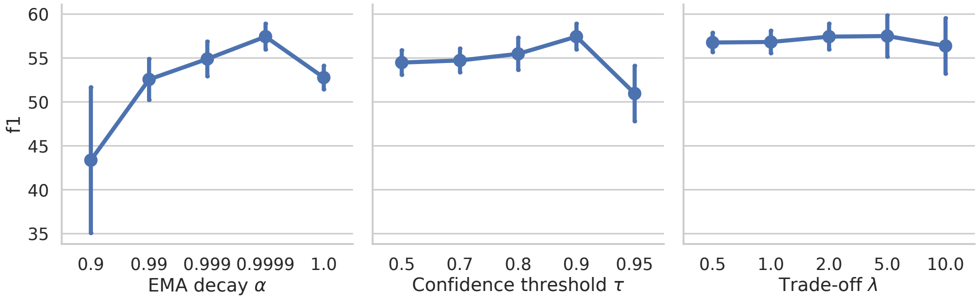


Figure 6.10: Sensitivity analysis of TimeMatch for the EMA decay rate, pseudo-label confidence threshold, and the trade-off in Eq. 6.13. The error bars show standard deviation.

## 6.6 Conclusion

This paper presented TimeMatch, a novel cross-region adaptation method for SITS. Unlike previous methods that solely match the feature distributions across domains, TimeMatch explicitly captures the underlying temporal discrepancy of the data by estimating the temporal shift between two regions. Through TimeMatch learning, we adapt a crop classifier trained in a labeled source region to an unlabeled target region. This is achieved by a learning algorithm that combines temporal shift estimation with self-training, where target pseudo-labels are generated using the estimated temporal shift from target to source. Lastly, we presented the TimeMatch dataset, a new large-scale cross-region UDA dataset with SITS from four different regions in Europe. Evaluated on this dataset, TimeMatch outperforms all existing approaches by 11% F1-score on average across five different adaptation tasks, setting a new state of the art in unsupervised cross-region adaptation. While this demonstrates that TimeMatch reaches strong results, there is still a gap with the performance obtained by fully supervised approaches. To overcome this limitation, we hypothesize that stronger temporal alignments, *e.g.* class-wise alignments or time warping, could further improve the performance. Another possibility is to perform domain adaptation across both time and space, which in addition to the temporal aspect also brings new considerations, such as the change in parcel shapes over time and crop rotations. We hope our proposed method and released dataset will encourage the remote sensing community to consider the challenging cross-region adaptation problem and its temporal aspect.

## 6.7 Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and constructive feedback. The work of Joachim Nyborg was funded by the *Innovation Fund Denmark* under reference *8053-00240*.

# Chapter 7

# Generalized Satellite Image Time Series Classification with Thermal Positional Encoding

JOACHIM NYBORG, AARHUS UNIVERSITY, DENMARK
CHARLOTTE PELLETIER, UNIVERSITÉ BRETAGNE SUD, FRANCE
IRA ASSENT, AARHUS UNIVERSITY, DENMARK

Submitted to *EarthVision 2022 Workshop* at the *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

### Abstract

Large-scale crop type classification is a task at the core of remote sensing efforts with applications of both economic and ecological importance. Current state-of-the-art deep learning methods are based on self-attention and use satellite image time series (SITS) to discriminate crop types based on their unique growth patterns. However, existing methods generalize poorly to regions not seen during training mainly due to not being robust to temporal shifts of the growing season caused by variations in climate. To this end, we propose Thermal Positional Encoding (TPE) for attention-based crop classifiers. Unlike previous positional encoding based on calendar time (*e.g.* day-of-year), TPE is based on thermal time, which is obtained by accumulating daily average temperatures over the growing season. Since crop growth is directly related to thermal time, but not calendar time, TPE addresses the temporal shifts between different regions to improve generalization. We propose multiple TPE strategies, including learnable methods, to further improve results compared to the common fixed positional encodings. We demonstrate our approach on a crop classification task across four different European regions, where we obtain state-of-the-art generalization results. Our source code is available at `https://github.com/jnyborg/tpe`.

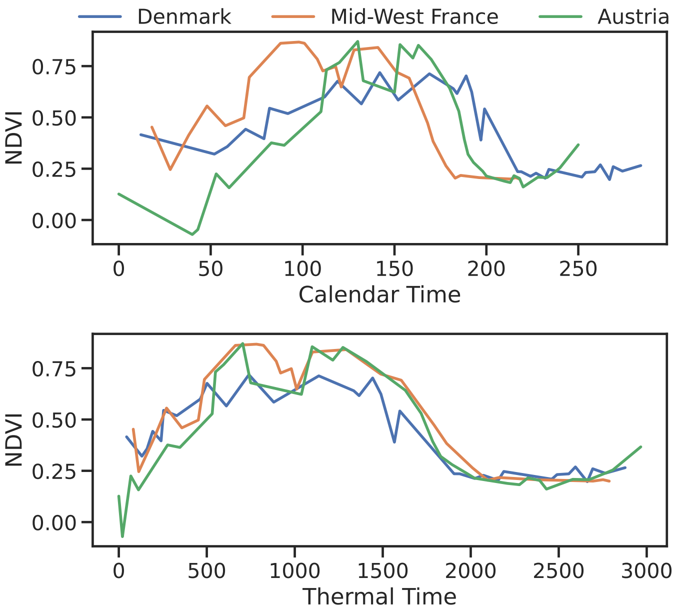


Figure 7.1: Winter wheat NDVI in different European regions with calendar time and thermal time. With thermal time, temporal shifts of crop growth in different regions are greatly reduced.

## 7.1 Introduction

The increase in openly accessible satellite image time series (SITS) has led to the development of deep learning models using remote sensing data that has significantly improved the state of the art in SITS classification tasks. Among these, crop type classification has numerous applications of economic and ecological importance, such as environmental monitoring, food security, and crop price prediction. Time series data is particularly valuable for crop classification, as it enables models to capture crop *phenology*, *i.e.* the progression of growth over time which characterizes different crop types. Specialized deep learning models for the task thus focus on the temporal aspect of the problem, proposing models based on neural network components that process time, such as temporal convolutions [125, 193], recurrent layers [65, 108, 112, 137], or most recently self-attention [136, 141, 142].

Since the growth patterns of crops are similar in different regions of the world [56], it is reasonable to expect that models trained in one region can generalize to another. However, recent works have found that existing models generalize poorly to other regions than those seen during training [101, 118]. Part of the challenge in generalization is the variability in climate which causes different timing of crop growth [35]. For example, in cooler regions, crops reach their growth stages later than in warmer regions, which models must account for to generalize [118].

To model the progression of time, the predominant approach in existing models is to use calendar time to include temporal context, either during pre-processing to interpolate the data into regular temporal sampling [67, 125, 136, 180] or as an explicit additional input [102, 138]. Notably, state-of-the-art methods based on self-attention input calendar time via *positional encoding* [41, 142]. Since self-attention is position agnostic [173], this provides explicit positional information about the temporal location of images within the growing season. This helps crop classification as the particular timing for the phenological events of a crop type can be an important clue in its classification, *e.g.* to distinguish spring wheat from winter wheat. However, the phenological calendar timing of one region is not generally shared with other regions due to temporal shifts, which causes existing models to generalize poorly [80, 118].

To overcome this challenge, we propose Thermal Positional Encoding (TPE) to improve the generalization of crop classifiers. Our core idea is to use a representation that captures the climatic variation affecting growth rates without relying on calendar time. To this end, we propose positional encoding based on *thermal time* [104, 105] for self-attention models. Thermal time is typically measured for crops by units of *Growing Degree Days* (GDD) [56, 104, 107, 189], computed by accumulating daily average temperatures above a baseline. As crop growth is directly related to the accumulation of heat over the growing season [35, 76], an earlier crop growth corresponds to an earlier increase in GDD and vice versa. This is illustrated in Figure 7.1 using normalized difference vegetation index (NDVI) to display winter wheat phenology in three different regions. Thermal time improves generalization of models by making SITS from different regions invariant to temporal shifts. At the same time, it provides a temporal location of images which allows thermal time to directly replace calendar time in crop classifiers.

To encode positional information, existing works generally use sinusoidal encoding [173]. However, as this approach is predefined and not learned, it lacks flexibility and may not capture crop-specific positional information. In this paper, we propose multiple TPE methods to encode thermal time in a data-driven way. By learning an encoding function instead of, *e.g.* an embedding vector for each position [27, 128], our approach is inductive. This allows us to handle when the thermal time of test regions differs from that of training, which is common in practice. We evaluate our approach on a crop classification task across four different European regions on the TimeMatch dataset [116], containing Sentinel-2 SITS expanded with daily temperature data, and demonstrate that we obtain state-of-the-art generalization results in new regions. Our main contributions are:

- We propose the use of thermal time in crop classification to increase robustness to temporal shifts and improve generalization.
- We propose TPE methods, which are based on thermal time and can easily be implemented in recent attention-based crop classifiers.
- We demonstrate that TPE greatly improves generalization across four different European regions.

## 7.2 Related Work

**Satellite Image Time Series Classification.** Multiple traditional machine learning approaches, such as random forests or support vector machines, have been applied to crop classification [66, 174, 176, 181]. These approaches require input features to be extracted by hand. For instance, a widely used feature is NDVI, combining the red and near-infrared spectral bands, which relates to the photosynthesis of crops [167]. Other works also include phenological features [74, 171] or meteorological information [192]. Although these handcrafted features are robust and interpretable, deep learning approaches are mostly employed as they enable the automatic extraction of richer features from raw SITS. Deep convolutional networks have been widely applied to process the spatial dimensions of the data [87, 138], while the temporal dimension has been processed by recurrent units [112, 137], 1D convolutions [125, 193], or combinations thereof [67, 138]. Recently, self-attention [173] has led to significant improvements in pixel [136] and parcel classification [141, 142], as well as semantic and panoptic segmentation [41]. Since self-attention is position-agnostic, existing works use sinusoidal positional encoding [173] of calendar time to capture the position of images in the growing season. We propose positional encoding based on thermal time [104, 105] to improve the generalization of the promising self-attention mechanism.

**Domain Generalization for SITS.** Several prior works have reported that existing crop classification models fail to generalize across space and time due to not being robust to temporal shifts of the growing season [80, 100, 118, 180]. This problem has mainly been tackled by unsupervised domain adaptation (UDA), where models are trained with labeled data from a source region and unlabeled data from a target region [168]. Phenology Alignment Network [180] addresses this problem by learning domain-invariant features obtained with a maximum mean discrepancy loss [170] for the unlabeled target data. TimeMatch [118] obtains further improvements by directly estimating the temporal shift of the target region, and utilizing the shift estimation to train with pseudo-labels for the unlabeled target region. Our setting differs from UDA, as we do not aim to adapt models to particular regions by training with unlabeled data, but to improve the generalization of a crop classifier model trained with labeled data from multiple areas to any new region.

Most similar to our work, Kerner *et al.* [80] improve the generalization of crop classifiers by inputting satellite data at specific time steps which correspond to particular growth stages (greenup, peak, and senescence), computed from the NDVI sequence for each input. By dynamically selecting these time steps, this approach can account for temporal shifts of the growing season, but information is lost since the complete time series is not involved in the prediction. In comparison, we aim to train self-attention models which attend to the most relevant time steps in the complete time series automatically by incorporating thermal time.

**Positional Encodings.** A vast literature exists in positional encoding for the self-attention mechanism. Absolute positional encoding is most widely used. In the original Transformers [173], vectors are encoded from the absolute position in the sequence by sinusoidal functions, but this approach is less flexible as the vectors are fixed and not learned. To overcome this issue, a common approach is to learn an embedding vector for each position [27, 128] similar to word embeddings, but this approach requires all possible positions to be seen during training to ensure all the embeddings are updated by gradient descent. This is ill-suited for irregularly sampled SITS, which does not guarantee that all possible (calendar or thermal) positions are available for training. Instead, approaches that learn a function that maps positions to vectors [92, 96, 113] do not have this requirement and can thus generalize to unseen positions at test time. We therefore build upon these in this paper.

Another line of work is relative positional encoding [23, 63, 151], which encodes the positional difference between each pair in the input sequence instead of the absolute position of individual elements. While relative positions can be more relevant than absolute in other tasks, in SITS classification, the absolute position is crucial information. For example, a satellite image taken during the winter will not contain the same information about crop growth compared to an image from the spring, which cannot be captured by only the relative positions, *e.g.* the difference in days between the two images. Thus, we focus on absolute positional encoding in this work.

## 7.3 Self-Attention for Crop Classification

In crop classification, we are given a satellite image time series $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}]$, where $T$ is the length of the time series. The goal of the classification task is to associate $\boldsymbol{x}$ with one of $K$ classes. In our setting, each $\boldsymbol{x}^{(t)} \in \mathbb{R}^{T \times N \times C}$ consists of a sequence of $N$ pixels of $C$ spectral bands within a *parcel*, *i.e.*, a homogeneous agricultural plot of land. This approach requires parcel shapes to be available in the region for classification, which is widely available in the European Union (EU) [147] or can alternatively be acquired by a segmentation step [41, 159].

Our goal is to improve the generalization of existing crop classifiers by accounting for temporal shifts of the growing season. Owing to its state-of-the-art performance, we build upon the PSE+LTAE model [141]. The network consists of the Pixel-Set Encoder (PSE) and the Lightweight Temporal Attention Encoder (LTAE). Given a randomly sampled pixel-set of size $S$ among the $N$ available pixels of an input $\boldsymbol{x}$, the PSE handles the spatial and spectral context of SITS by processing each time step individually to a sequence of embedding vectors $\boldsymbol{e} = [\boldsymbol{e}^{(1)}, \ldots, \boldsymbol{e}^{(T)}] \in \mathbb{R}^{T \times D}$, where $D$ is the embedding dimension. PSE does not process the temporal dimension. We thus focus on handling temporal shifts in the LTAE module. Given $\boldsymbol{e}$, LTAE extracts temporal features using a simplified version of the multi-headed self-attention, as we describe next.

**Self-Attention.** In the original Transformer model [173], self-attention is computed with a query-key-value triplet $(\boldsymbol{q}^{(t)}, \boldsymbol{k}^{(t)}, \boldsymbol{v}^{(t)})$ for each element in the input sequence using three fully-connected layers. The output is a sequence where each element is a sum of all values $\boldsymbol{v}^{(t)}$ weighted by their attention score. The attention scores for a time step $t$ are computed as the similarity (dot product) between all keys and the query $\boldsymbol{q}^{(t)}$, re-scaled by a softmax layer. The computation of the query-key-value triplets can be performed in parallel, which enables the Transformer model to take full advantage of GPUs for a significant speed increase compared to the sequential computation of recurrent neural networks (RNN). In multi-headed self-attention, the triplets are computed multiple times in parallel with different parameters, or "heads", which further increase efficiency and also the representational capacity as each head can specialize in different parts of the sequence.

**Sinusoidal Positional Encoding.** As the self-attention mechanism is position-agnostic [173], various positional encodings (PE) have been introduced to capture positional information. This is typically done by mapping scalar positions to a vector, either by learning or by heuristics, and adding each embedding vector with their positional encoding $\boldsymbol{e}^{(t)} + \boldsymbol{p}^{(t)}$ before applying self-attention. The original Transformer model [173] uses a fixed sinusoidal encoding with predefined wavelengths, defined as:

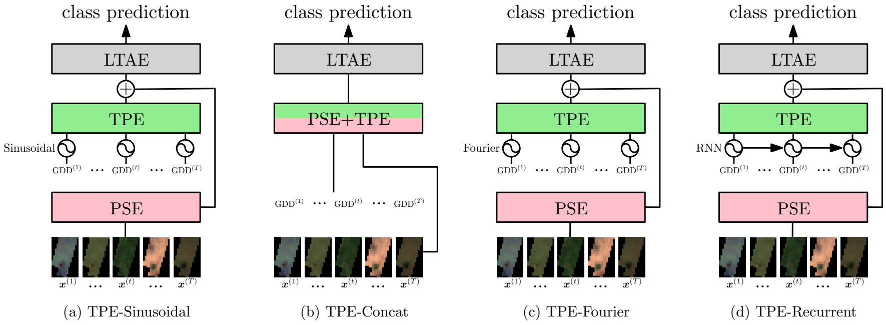
$$\boldsymbol{p}^{(t)} = [\sin(\omega_i t), \cos(\omega_i t)]_{i=1}^{D/2} \tag{7.1}$$

where $\omega_i = (1/\tau)^{2i/D}$ and $\tau = 10000$.

**Lightweight Temporal Attention Encoder.** While the original self-attention maps the input embeddings $\boldsymbol{e}$ to an output sequence of embeddings, the goal of SITS classification is to map the entire time series into a single embedding. To address this, the LTAE module [141] modifies the self-attention mechanisim by replacing the queries $\boldsymbol{q}^{(t)}$ with a single learnable "master" query $\hat{\boldsymbol{q}}$, resulting in a single output embedding instead of a sequence. The computation is also made more lightweight by employing a channel grouping strategy [186], where each attention head operates on its own subset of input channels. The LTAE module uses the sinusoidal PE (Equation 7.1 with $\tau = 1000$) but encodes the day of the year $\text{day}^{(t)}$ instead of the position index $t$. This enables the model to account for the inconsistent temporal sampling of SITS, but also introduces problems with handling temporal shift [118].

## 7.4 Method

In this work, we observe that the positional encoding used by the LTAE module has two issues. First, since it encodes calendar time, it introduces the temporal shift problem as displayed in Figure 7.1. While calendar time is useful to identify the crop types in a particular region, it hinders generalization to new regions [80]. For example, while spring and winter crops can be similar in appearance, they are easily separated by the timing of their growth stages as spring crops are planted later in a


class prediction
LTAE
TPE
PSE
PSE+TPE
Sinusoidal
Fourier
RNN
GDD$^{(1)}$ ··· GDD$^{(t)}$ ··· GDD$^{(T)}$
$\boldsymbol{x}^{(1)}$ ··· $\boldsymbol{x}^{(t)}$ ··· $\boldsymbol{x}^{(T)}$

(a) TPE-Sinusoidal (b) TPE-Concat (c) TPE-Fourier (d) TPE-Recurrent

Figure 7.2: Schematic illustration of our Thermal Positional Encoding (TPE) methods with the PSE+LTAE model [141].

growing season than winter crops. However, because of temporal shifts, the same time positions of spring crops could represent winter crops in another region. Without any way of accounting for temporal shifts, calendar time positional encodings are unlikely to generalize. Second, since the positional encoding is fixed and not learned, it prevents the model from taking advantage of end-to-end training the encoding function to further improve generalization [92, 96, 128].

A possible remedy to the first issue is to augment the training data with random temporal shifts, such as ShiftAug, a SITS augmentation technique proposed in [118], so that the model does not learn to associate a specific position with the phenological events seen in the training data. While this solution increases the invariance of the model to temporal shifts, the temporal shift is in some cases an important clue to distinguish crop types—such as the spring and winter crops. Instead, we want models that are shift-invariant between different regions, but shift-variant within the same region. That is, we want models which can use class-wise temporal shifts for classification but are unaffected by temporal shifts of the growing season.

To address the second issue, a common alternative to the fixed sinusoidal encodings is to treat each position as a discrete token that can be uniquely represented as a learnable vector [27, 43, 128]. While this approach enables the model to learn the positional encoding from data, it fails to generalize to positions not encountered during training. This is an issue for high-resolution SITS, as we typically do not have an observation for every possible position. For example, the Sentinel-2 satellites acquire images every five days. Moreover, images with high cloud coverage are often filtered, further reducing the positions available. In comparison, sinusoidal positional encoding is more practical for SITS, as an encoding vector is well-defined for every position independent of the training data.

## Thermal Positional Encodings

We argue that successful positional encoding for SITS should meet the following requirements:

(1) Making SITS from *different* regions shift-*invariant* to address the temporal shift problem.

(2) Making SITS from the *same* region shift-*variant* by providing absolute information of where an observation is located in the growing season.

(3) Must be inductive to be able to handle positions not seen during training.

(4) Being data-specific and thus learnable.

While the LTAE sinusoidal positional encoding based on calendar time meets the second and third requirements, it is not invariant to temporal shifts between different regions or trainable which violates the first and fourth requirements. To address this, we replace calendar time with thermal time to meet both the first and second requirements and propose four TPE strategies, including learnable methods to meet the third and fourth requirements.

**Thermal time.** When studying crop phenology, thermal time is a good proxy for the rate of crop growth [35, 104, 166]. Thermal time is typically measured in units of *growing degree days* (GDD). The GDD measured at a time $t$ is computed by accumulating daily average temperatures above a baseline:

$$\text{GDD}^{(t)} = \sum_{i=1}^{t} \max\left(\frac{T_{min}^{(i)} + T_{max}^{(i)}}{2} - T_{base}, 0\right) \tag{7.2}$$

where $T_{min}^{(i)}, T_{max}^{(i)}$ is the minimum and maximum temperatures for day $i$, accumulated for all the previous days $i = 1, 2, \ldots, t$. Temperature values are often clipped to a range $[T_{base}, T_{cap}]$ chosen depending on the crop type. Since we do not know the crop type of the input beforehand, we choose standard values $T_{base} = 0$ and $T_{cap} = 30$ [104, 107] for all crops, since growth typically stagnates below 0°C and does not grow any faster above 30°C. We accumulate from the starting day of the input SITS, in our case January 1. Since GDD is computed by a cumulative sum, it is a monotonically increasing function and thus preserves the order of the input time series. This enables GDD to directly replace day of year for the time positions in the self-attention computation. By replacing calendar time with thermal time, we can reduce the temporal shift of SITS between different regions while retaining the shift between classes within the same region and thereby satisfy the first and second requirements.

**TPE Methods.** We propose the following TPE methods to input thermal time to PSE+LTAE [141].

- TPE-Sinusoidal: We replace calendar time with thermal time in the sinusoidal PE, but the encoding is not learned.
- TPE-Concat: We learn SITS and positional input embeddings jointly by concatenating thermal time to an intermediate feature of the PSE module.

- TPE-Fourier: We learn the sinusoidal PE function by the method proposed in [92].
- TPE-Recurrent: We learn a positional encoding function that captures the development in GDD by a recurrent neural network (RNN).

An overview of the TPE methods is shown in Figure 7.2.

### TPE-Sinusoidal

To use GDD with the sinusoidal PE, we follow Equation 7.1 but replace $t$ with $\text{GDD}^{(t)}$. The benefit of using the sinusoidal positional encoding for GDD is that an encoding vector is well-defined for every possible GDD value. This ensures that even if we train with only a subset of possible accumulated temperatures, a positional encoding exists for unseen positions at test time. However, as the sinusoidal PE is fixed and not learned, it prevents the model from capturing data-specific positional information for the crop classification task.

### TPE-Concat

While the original Transformer network [173] takes pre-trained word embeddings as inputs, in our case, the embeddings are learned by the PSE module, which is learned simultaneously to the LTAE module. Thus, we propose an alternative to positional encoding where the encoding for the SITS and positions are learned jointly by the PSE. In particular, for each time step $t$, we concatenate $\text{GDD}^{(t)}$ to the intermediate PSE embedding $\hat{\boldsymbol{e}}^{(t)}$ before the final PSE output layer $\text{MLP}_2$:

$$\boldsymbol{e}^{(t)} = \text{MLP}_2([\hat{\boldsymbol{e}}^{(t)} \mid\mid \text{GDD}^{(t)}]), \tag{7.3}$$

where $[\cdot \mid\mid \cdot]$ indicates concatenation. The PSE output layer $\text{MLP}_2$ [142] is a multilayer perceptron (MLP) consisting of a linear layer, batch normalization [68], and ReLU [111] activation function. We note that this approach is similar to the method of inputting extra parcel geometric features in the original PSE. By concatenating positions to the embedding function, TPE-Concat removes the need for complex positional encoding functions, which may be more beneficial for SITS.

### TPE-Fourier

Li *et al*. [92] propose a learnable PE based on Fourier features [129], which can also be viewed as a generalization of the sinusoidal PE. For a position $t \in \mathbb{R}$, the Fourier PE is computed by:

$$\boldsymbol{r}^{(t)} = \frac{1}{\sqrt{D}}[\cos(\boldsymbol{W}_r t) \mid\mid \sin(\boldsymbol{W}_r t)], \tag{7.4}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{D/2}$ is a trainable vector. To give the representation additional capacity, the encoding is passed through an MLP:

$$\boldsymbol{p}^{(t)} = \text{MLP}(\boldsymbol{r}^{(t)})\boldsymbol{W}_p \tag{7.5}$$

where MLP consists of a linear layer with GeLU [60] activation function, and $\boldsymbol{W}_p$ are parameters for projecting the representation to the dimension of the input embeddings. The TPE-Fourier reveals whether it is more beneficial to learn the sinusoidal PE compared to the fixed TPE-Sinusoidal.

### TPE-Recurrent

Compared to natural language processing (NLP), where positions typically increase linearly with the sequence length, GDD increases non-linearly over the growing season (see Figure 7.4), as a result of the higher daily temperatures during the summer than the winter. It may therefore be beneficial not to only encode independent GDD values, but also incorporate previous values to account for different rates of crop growth over the year. To handle this, we propose to use an RNN to learn the positional encoding. RNNs have been successfully used for positional encoding in NLP tasks [96, 113]. We follow the RNN approach of Liu *et al.* [96]. In particular, we use a GRU [18], which computes its output $\boldsymbol{h}^{(t)} \in \mathbb{R}^{H_{out}}$ for each time step $t$ given an input $\boldsymbol{z}^{(t)} \in \mathbb{R}^{H_{in}}$ and the previous hidden state $\boldsymbol{h}^{(t-1)}$ by:

$$\boldsymbol{h}^{(t)} = \mathrm{GRU}(\boldsymbol{z}^{(t)}, \boldsymbol{h}^{(t-1)}). \tag{7.6}$$

Then, we obtain a positional encoding with target dimension $D$ by a linear projection:

$$\boldsymbol{p}^{(t)} = \boldsymbol{W}_p^{\top} \boldsymbol{h}^{(t)} + \boldsymbol{b}_p, \tag{7.7}$$

where $\boldsymbol{W}_p \in \mathbb{R}^{H_{out} \times D}$ and $\boldsymbol{b}_p \in \mathbb{R}^{D}$. Instead of scalar values $\mathrm{GDD}^{(t)}$, we use vectorized positions as the inputs $\boldsymbol{z}^{(t)}$, which are obtained by obtained by the sinusoidal positional encoding of $\mathrm{GDD}^{(t)}$ (Equation 7.1) as done in [96]. TPE-Recurrent learns a positional encoding that captures the temporal development in GDD, but is more computationally expensive due to the sequential computation of an RNN.

## 7.5 Experiments

### Setup

**Dataset.** We evaluate our approach on the TimeMatch dataset [116] with Sentinel-2 L1C SITS from four different tiles: 33UVP (Austria), 32VNH (Denmark), 30TXT (mid-west France), and 31TCJ (southern France). We refer to these regions by AT1, DK1, FR1, and FR2, respectively. We display the locations of these tiles in Figure 7.3. The dataset contains all available observations of these tiles between January 1, 2017, and December 31, 2017, with cloud cover $\leq 80\%$ and coverage $\geq 50\%$. The atmospheric bands (1, 9, and 10) are left out, keeping the remaining 10 spectral bands. The 20m bands are bilinearly interpolated to 10m.

The dataset is prepared for parcel classification by cutting the pixels of each parcel from the SITS using geo-referenced parcel shapes available from the Land Parcel Identification System (LPIS) in each country. The total amount of parcels is 280K

| | AT1 | | DK1 | | FR1 | | FR2 | | Avg. | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | F1 | OA | F1 | OA | F1 | OA | F1 | OA | F1 | OA |
| PSE+LTAE [141] | 68.3 | 90.5 | 55.4 | 62.6 | 74.6 | 90.9 | 73.5 | 87.5 | 68.0 | 82.9 |
| + w/o PE | 84.1 | 94.4 | 66.3 | 76.2 | 79.3 | 91.9 | 74.0 | 86.4 | 75.9 | 87.2 |
| + w/ ShiftAug [118] | 84.2 | 94.1 | 71.6 | 78.5 | 83.9 | 93.3 | 79.8 | 89.4 | 79.9 | 88.8 |
| + TPE-Sinusoidal | 85.6 | 94.7 | 78.7 | 84.8 | 83.0 | 92.6 | 81.1 | **90.4** | 82.1 | 90.6 |
| + TPE-Concat | 85.7 | 94.7 | 78.6 | 83.1 | 85.1 | 93.3 | **81.4** | 89.6 | 82.7 | 90.2 |
| + TPE-Fourier | 84.7 | 94.4 | 79.0 | **86.0** | 77.3 | 91.5 | 80.0 | 89.4 | 80.3 | 90.3 |
| + TPE-Recurrent | **86.5** | **95.0** | **80.3** | 85.4 | **86.0** | **93.8** | 80.5 | 89.8 | **83.3** | **91.0** |
| Upper-bound | 94.6 | 97.5 | 92.0 | 94.0 | 93.1 | 96.4 | 87.4 | 93.9 | 91.8 | 95.4 |

Table 7.1: Leave-one-region-out spatial generalization results in macro F1 score (F1) and overall accuracy (OA) (both in %). Each column shows the classification results in a new region after training on the others.

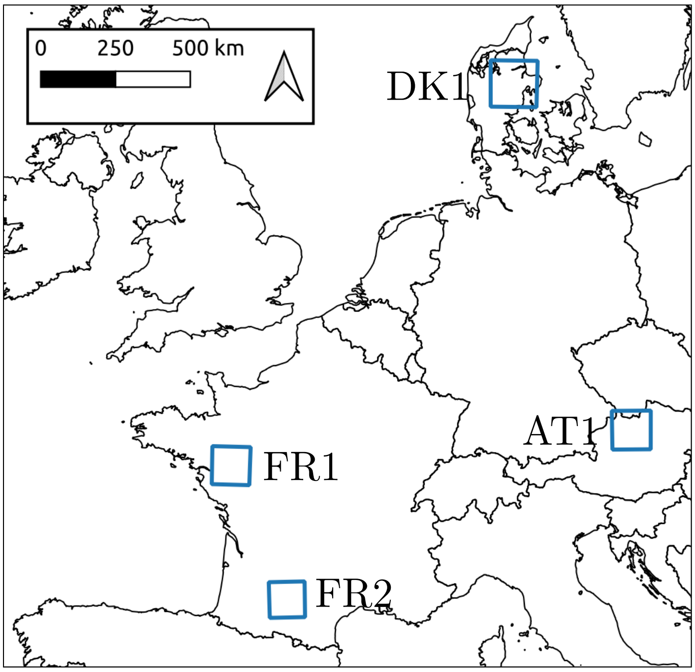


Figure 7.3: The geographical locations in Europe of the four Sentinel-2 tiles in the dataset [116]. Figure adapted from [118].

with 15 crop classes. The frequency of these classes varies greatly between tiles, for example, sunflowers are only frequent in the two France tiles. To ensure all tiles have enough samples of each class to learn their classification, we select the 9 crop types with at least 200 samples in all tiles: corn, horsebeans, meadow, spring barley, winter barley, winter rapeseed, winter triticale, winter wheat, and unknown. Here, the unknown class contains all parcels with crop type not of the other 8 classes. Each tile has its own train/validation/test sets, created by assigning all parcels in a tile at random to these sets by a 70%/10%/20% ratio.

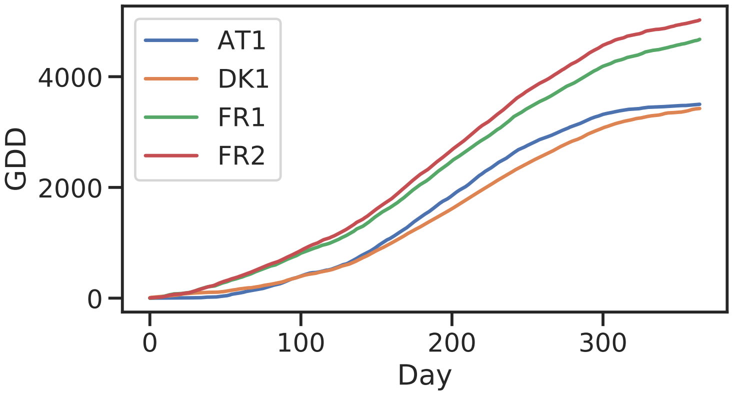


Figure 7.4: The development of GDD on average in the different Sentinel-2 tiles from January 1 to December 31, 2017.

We expand the TimeMatch dataset with weather information from the Europe-wide E-OBS dataset [22]. We use the daily minimum and maximum temperature from the 0.1° regular grid of 2017 to compute GDD for each parcel, geo-referenced by the parcel centroid. Figure 7.4 displays the average GDD computed for the four regions, showing the southern France tile FR2 is the warmest and the Danish tile DK1 the coldest.

**Implementation details.** We follow the original implementation of PSE+LTAE [141]. All models are trained for 100 epochs with a batch size of 128 on a single GTX 1080Ti GPU with Adam optimizer [82]. The learning rate is initialized to $1e{-3}$ and decayed each epoch by cosine annealing [99]. We use weight decay of $1e{-4}$. The 16-bit input pixels are normalized to $[0,1]$ by dividing by $2^{16}-1$. Our code is available at `https://github.com/jnyborg/tpe`.

**Experimental setup.** To evaluate whether our proposed thermal positional encoding improves generalization to new regions, we adopt a leave-one-region-out setup where we hold one Sentinel-2 tile out for testing and train on the remaining. In contrast to the domain adaptation setup of TimeMatch [118], where data is only available from one tile for training, our setup contains multiple different regions for training. In practice, we typically have many tiles available for training [147], so this setup allows us to evaluate against the naive approach of improving generalization by adding more training data.

**Model variants.** In comparison to TPE, we consider the following model variants:

- *PSE+LTAE* [141]. This is the baseline model which encodes calendar time (day of the year) with the sinusoidal positional encoding [173].

- *w/o PE*. This is PSE+LTAE where self-attention is computed without any positional information.
- *w/ ShiftAug* [118]. PSE+LTAE trained with calendar time augmented with random temporal shifts.
- *Upper-bound*. We train the best performing TPE method (TPE-Recurrent) with all four available regions to obtain the results of a fully-supervised upper bound.

### Parcel Classification Results

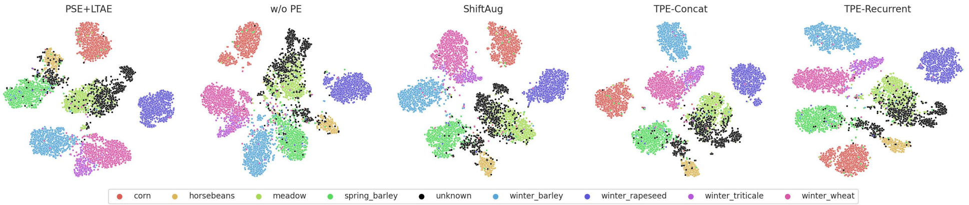


Figure 7.5: LTAE features of different methods embedded with t-SNE [172] for DK1 after training with the remaining regions.

In Table 7.1, we detail the performance obtained for the leave-one-region-out spatial generalization experiments. We report the class-averaged F1 score (F1) and the overall accuracy (OA). Compared to calendar time models (top), all our TPE models (bottom) have much better generalization results with the use of thermal time. TPE-Recurrent shows the best performance by being learnable and capturing the temporal development in GDD, increasing F1 on average by +15.3% over the default PSE+LTAE [141] model and +3.4% over the ShiftAug [118] augmented model. Our TPE greatly improves generalization, but there is still a gap to the upper-bound performance. TPE addresses the temporal shifts between regions but does not account for changes in the spectral signature of crops, which can be caused by differences in *e.g*. the topography, soil, or varieties of the cultivated crop type. We leave this direction to future work.

**Analysis of results.** We observe that the default PSE+LTAE with calendar time generalizes worst, obtaining an F1 score of 68.0% on average. Interestingly, simply removing the positional encoding outperforms the baseline significantly, leading to an average performance increase of +7.9%. Since this model variant is given no information about the order of images in the SITS, it is also invariant to temporal shifts, which explains the performance increase. However, without positional information, the model should not be able to model the class-wise timing differences, which should degrade performance. But the performance increase indicates the model is able to do so. We argue that this is because the model is able to extract some positional information from the SITS. For example, satellite images taken during the winter differ from those during the summer, enabling the model to extract some degree of

| Method | Training time (s/epoch) |
|---|---|
| TPE-Sinusoidal | 16.1 |
| TPE-Concat | **15.5** |
| TPE-Fourier | 16.4 |
| TPE-Recurrent | 17.2 |

Table 7.2: The training time of TPE in seconds per training epoch.

temporal order. However, in the case that two images at different times appear similar, the extracted positions can be ambiguous, which is avoided by providing explicit positional information. This is also indicated by the result of ShiftAug [118], where calendar time is augmented with random temporal shifts, which further increases the F1 results by $+11.9\%$ on average over the baseline, outperforming no positional encoding by $+4.0\%$. This indicates that direct positional information is indeed important to the crop classification task to avoid ambiguous order information from images only.

In comparison, our TPE models outperform all calendar time models. This highlights the benefits of using thermal time for reducing the temporal shift between different regions without introducing any augmentations, while also providing explicit positional information for modelling the class-wise timing differences. The TPE-Sinusoidal model is the default PSE+LTAE model but where calendar time positions are replaced with thermal time. This simple change significantly improves the F1 generalization results by $+14.1\%$ on average. Learning a sinusoidal PE with TPE-Fourier, however, is not beneficial, resulting in a decrease in F1 compared to TPE-Sinusoidal by $-1.8\%$. TPE-Concat learns embedding and positional representations jointly in the PSE module, and obtains comparable results to TPE-Sinusoidal, with higher F1 ($+0.6\%$) but lower OA ($-0.4\%$). But as TPE-Sinusoidal introduces extra computation because of the sinusoidal encoding function, TPE-Concat is computationally more efficient as shown in Table 7.2. This indicates that the approach of adding positional encodings to input embeddings common in natural language processing may be unnecessary for SITS classification. TPE-Recurrent learns a positional encoding that captures the development in GDD, leading to an increase in F1 of $+1.2\%$ over TPE-Sinusoidal. TPE-Recurrent thus shows the best performance but also introduces sequential computation which increases computation requirements as shown in Table 7.2. We suggest the choice of TPE method is a trade-off between performance and efficiency. Practitioners can easily implement TPE-Concat by concatenating thermal time in PSE [141], and enjoy improved generalization and efficiency. If more computation can be afforded, TPE-Recurrent offers the best results.

### Visual Analysis

To better understand how TPE obtains improvements, we visualize in Figure 7.5 t-SNE [172] embeddings of features output by the LTAE. For TPE methods, we observe denser and better separated clusters, indicating better class separation by accounting for temporal shifts. For the baseline PSE+LTAE model [141], we observe some classes are well clustered despite the temporal shift, such as corn and winter rapeseed, indicating these classes are less impacted by temporal shifts. Others are mixed, such as spring barley/horsebeans and winter wheat/winter triticale. We observe that temoving the PE results in less dense clusters. Particularly, the clusters for spring barley and winter barley overlaps. This could indicate difficulties in resolving class-wise temporal shifts, since these are better separated with ShiftAug [118].

## 7.6 Conclusion

In this work, we propose Thermal Positional Encodings (TPE) to address the temporal shift issue of SITS classifiers and improve generalization. While existing work uses calendar time, our TPE uses thermal time, which enables models to account for the varying rates of crop growth in different climates and thereby address the temporal shift issue. We propose different methods to positional encode thermal time, including fixed and learned approaches. On a parcel classification dataset with SITS from four different European regions, we demonstrate that TPE significantly improves generalization compared to existing methods.

**Acknowledgements.** Joachim Nyborg is funded by the *Innovation Fund Denmark* under reference *8053-00240*. We acknowledge the E-OBS dataset from the EU-FP6 project UERRA (`https://www.uerra.eu`) and the Copernicus Climate Change Service, and the data providers in the ECA&D project (`https://www.ecad.eu`).

# Bibliography

[1] Sajjad Ahmad, Ajay Kalra, and Haroon Stephen. Estimating soil moisture using remote sensing data: A machine learning approach. *Advances in water resources*, 33(1):69–80, 2010. 11

[2] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2019. 26, 40

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.0473`. 12

[4] Adeline Bailly, Laetitia Chapel, Romain Tavenard, and Gustau Camps-Valls. Nonlinear time-series adaptation for land cover classification. *IEEE Geoscience and Remote Sensing Letters*, 14(6):896–900, 2017. 55

[5] Shane Barratt and Rishi Sharma. A note on the inception score. In *Workshop on Theoretical Foundations and Applications of Deep Generative Models, ICML*, 2018. 61

[6] Inbal Becker-Reshef, Eric Vermote, Mark Lindeman, and Christopher Justice. A generalized regression-based model for forecasting winter wheat yields in kansas and ukraine using modis data. *Remote sensing of environment*, 114(6): 1312–1323, 2010. 15

[7] Mariana Belgiu and Ovidiu Csillik. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote Sensing of Environment*, 204:509–523, 2018. 30

[8] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010. doi: 10.1007/s10994-009-5152-4. 28, 53, 54

[9] Joseph K Berry, JA Detgado, Rajiv Khosla, and FJ Pierce. Precision conservation for environmental sustainability. *Journal of Soil and Water Conservation*, 58(6):332–339, 2003. 3

[10] Claire Boryan, Zhengwei Yang, Rick Mueller, and Mike Craig. Monitoring us agriculture: the us department of agriculture, national agricultural statistics service, cropland data layer program. *Geocarto International*, 26(5):341–358, 2011. 15

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3, 13, 18

[12] Lyndon Chan, Mahdi S Hosseini, and Konstantinos N Plataniotis. A comprehensive analysis of weakly-supervised semantic segmentation in different image domains. *International Journal of Computer Vision*, pages 1–24, 2020. 38, 40

[13] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019. 71

[14] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 55

[15] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *Winter Conference on Applications of Computer Vision*, pages 839–847. IEEE, 2018. 40, 46, 47

[16] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3521–3528, 2020. 53, 56, 66, 69, 70, 71

[17] Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *Nips*, volume 24, pages 2456–2464. Citeseer, 2011. 56

[18] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1724–1734. ACL, 2014. doi: 10.3115/v1/d14-1179. URL https://doi.org/10.3115/v1/d14-1179. 12, 86

[19] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 42

[20] Casey Chu, Andrey Zhmoginov, and Mark Sandler. CycleGAN, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017. 27

[21] European Commission. The common agricultural policy at a glance. `https://ec.europa.eu/info/food-farming-fisheries/key-policies/common-agricultural-policy/cap-glance_en`, 2022. Accessed: 2022-02-04. 5

[22] Richard C Cornes, Gerard van der Schrier, Else JM van den Besselaar, and Philip D Jones. An ensemble version of the E-OBS temperature and precipitation data sets. *Journal of Geophysical Research: Atmospheres*, 123(17): 9391–9409, 2018. doi: 10.1029/2017JD028200. 88

[23] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1285. URL `https://doi.org/10.18653/v1/p19-1285`. 81

[24] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 447–463, 2018. 55

[25] Jorge A Delgado, Nicholas M Short Jr, Daniel P Roberts, and Bruce Vandenberg. Big data analysis for sustainable agriculture on a geospatial cloud framework. *Frontiers in Sustainable Food Systems*, 3:54, 2019. 3

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 12, 13, 26

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics, 2019. 18, 79, 81, 83

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`. 13

[29] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: ESA's optical high-resolution mission for GMES operational services. *Remote sensing of Environment*, 120:25–36, 2012. 4, 10, 52

[30] Kenji Enomoto, Ken Sakurada, Weimin Wang, Hiroshi Fukui, Masashi Matsuoka, Ryosuke Nakamura, and Nobuo Kawaguchi. Filmy cloud removal on satellite imagery with multispectral conditional generative adversarial nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 48–56, 2017. 39

[31] Kilian Fatras, Thibault Séjourné, Nicolas Courty, and Rémi Flamary. Unbalanced minibatch optimal transport; applications to domain adaptation. In *International Conference on Machine Learning*, 2021. 55, 67, 69, 70, 71

[32] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2960–2967, 2013. 55

[33] Saskia Foerster, Klaus Kaden, Michael Foerster, and Sibylle Itzerott. Crop type mapping using spectral–temporal profiles and phenological information. *Computers and Electronics in Agriculture*, 89:30–40, 2012. 11

[34] Steve Foga, Pat L. Scaramuzza, Song Guo, Zhe Zhu, Ronald D. Dilley, Tim Beckmann, Gail L. Schmidt, John L. Dwyer, M. Joseph Hughes, and Brady Laue. Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sensing of Environment*, 194:379 – 390, 2017. ISSN 0034-4257. 26, 39, 46, 49

[35] B. Franch, E.F. Vermote, I. Becker-Reshef, M. Claverie, J. Huang, J. Zhang, C. Justice, and J.A. Sobrino. Improving the timeliness of winter wheat production forecast in the United States of America, Ukraine and China using MODIS data and NCAR Growing Degree Day information. *Remote Sensing of Environment*, 161:131–148, 2015. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2015.02.014. URL `https://www.sciencedirect.com/science/article/pii/S003442571500067X`. 15, 78, 79, 84

[36] Kun Fu, Wanxuan Lu, Wenhui Diao, Menglong Yan, Hao Sun, Yi Zhang, and Xian Sun. WSF-NET: Weakly supervised feature-fusion network for binary segmentation in remote sensing image. *Remote Sensing*, 10(12):1970, 2018. 38, 40

[37] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015. 53, 55, 69, 70

[38] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 55, 66

[39] Feng Gao, Martha C. Anderson, Xiaoyang Zhang, Zhengwei Yang, Joseph G. Alfieri, William P. Kustas, Rick Mueller, David M. Johnson, and John H. Prueger. Toward mapping crop progress at field scales through fusion of landsat and MODIS imagery. *Remote Sensing of Environment*, 188:9 – 25, 2017. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2016.11.004. 38

[40] V Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano, and Nesrine Chehata. Time-space tradeoff in deep learning models for crop classification on satellite multi-spectral image time series. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 6247–6250. IEEE, 2019. 17

[41] Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4872–4881, 2021. 15, 21, 79, 80, 81

[42] Yufeng Ge, J Alex Thomasson, and Ruixiu Sui. Remote sensing of soil properties in precision agriculture: A review. *Frontiers of Earth Science*, 5(3): 229–238, 2011. 3

[43] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR, 2017. 83

[44] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 13

[45] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 12

[46] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 12

[47] Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R Sveinsson. Random forests for land cover classification. *Pattern recognition letters*, 27(4): 294–300, 2006. 11

[48] Cristina Gómez, Joanne C White, and Michael A Wulder. Optical remotely sensed time series data for land cover classification: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116:55–72, 2016. 11

[49] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012. 55

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3, 12, 38, 41, 42

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. 4

[52] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013. 12

[53] C. Grohnfeldt, M. Schmitt, and X. Zhu. A conditional generative adversarial network to fuse sar and multispectral optical data for cloud removal from sentinel-2 images. In *2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1726–1729, 2018. 39

[54] Olivier Hagolle, Mireille Huc, D Villa Pascual, and Gérard Dedieu. A multi-temporal method for cloud detection, applied to FORMOSAT-2, VEN$\mu$S, LANDSAT and SENTINEL-2 images. *Remote Sensing of Environment*, 114 (8):1747–1755, 2010. 14, 38

[55] Pengyu Hao, Yulin Zhan, Li Wang, Zheng Niu, and Muhammad Shakir. Feature selection of time series modis data for early crop classification using random forest: A case study in kansas, usa. *Remote Sensing*, 7(5):5347–5369, 2015. 11

[56] Pengyu Hao, Liping Di, Chen Zhang, and Liying Guo. Transfer learning for crop classification with cropland data layer data (CDL) as training samples. *Science of The Total Environment*, 733:138869, 2020. ISSN 0048-9697.

doi: https://doi.org/10.1016/j.scitotenv.2020.138869. URL https://www.sciencedirect.com/science/article/pii/S004896972032386X. 78, 79

[57] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009. 3

[58] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. 75

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 12, 46

[60] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint*, abs/1606.08415, 2016. 86

[61] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 54, 59

[62] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 12, 17

[63] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, and Douglas Eck. An improved relative self-attention mechanism for transformer with application to music generation. *arXiv preprint*, abs/1809.04281, 2018. URL http://arxiv.org/abs/1809.04281. 81

[64] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018. 26, 40

[65] Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1685–1689, 2017. 17, 52, 78

[66] Jordi Inglada, Marcela Arias, Benjamin Tardy, Olivier Hagolle, Silvia Valero, David Morin, Gérard Dedieu, Guadalupe Sepulcre, Sophie Bontemps, Pierre Defourny, et al. Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery. *Remote Sensing*, 7(9):12356–12379, 2015. 80

[67] Roberto Interdonato, Dino Ienco, Raffaele Gaetano, and Kenji Ose. DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:91–104, 2019. doi: https://doi.org/10.1016/j.isprsjprs.2019.01.011. 18, 52, 79, 80

[68] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 71, 85

[69] James R Irons, John L Dwyer, and Julia A Barsi. The next landsat satellite: The landsat data continuity mission. *Remote Sensing of Environment*, 122:11–21, 2012. 10

[70] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 26, 39, 41, 42

[71] Jacob Høxbroe Jeppesen, Rune Hylsberg Jacobsen, Fadil Inceoglu, and Thomas Skjødeberg Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. *Remote sensing of environment*, 229:247–259, 2019. 4, 14

[72] Jacob Høxbroe Jeppesen, Rune Hylsberg Jacobsen, Fadil Inceoglu, and Thomas Skjødeberg Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sensing of Environment*, 229:247 – 259, 2019. ISSN 0034-4257. 38, 45, 49

[73] Shunping Ji, Chi Zhang, Anjian Xu, Yun Shi, and Yulin Duan. 3d convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote Sensing*, 10(1):75, 2018. 18

[74] Kun Jia, Shunlin Liang, Xiangqin Wei, Yunjun Yao, Yingru Su, Bo Jiang, and Xiaoxia Wang. Land cover classification of Landsat data with phenological features extracted from time series MODIS NDVI data. *Remote Sensing*, 6(11): 11518–11532, 2014. 80

[75] Junguang Jiang, Baixu Chen, Bo Fu, and Mingsheng Long. Transfer learning library. `https://github.com/thuml/Transfer-Learning-Library`, 2020. 71

[76] Hamlyn G Jones. *Plants and microclimate: a quantitative approach to environmental plant physiology*. Cambridge University Press, 2013. 31, 79

[77] Per Jönsson and Lars Eklundh. Timesat—a program for analyzing time-series of satellite sensor data. *Computers & geosciences*, 30(8):833–845, 2004. 11, 32

[78] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00453. 12

[79] Benjamin Kellenberger, Onur Tasar, Bharath Bhushan Damodaran, Nicolas Courty, and Devis Tuia. *Deep Domain Adaptation in Earth Observation*, chapter 7, pages 90–104. John Wiley & Sons, Ltd, 2021. ISBN 9781119646181. 53

[80] Hannah Kerner, Ritvik Sahajpal, Sergii Skakun, Inbal Becker-Reshef, Brian Barker, Mehdi Hosseini, Estefania Puricelli, and Patrick Gray. Resilient in-season crop type classification in multispectral satellite observations using growth stage normalization. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining Workshops*, 2020. 5, 32, 79, 80, 82

[81] Sami Khanal, John Fulton, and Scott Shearer. An overview of current and potential applications of thermal remote sensing in precision agriculture. *Computers and Electronics in Agriculture*, 139:22–32, 2017. 3

[82] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`. 47, 70, 88

[83] Brad Koch, Raj Khosla, WM Frasier, DG Westfall, and D Inman. Economic feasibility of variable-rate nitrogen application utilizing site-specific management zones. *Agronomy Journal*, 96(6):1572–1580, 2004. 3

[84] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European conference on computer vision*, pages 695–711. Springer, 2016. 40

[85] Lukas Kondmann, Aysim Toker, Marc Rußwurm, Andrés Camero, Devis Peressuti, Grega Milcinski, Pierre-Philippe Mathieu, Nicolas Longépé, Timothy Davis, Giovanni Marchisio, Laura Leal-Taixé, and Xiao Xiang Zhu. DENETHOR: The DynamicEarthNET dataset for harmonized, inter-operable, analysis-ready, daily crop monitoring from space. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 53

[86] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 3, 12

[87] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017. 80

[88] Loic Landrieu and Vivien Sainte Fare Garnot. Leveraging Class Hierarchies with Metric-Guided Prototype Learning. In *British Machine Vision Conference (BMVC)*, Virtual, United Kingdom, November 2021. URL `https://hal.archives-ouvertes.fr/hal-03500516`. 22

[89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 12

[90] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 105–114. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.19. URL `https://doi.org/10.1109/CVPR.2017.19`. 12

[91] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 896, 2013. 53, 56

[92] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. *Advances in Neural Information Processing Systems*, 34, 2021. 81, 83, 85

[93] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *Pattern Recognition*, 80, 03 2016. doi: 10.1016/j.patcog.2018.03.005. 71

[94] Zhiwei Li, Huanfeng Shen, Qing Cheng, Yuhao Liu, Shucheng You, and Zongyi He. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:197–212, 2019. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2019.02.017. URL `https://www.sciencedirect.com/science/article/pii/S0924271619300565`. 14

[95] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2980–2988, 2017. 63

[96] Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. Learning to encode position for transformer with continuous dynamical model. In

*International Conference on Machine Learning*, pages 6327–6335. PMLR, 2020. 81, 83, 86

[97] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 12

[98] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1647–1657, 2018. 55, 66, 69, 70, 71

[99] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 70, 88

[100] Benjamin Lucas, Charlotte Pelletier, Daniel Schmidt, Geoffrey I Webb, and François Petitjean. Unsupervised domain adaptation techniques for classification of satellite image time series. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1074–1077. IEEE, 2020. doi: 10.1109/IGARSS39084.2020.9324339. 4, 5, 53, 55, 80

[101] Benjamin Lucas, Charlotte Pelletier, Daniel Schmidt, Geoffrey I Webb, and François Petitjean. A bayesian-inspired, deep learning-based, semi-supervised domain adaptation technique for land cover mapping. *Machine Learning*, pages 1–33, 2021. 53, 58, 78

[102] Rose M Rustowicz, Robin Cheong, Lijing Wang, Stefano Ermon, Marshall Burke, and David Lobell. Semantic segmentation of crop type in Africa: A novel dataset and analysis of deep learning methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 75–82, 2019. 79

[103] Iwake Masialeti, Stephen Egbert, and Brian D Wardlow. A comparative analysis of phenological curves for major crops in kansas. *GIScience & Remote Sensing*, 47(2):241–259, 2010. 11

[104] Gregory S. McMaster and W.W. Wilhelm. Growing degree-days: one equation, two interpretations. *Agricultural and Forest Meteorology*, 87(4):291–300, 1997. ISSN 0168-1923. doi: https://doi.org/10.1016/S0168-1923(97)00027-0. URL `https://www.sciencedirect.com/science/article/pii/S0168192397000270`. 29, 31, 79, 80, 84

[105] HJ Mederski, ME Miller, and CR Weaver. Accumulated heat units for classifying corn hybrid maturity 1. *Agronomy Journal*, 65(5):743–747, 1973. 6, 29, 31, 79, 80

[106] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL `http://arxiv.org/abs/1301.3781`. 12

[107] Perry Miller, Will Lanier, and Stu Brandt. Using growing degree days to predict plant stages. *Ag/Extension Communications Coordinator, Communications Services, Montana State University-Bozeman, Bozeman, MO*, 59717(406): 994–2721, 2001. 79, 84

[108] Dinh Ho Tong Minh, Dino Ienco, Raffaele Gaetano, Nathalie Lalande, Emile Ndikumana, Faycal Osman, and Pierre Maurel. Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR Sentinel-1. *IEEE Geoscience and Remote Sensing Letters*, 15(3):464–468, 2018. 17, 52, 78

[109] Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *Proceedings of the 29th International conference on machine learning (ICML-12)*, pages 567–574, 2012. 45

[110] Pietro Morerio, Riccardo Volpi, Ruggero Ragonesi, and Vittorio Murino. Generative pseudo-label refinement for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3130–3139, 2020. 53, 56

[111] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814. Omnipress, 2010. URL `https://icml.cc/Conferences/2010/papers/432.pdf`. 85

[112] Emile Ndikumana, Dinh Ho Tong Minh, Nicolas Baghdadi, Dominique Courault, and Laure Hossard. Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France. *Remote Sensing*, 10(8):1217, 2018. 17, 52, 78, 80

[113] Masato Neishi and Naoki Yoshinaga. On the relation between position information and sentence length in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338, 2019. 81, 86

[114] Adrien Nivaggioli and Hicham Randrianarivo. Weakly supervised semantic segmentation of satellite images. In *2019 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4. IEEE, 2019. 38, 40

[115] Joachim Nyborg and Ira Assent. Weakly-Supervised Cloud Detection with Fixed-Point GANs. In *2021 IEEE International Conference on Big Data*

*(Big Data)*, pages 4191–4198. IEEE, 2021. doi: 10.1109/BigData52589.2021.9671405. 6, 25

[116] Joachim Nyborg, Charlotte Pelletier, Sébastien Lefèvre, and Ira Assent. The TimeMatch Dataset, 2021. 54, 64, 66, 79, 86, 87

[117] Joachim Nyborg, Charlotte Pelletier, and Ira Assent. Generalized classification of satellite image time series with thermal positional encoding. *arXiv preprint*, abs/2203.09175, 2022. 6

[118] Joachim Nyborg, Charlotte Pelletier, Sébastien Lefèvre, and Ira Assent. TimeMatch: Unsupervised cross-region adaptation by temporal shift estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2022. 6, 27, 30, 78, 79, 80, 82, 83, 87, 88, 89, 90, 91

[119] Julie B. Odenweller and Karen I. Johnson. Crop identification using landsat temporal-spectral profiles. *Remote Sensing of Environment*, 14(1):39–54, 1984. ISSN 0034-4257. doi: 10.1016/0034-4257(84)90006-3. 15, 52

[120] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009. doi: 10.1109/TKDE.2009.191. 53

[121] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017. 30, 58

[122] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019. 70

[123] Planet Labs PBC. Planet application program interface: In space for life on earth, 2018–. URL `https://api.planet.com`. 10

[124] Charlotte Pelletier, Silvia Valero, Jordi Inglada, Nicolas Champion, Claire Marais Sicre, and Gérard Dedieu. Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sensing*, 9(2):173, 2017. 4

[125] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11(5):523, 2019. doi: 10.3390/rs11050523. 11, 18, 52, 78, 79, 80

[126] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*, 2017. 55, 69

[127] Shi Qiu, Binbin He, Zhe Zhu, Zhanmang Liao, and Xingwen Quan. Improving Fmask cloud and cloud shadow detection in mountainous area for Landsats 4–8 images. *Remote Sensing of Environment*, 199:107–119, 2017. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2017.07.002. URL `https://www.sciencedirect.com/science/article/pii/S0034425717303073`. 14

[128] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 79, 81, 83

[129] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007. 85

[130] Md Mahfuzur Rahman Siddiquee, Zongwei Zhou, Nima Tajbakhsh, Ruibin Feng, Michael B Gotway, Yoshua Bengio, and Jianming Liang. Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 191–200, 2019. 26, 38, 39, 40, 41, 43, 46

[131] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 12

[132] Bradley C Reed, Jesslyn F Brown, Darrel VanderZee, Thomas R Loveland, James W Merchant, and Donald O Ohlen. Measuring phenological variability from satellite imagery. *Journal of Vegetation Science*, 5(5):703–714, 1994. 15, 52

[133] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017. 26, 45

[134] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 12, 17, 26

[135] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 21, 40, 45, 47

[136] Marc Rußwurm and Marco Körner. Self-attention for raw optical satellite time series classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:421–435, 2020. doi: 10.1016/j.isprsjprs.2020.06.006. 18, 20, 21, 28, 52, 78, 79, 80

[137] Marc Rußwurm and Marco Körner. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 11–19, 2017. 17, 52, 56, 78, 80

[138] Marc Rußwurm and Marco Körner. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS International Journal of Geo-Information*, 7(4):129, 2018. doi: 10.3390/ijgi7040129. URL `https://doi.org/10.3390/ijgi7040129`. 11, 15, 17, 28, 52, 79, 80

[139] Marc Rußwurm, Sherrie Wang, Marco Korner, and David Lobell. Meta-learning for few-shot land cover classification. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops*, pages 200–201, 2020. 4

[140] Rose M. Rustowicz, Robin Cheong, Lijing Wang, Stefano Ermon, Marshall Burke, and David B. Lobell. Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 75–82. Computer Vision Foundation / IEEE, 2019. URL `http://openaccess.thecvf.com/content_CVPRW_2019/html/cv4gc/Rustowicz_Semantic_Segmentation_of_Crop_Type_in_Africa_A_Novel_Dataset_CVPRW_2019_paper.html`. 15, 17, 18, 28

[141] Vivien Sainte Fare Garnot and Loic Landrieu. Lightweight temporal self-attention for classifying satellite images time series. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 171–181. Springer, 2020. 20, 21, 30, 65, 70, 78, 80, 81, 82, 83, 84, 87, 88, 89, 90, 91

[142] Vivien Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano, and Nesrine Chehata. Satellite image time series classification with pixel-set encoders and temporal self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12325–12334, 2020. 11, 13, 17, 20, 21, 28, 52, 56, 65, 70, 78, 79, 80, 85

[143] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017. 53, 56

[144] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019. 71

[145] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, 29:2234–2242, 2016. 60

[146] Vishnu Sarukkai, Anirudh Jain, Burak Uzkent, and Stefano Ermon. Cloud removal from satellite images using spatiotemporal generator networks. In *Winter Conference on Applications of Computer Vision*, pages 1796–1805, 2020. 39

[147] Maja Schneider, Amelie Broszeit, and Marco Körner. EuroCrops: A pan-european dataset for time series crop type classification. *arXiv preprint*, abs/2106.08151, 2021. URL `https://arxiv.org/abs/2106.08151`. 5, 15, 81, 88

[148] Michal Segal-Rozenhaimer, Alan Li, Kamalika Das, and Ved Chirayath. Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks. *Remote Sensing of Environment*, 237:111446, 2020. ISSN 0034-4257. 38

[149] Joel Segarra, Maria Luisa Buchaillot, Jose Luis Araus, and Shawn C Kefauver. Remote sensing for precision agriculture: Sentinel-2 improved features and applications. *Agronomy*, 10(5):641, 2020. 10

[150] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 40, 46, 47

[151] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 464–468. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2074. URL `https://doi.org/10.18653/v1/n18-2074`. 81

[152] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 17

[153] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 53, 56

[154] P. Singh and N. Komodakis. Cloud-GAN: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. In *2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1772–1775, July 2018. 39

[155] Rajendra P Sishodia, Ram L Ray, and Sudhir K Singh. Applications of remote sensing in precision agriculture: A review. *Remote Sensing*, 12(19):3136, 2020. 3, 10

[156] Sergii Skakun, Nataliia Kussul, Andrii Shelestov, and Olga Kussul. The use of satellite data for agriculture drought risk quantification in ukraine. *Geomatics, Natural Hazards and Risk*, 7(3):901–917, 2016. 15

[157] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fix-Match: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020. 56, 58, 59, 62, 63, 69, 70, 71, 72

[158] Ancha Srinivasan. *Handbook of precision agriculture: principles and applications*. CRC press, 2006. 3

[159] Andrei Stoian, Vincent Poulain, Jordi Inglada, Victor Poughon, and Dawa Derksen. Land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems. *Remote Sensing*, 11(17):1986, 2019. 81

[160] C. J. Stubenrauch, W. B. Rossow, S. Kinne, S. Ackerman, G. Cesana, H. Chepfer, L. Di Girolamo, B. Getzewich, A. Guignard, A. Heidinger, B. C. Maddux, W. P. Menzel, P. Minnis, C. Pearl, S. Platnick, C. Poulsen, J. Riedi, S. Sun-Mack, A. Walther, D. Winker, S. Zeng, and G. Zhao. Assessment of global cloud datasets from satellites: Project and database initiated by the gewex radiation panel. *Bulletin of the American Meteorological Society*, 94(7):1031 – 1049, 2013. doi: 10.1175/BAMS-D-12-00117.1. URL `https://journals.ametsoc.org/view/journals/bams/94/7/bams-d-12-00117.1.xml`. 14

[161] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016. 55

[162] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 4

[163] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014. 12

[164] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 12

[165] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017. 56, 58, 62, 64

[166] David Trudgill, Alois Honek, Daiqin Li, and Nico M. van Straalen. Thermal time - concepts and utility. *Annals of Applied Biology*, 146:1–14, 01 2005. doi: 10.1111/j.1744-7348.2005.04088.x. 84

[167] Compton J Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150, 1979. 10, 80

[168] Devis Tuia, Claudio Persello, and Lorenzo Bruzzone. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):41–57, 2016. 4, 5, 53, 57, 80

[169] Mehmet Ozgur Turkoglu, Stefano D'Aronco, Gregor Perich, Frank Liebisch, Constantin Streit, Konrad Schindler, and Jan Dirk Wegner. Crop mapping from image time series: Deep learning with multi-scale label hierarchies. *Remote Sensing of Environment*, 264:112603, 2021. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2021.112603. URL `https://www.sciencedirect.com/science/article/pii/S0034425721003230`. 22

[170] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint*, abs/1412.3474, 2014. 28, 53, 55, 66, 69, 70, 80

[171] Silvia Valero, David Morin, Jordi Inglada, Guadalupe Sepulcre, Marcela Arias, Olivier Hagolle, Gérard Dedieu, Sophie Bontemps, Pierre Defourny, and Benjamin Koetz. Production of a dynamic cropland mask by processing remote sensing image series at high temporal and spatial resolutions. *Remote Sensing*, 8(1):55, 2016. 80

[172] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008. 74, 75, 89, 91

[173] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 3, 12, 18, 19, 30, 31, 65, 79, 80, 81, 82, 85, 88

[174] Francesco Vuolo, Martin Neuwirth, Markus Immitzer, Clement Atzberger, and Wai-Tim Ng. How much does multi-temporal Sentinel-2 data improve crop type classification? *International Journal of Applied Earth Observation and Geoinformation*, 72:122–130, 2018. 11, 52, 80

[175] Guido Waldhoff, Ulrike Lussem, and Georg Bareth. Multi-data approach for remote sensing-based regional crop rotation mapping: A case study for the rur catchment, germany. *International Journal of Applied Earth Observation and Geoinformation*, 61:55–69, 2017. ISSN 0303-2434. doi: https://doi.org/10.1016/j.jag.2017.04.009. URL `https://www.sciencedirect.com/science/article/pii/S0303243417300934`. 5

[176] Sherrie Wang, George Azzari, and David B. Lobell. Crop type mapping without field-level labels: Random forest transfer and unsupervised clustering techniques. *Remote Sensing of Environment*, 222:303–317, 2019. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2018.12.026. URL `https://www.sciencedirect.com/science/article/pii/S0034425718305790`. 80

[177] Sherrie Wang, William Chen, Sang Michael Xie, George Azzari, and David B Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 12(2):207, 2020. 4, 38, 40

[178] Sherrie Wang, Stefania Di Tommaso, Jillian M Deines, and David B Lobell. Mapping twenty years of corn and soybean across the us midwest using the landsat archive. *Scientific Data*, 7(1):1–14, 2020. 5

[179] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020. 26, 40

[180] Ziqiao Wang, Hongyan Zhang, Wei He, and Liangpei Zhang. Phenology alignment network: A novel framework for cross-regional time series crop classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2940–2949, 2021. 28, 53, 55, 69, 79, 80

[181] Brian D Wardlow and Stephen L Egbert. Large-area crop mapping using time-series MODIS 250 m NDVI data: An assessment for the US Central Great Plains. *Remote Sensing of Environment*, 112(3):1096–1116, 2008. 80

[182] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1576, 2017. 26, 40

[183] Mette Wik, Prabhu Pingali, and Sumiter Brocai. Global agricultural performance: past trends and future prospects. 2008. 3

[184] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(5): 1–46, 2020. doi: 10.1145/3400066. 53, 54

[185] Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1768–1778, 2020. 55, 69

[186] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 82

[187] F. Xie, M. Shi, Z. Shi, J. Yin, and D. Zhao. Multilevel cloud detection in remote sensing images based on deep learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3631–3640, 2017. ISSN 2151-1535. doi: 10.1109/JSTARS.2017.2686488. 38

[188] Pavel Yakubovskiy. Segmentation models Pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2020. 47

[189] Senshan Yang, Joanne Logan, and David L Coffey. Mathematical formulae for calculating the base temperature for growing degree days. *Agricultural and Forest Meteorology*, 74(1-2):61–74, 1995. 79

[190] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI conference on artificial intelligence*, 2017. 3

[191] Yongjie Zhan, Jian Wang, Jianping Shi, Guangliang Cheng, Lele Yao, and Weidong Sun. Distinguishing cloud and snow in satellite images via deep convolutional network. *IEEE Geosci. Remote. Sens. Lett.*, 14(10):1785–1789, 2017. doi: 10.1109/LGRS.2017.2735801. URL `https://doi.org/10.1109/LGRS.2017.2735801`. 14

[192] Liheng Zhong, Peng Gong, and Gregory S Biging. Efficient corn and soybean mapping with temporal extendability: A multi-year experiment using landsat imagery. *Remote Sensing of Environment*, 140:1–13, 2014. 80

[193] Liheng Zhong, Lina Hu, and Hang Zhou. Deep learning based multi-temporal crop classification. *Remote Sensing of Environment*, 221:430–443, 2019. 18, 52, 78, 80

[194] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 26, 40, 46, 47

[195] Xudong Zhou, Xinkai Zhu, Zhaodi Dong, Wenshan Guo, et al. Estimation of biomass in wheat using random forest regression algorithm and remote sensing data. *The Crop Journal*, 4(3):212–219, 2016. 11

[196] Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. Activation maximization generative adversarial nets. In *International Conference on Learning Representations*, 2018. 60, 61

[197] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 12, 26, 39, 41, 42

[198] Zhe Zhu. Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:370 – 384, 2017. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2017.06.013. 38

[199] Zhe Zhu, Shixiong Wang, and Curtis E. Woodcock. Improvement and expansion of the fmask algorithm: cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159: 269 – 277, 2015. ISSN 0034-4257. 14, 27, 38

[200] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 53, 56

[201] Zhengxia Zou, Wenyuan Li, Tianyang Shi, Zhenwei Shi, and Jieping Ye. Generative adversarial training for weakly supervised cloud matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 201–210, 2019. 27