UNIVERSITÄT BONN

Exploring Self-Supervised Learning Methods for Landcover Applications Using Remote Sensing Data

DISSERTATION

zur Erlangung des Doktorgrades (*Dr.rer.nat.*)
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Ankit Patnala
aus
Berhampur, Indien

Bonn, 2025

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer / Reviewer/Supervisor: Prof. Dr. Martin Schultz

Gutachter / Reviewer: Prof. Dr. Jürgen Gall

Tag der Promotion/ Day of Promotion: 22.10.2025

Erscheinungsjahr/ Year of Publication: 2026

Lists of Figures and Tables

Figures

Tables

Abstract

Exploring Self-Supervised Learning Methods for Land Cover Applications using Remote Sensing Data

Anthropogenic activities such as crop cultivation, forest conservation, urban development, and industrial establishment significantly alter the Earth's surface characteristics. Thus, assessing such alteration to the Earth's surface is vital for both society and the economy. Monitoring and analyzing the Earth's surface helps to understand and manage the limited natural resources, making it essential for sustainable land use planning and informed decision-making processes. Analyzing the land surface usage thereby involves the intersection of geospatial data, remote sensing technologies, and modeling techniques. While advancements in remote sensing have dramatically increased the availability of data, there remains a need for specialized expertise to handle and interpret such data. While the large amount of remote sensing data enables the use of machine learning, annotating data is a time-consuming and expensive process limiting the applicability of supervised training techniques. Recent developments in self-supervised learning, particularly contrastive learning, have shown promising results. A key advantage is that this method is largely independent of manual annotation since the target for the optimization process can be constructed from the available data itself. Self-supervised learning uses a two-step approach: first, the machine learning model is trained on unlabeled data in a fully self-supervised way. In the second step, transfer learning techniques are applied to adapt the pre-trained model to an annotated dataset. Due to the pre-training, the required amount of annotated data is significantly reduced compared to applications where the machine learning model is trained on annotated data only. Thus, self-supervised learning is more efficient and cost-effective, addressing the challenges associated with manual annotation and offering substantial benefits in Earth observation applications. However, these techniques have been primarily designed for natural images in computer vision. The multi-spectral imagery of Earth observations exhibits unique challenges that set them apart from standard computer vision tasks. For instance, Earth observation data often includes multiple spectral bands beyond the visible spectrum, contains temporal information, and requires domain-specific knowledge for interpretation. By addressing these differences, this thesis aims to bridge the gap between traditional self-supervised learning techniques and the specific needs of Earth observation.

The initial phase of this research highlights the importance of spectral bands beyond RGB in land cover analysis. Building on these findings, atmospheric transformation is proposed for contrastive self-supervised learning on remote sensing images, addressing the challenge of meaningfully handling multiple spectral bands. Upon comparing against a baseline, the following method was superior. While effective for land cover classification, these methods are less suitable for time series crop classification. To develop a self-supervised approach for time series data,

multi-modal self-supervised methods for crop classification have been proposed. When evaluated on 9 different time series crop classification tasks, the multi-modal approach outperforms existing uni-modal approaches developed for tabular data such as spectral measurements. To leverage the multi-modal characteristics of remote sensing data and the sequential nature of spectro-temporal data, the approach is further refined by adapting the bi-modal BERT training technique, a prominent self-supervised algorithm from natural language processing. Although these methods effectively handle complex multi-spectral temporal data for crop classification, future research should explore techniques that can also utilize dense spatial information to develop more capable models.

Contents

CHAPTER 1

Introduction

Contents

1.1 Motivation

Land cover is defined as the physical material present on the outer surface of the Earth, such as grasses, trees, lands, water bodies, etc. Understanding land cover and its dynamics is crucial for several reasons. The Earth's surface has undergone rapid transformations in the recent past due to anthropogenic activities and natural phenomena. These changes profoundly impact ecosystems, biodiversity, and human societies. Land cover alterations influence various ecological processes, including the water cycle, carbon cycle, and climate patterns.

The term land cover is often associated with land use and land cover. Land use refers to the activities undertaken by people on a particular land cover type. For instance, grasslands provide habitation to plant, animal, and bird species, while urban areas are used for residential, commercial, and industrial purposes. The industrial purposes includes manufacturing, warehousing, distribution, and technological advancement, and the conversion

of natural areas results in loss of biodiversity. The land cover, through land surface evapotranspiration, is also one of the factors that influence Earth's climate (Jasechko et al. 2013). Numerous other examples underscore the influence of land cover on a wide range of ecological, social, and economic processes. That's why studying and understanding the land surface is crucial.

Comprehending the static and dynamic nature of land cover is crucial for devising conservation strategies and policies aimed at effective land management. Several programs, such as the International Geosphere-Biosphere Program (IGBP) (Loveland and Belward 1997) and the Land Cover and Land Use Change program, have emphasized the need to study land cover dynamics at local, national, continental, and global levels for better awareness[1]. National Aeronautics and Space Administration (NASA) uses land surface information to assess the ecological state of the planet (*NASA Earth Data: Land Surface* 2024).

In order to study the land surface, it is important to study and analyze each land use separately. The process of creating detailed maps delineating different land cover types, such as forests, water bodies, and industrial areas, is called land cover mapping. There are two primary approaches to generating these maps: field campaigns and analysis of remote sensing imagery. This work focuses on the latter approach.

NASA's 1972 launch of the Earth Resources Technology Satellite, later renamed Landsat, marked a significant milestone in remote sensing. This satellite pioneered the capture of land surface data across four spectral bands with a spatial resolution of approximately 80 meters (modern publicly available satellites provide surface data at a spatial resolution of 10-30 meters). Despite the coarse resolution, researchers utilized spectral indices derived from this data for land cover mapping.

In the following years, a series of satellite missions with enhanced spectral and spatial capabilities were deployed, revolutionizing land cover mapping applications. Instruments such as Advanced Very High-Resolution Radiometer (AVHRR) aboard NOAA polar-orbiting satellites, Moderate Resolution Imaging Spectroradiometer (MODIS) onboard NASA's Terra and Aqua satellites, SPOT Vegetation instrument on the SPOT-4 and SPOT-5 satellites, and the Envisat mission by the European Space Agency (ESA) (Tucker, Townshend, and Goff 1985; Stone et al. 1994; Tateishi and Kajiwara 1991; Achard and Estreguil 1995) significantly expanded the scope and precision of Earth observation data. Among these instruments and missions, only MODIS remains operational, while the others have been decommissioned.

In 2015, the European Space Agency (ESA) launched the Sentinel2, an optical remote sensing satellite mission advancing Earth observation with its high-resolution measurements, fine temporal scale, and global coverage (Drusch et al. 2012). A detailed overview of Sentinel2's spectral bands and their resolutions is provided in Chapter 2. The open accessibility of Sentinel2 data has led to its widespread adoption across various research domains. Sentinel2 offers frequent, high-resolution, up-to-date global maps, and has significantly enhanced analysis capabilities by providing satellite derivatives at improved spatial and temporal resolutions (Kussul et al. 2017). The extensive time series data from

[1] https://eo4society.esa.int/wp-content/uploads/2021/02/D3T1b_LTC2015_Caetano.pdf

Sentinel2 has enabled the creation of sophisticated maps, broadening applications to fields that rely on temporal patterns (Inglada et al. 2015). As Earth observation data continues to evolve, the ongoing development of novel methodologies will hold significant potential for further advancements. Existing satellite missions provide a continuous stream of data with high temporal frequency. However, the available data are often complex and challenging to comprehend due to various factors, such as the presence of clouds in the atmosphere, calibration errors in sensors, and differences in alignment between measurements. These factors contribute to the presence of noise in the data, thus complicating its analysis. Nevertheless, such a large amount of available data does facilitate the use of machine learning.

In the field of Earth observation, a significant portion of the data is presented as patches covering spatial regions over the Earth's surface. These patches can be referred to as natural images. The computer vision community actively developed methods and algorithms to extract useful information from images, enabling their analysis for various applications. Common tasks that employ these algorithms are image classification, image segmentation (semantic and instance), object detection, image reconstruction, and others. These tasks are also relevant to Earth observation applications, such as land cover classification (image classification), land cover segmentation (image segmentation), object detection (e.g., parks, airports) on the land surface, and land cover retrieval (image reconstruction). However, unlike natural images that typically capture reflections from the three visible channels (Red, Green, and Blue), these patches encompass a broader range of channels. These include visible channels, near-infrared channels, and short-wave infrared channels. Nevertheless, still, the similarity that exists between Earth observation data and natural images motivates the adaption of computer vision methods to the Earth observation remote sensing data.

DeepSat (Basu et al. 2015) was one of the pioneering works that identified the challenges at the intersection of remote sensing, computer vision, and machine learning. Using small-sized datasets, the authors demonstrated that their proposed approach, which involved extracting meaningful features from images using spectral indices and transforming them into 1D vectors, normalizing them and feeds the normalized features to deep belief network (DBN) classification. This could outperform conventional methods like Random Forest as well as the then-modern shallow convolutional neural networks (CNNs) when used with a deep belief network (DBN). The authors concluded that satellite datasets exhibit high intra- and inter-class variability, and shape or edge-based features are not sufficiently distinctive for most land cover classification tasks. They highlighted the difficulty in obtaining the manual annotation of these images. Additionally, they pointed out that the small size of available datasets hindered the potential of more advanced networks, such as deeper CNNs and transformer networks (Vaswani et al. 2017), which are widely used in the field of computer vision. These sophisticated networks, with their higher number of training parameters, require vast amounts of data to effectively capture important image features.

1.2 Challenges

A primary issue in applying machine learning to Earth observation is the limited availability of reference data. As mentioned earlier, the ground truth information and annotated labels are essential for training large-scale models. The success of modern deep learning algorithms in computer vision can be attributed to the availability of massive datasets like ImageNet (Russakovsky et al. 2015), Instagram 1B comprising over a billion Instagram images (Yalniz et al. 2019), and Wikipedia data (Dinan et al. 2018). However, annotating Earth observation data necessitates domain expertise and often involves a laborious, expensive, and time-consuming process of field campaigns. Furthermore, Earth observation data presents additional difficulties, including the challenge of obtaining balanced datasets, dealing with sparse information, and handling point observations that lack contextual information from the surroundings. Overcoming these obstacles is crucial for leveraging the full potential of machine learning techniques in Earth observation applications.

The Earth observation community has recognized the need for large-scale datasets and has begun addressing this requirement. Datasets such as DeepGlobe (I. Demir et al. 2018), Bigearthnet (Sumbul et al. 2019), and Sen12MS (Schmitt et al. 2019) are examples of efforts to create datasets with up to the order of million samples. However, compared to datasets in computer vision, these Earth observation benchmark datasets are still relatively small. Researchers often rely on information from land cover maps, which are not frequently updated, to obtain labels for these datasets. While most supervised learning algorithms employing deep networks have demonstrated the ability to learn from this data, but the process of collecting and annotating data remains tedious, especially in regions where up-to-date land cover maps are not available.

To overcome the limitation of annotated data in supervised learning, self-supervised learning methods have been developed. The self-supervised learning methods have recently gained considerable attention and has seen widespread adoption in the machine learning community. These methods have the ability to leverage unlabeled data, thereby eliminating the dependence on labeled data. Furthermore, self-supervised learning techniques can be seamlessly integrated with modern deep learning architectures, which are well suited for modeling complex, high-dimensional data such as images. This integration makes them particularly effective for domains with abundant unlabeled data and intricate data structures where traditional unsupervised algorithms often fail. Although self-supervised learning starts with a vast amount of unlabeled data, techniques like transfer learning, fine-tuning, and zero-shot learning allow the pre-trained models to be applied to new tasks with fewer labeled samples and achieve better or equal compared to applications trained in a fully supervised way. The self-supervised learning has great potential for applications in the field of Earth observation. Satellite missions such as Sentinel2 (ESA 2021) and Landsat (Cai et al. 2018) have been actively monitoring the entire globe for several years, providing a vast amount of data with the temporal resolution of a few days. With newer satellite constellations such as Planetscope (PBC 2017), the temporal and spatial resolutions have significantly improved, further augmenting the availability of

landcover data. Although these datasets are commercially acquired, they play an increasingly important role in high-resolution Earth observation studies. Despite having such vast amounts of data, manually annotating them remains a demanding and resource-intensive task. So, self-supervised learning is believed to be beneficial for these applications. However, it poses different challenges. One of the biggest challenge is that most of the self-supervised algorithms are developed for natural images and text but as mentioned earlier, the properties of Earth observation are fundamentally different from natural images. This motivates to explore different strategies in this thesis. This study focuses on developing self-supervised learning strategies for two categories of Earth observation applications: static landcover classification and time series crop classification. The following three subsections outline the key challenges addressed in this work.

1.2.1 Self-Supervision for Earth Observation

To achieve the similar effectiveness of self-supervised learning for Earth observation data as seen with natural images like ImageNet (Russakovsky et al. 2015), researchers must overcome Earth observation specific challenges. In contrast to natural images, Earth observation data are derived from surface reflectance measurements across multiple wavelengths, capturing a broad range of physical and spectral characteristics of the land surface. The spectral signature varies systematically across surface materials, and these inherent optical properties are fundamental for distinguishing and classifying different land-cover types.

Earth observation data, obtained by measuring surface reflectance at different wavelengths, encompasses a broader range of physical properties beyond texture, which are crucial in defining land surface features.

Applying existing algorithms directly to remote sensing data is non-trivial. Contrastive self-supervised learning, a state-of-the-art self-supervised learning algorithm, relies on meaningful transformations. In contrastive learning, the supervision signal comes from positive and negative pairs rather than manual annotations. Each data instance is treated as its own class, with positive pairs obtained through stochastic transformations and negative pairs comprising other data instances. The goal is to learn semantic information shared between a data instance and its augmented version by aligning representations of positive pairs while repelling negative pairs. However, most transformations used for natural images are texture-based and cannot be applied to Earth observation data due to the presence of non-visible bands like infrared. Naive application would alter the physical meaning of surface states, such as vegetation health. This limitation necessitates the development of Earth observation-specific approaches that respect the unique properties of satellite imagery.

Research Question 1
How can contrastive self-supervised learning be utilized for land cover applications using remote sensing data? How can the strategy be developed that must tackle the unique challenges of integrating non-visible spectral bands, such as infrared while respecting the physical principles that characterize the land surface?

To address this, atmospheric transformation is proposed in this thesis instead of color-based transformations commonly employed for natural images. Color-based transformations used in natural image processing, such as color jittering and grayscaling, are designed to preserve texture information in natural images. However, when applied to remote sensing images, the atmospheric transformations ensure the physical significance of non-visible bands like near-infrared is not altered, preserving indicators of vegetation health. In contrast, remote sensing images can benefit from domain-specific transformations, particularly atmospheric correction. Atmospheric correction algorithms are widely used in the remote sensing community to obtain atmospherically corrected images by accounting for the scattering and absorption of both emitted and reflected radiation from the atmosphere. In this thesis, these atmospheric correction algorithms are being utilized to develop the atmospheric transformation algorithm that is compatible with all bands, including those beyond the visible spectrum, and can be seamlessly integrated into the contrastive learning framework. The proposed atmospheric transformation approach demonstrates superiority over color-based transformations. By employing atmospheric transformations, one can effectively exploit the vast amount of publicly available remote sensing images without incurring the substantial cost associated with annotating labels manually. This approach enables the utilization of self-supervised contrastive learning techniques on remote sensing data.

1.2.2 Self-Supervised Learning for Crop Classification Tasks

The applications mentioned previously utilize remote sensing data. These applications often process a single image as the input to the model. However, there are numerous applications that require more than just a single timestamp of information, such as landcover change detection, natural resource management, and various agricultural applications including crop classification, yield prediction, and crop growth monitoring. This thesis focuses on developing self-supervised strategies to improve crop classification.

Crop data presents unique challenges for contrastive learning algorithms typically used in remote sensing. Unlike standard satellite images, crop information is often not publicly available and is characterized by distinct temporal patterns reflecting crop development cycles. This temporal aspect is crucial for analysis. Additionally, crop data is associated with field parcels, with each parcel usually containing a single crop type. This parcel-based organization limits the analysis with this confined spatial location, contrasting with typical satellite images. These factors - the temporal nature of crop development, the parcel-based spatial structure, and the variety of crop types at each geographical location - make it difficult to directly apply existing contrastive learning approaches on remote

sensing images to crop classification tasks. Consequently, novel adaptations are necessary to effectively leverage these techniques for crop data analysis.

Research Question 2
How can the contrastive learning algorithm be effectively adapted for crop classification explicitly, considering the challenges posed by the temporal nature of crop data and the spatial dimension being confined to field parcels? Specifically, how to devise meaningful transformations that generate diverse views of the crop data?

The assumption that the boundaries of field parcels are not known poses a significant complication in applying unsupervised algorithms. Consequently, the option of handling the data as an image is discarded. The alternative is to consider the data in a tabular form, i.e., measured reflectance at a particular location. However, finding meaningful transformations for tabular data is non-trivial. One advantageous aspect of remote sensing data is the availability of multiple land cover satellite missions with different configurations and capabilities, observing the same part of the land, although at slightly different times. This allows for the implementation of multi-modal contrastive learning on this tabular data. The overall idea behind using a multi-modal contrastive learning approach is to address the challenge of identifying suitable transformations by leveraging data from different sources. In this work, Sentinel2 and Planetscope are used as two different satellite sources. The proposed multi-modal contrastive learning with these two satellite sources is found to be superior compared to uni-modal contrastive learning employing feature corruption as a means to generate an augmented view of the tabular data. In this approach, the spectral component of both sources is utilized for aligning the representation. The added advantage of the multi-modal approach is that while applying, it does not necessarily require both sources during inference.

1.2.3 Alternate Self-Supervised Techniques for Crop Classification

The multi-modal contrastive learning approach produces an effective pre-trained model for learning representations of crops. However, this approach only exploits the complementary spectral components of the data sources, while this data is also rich in temporal information. This study explores different self-supervised approaches to incorporate both spectral and temporal components in the learning process. State-of-the-art self-supervised techniques exist in the NLP community, which employs modern architectures such as transformers (Vaswani et al. 2017). These methods are instrumental in addressing complex language-related tasks and form the foundation of Large Language Models (LLMs). Since most text data follows a sequential pattern similar to time series, these techniques can be explored in other domains like remote sensing time series data. Exploring such techniques for remote sensing time series data will enable the self-supervised model to exploit both the spectral and temporal components of the satellite data. The domain of self-supervised learning also employs auxiliary losses, where the task is not di-

rectly relevant for end applications, and labels can be generated using heuristic algorithms rather than manual annotation. Combining such domain-related auxiliary tasks and the other self-supervised techniques from natural language processing (NLP) can enhance the learning process.

Research Question 3
How can the self-supervised learning algorithms that effectively exploit both the spectral and temporal components of remote sensing data be developed for crop classification? Specifically, how to adapt techniques from natural language processing to leverage the sequential nature and multiple modes of remote sensing data?

To address the issue of adapting contrastive learning for time-series crop classification, the spectral multi-modal contrastive learning approach is extended to spectro-temporal data, where the data is a time series of measured reflectance at a specific pixel or location on the land surface. Handling temporal data requires the use of transformer models. However, due to the nature of transformers, it is not feasible to use large batch sizes, which can affect the generalization of the self-supervised model. To overcome this challenge, BERT, a popular self-supervised algorithm in the natural language processing NLP community, is proposed in a multi-modal approach. BERT employs masked language modeling for pre-training sequential data like text. With a few adaptations, BERT is integrated into our multi-modal framework. The BERT model addresses the issue of requiring large batch sizes for contrastive learning. Furthermore, the learning process is boosted by defining auxiliary losses, such as predicting proxy tasks like the seasonality of the representation and the cloud cover level of the measured reflectance. These auxiliary losses help the model implicitly learn seasonality and the perturbations caused by clouds, respectively. This approach utilizes the spectro-temporal component in the complementary domain during pre-training while still retaining the advantages of the multi-modal methods, i.e., not necessarily requiring the use of both data sources.

1.3 Contributions

The previous section outlined three research questions addressed in this work. Research question 1 focuses on the challenges of applying contrastive learning techniques to Earth observation data, particularly remote sensing images. The proposed atmospheric transformation method offers a solution to the issues to overcome the difficulties. By providing a solution to the issues of applying contrastive learning to Earth observation data, this method advances the field of self-supervised learning for remote sensing images. The details of this contribution from the work focussing on this issue are elaborated in subsection 1.3.1.

Research question 2 highlights the distinct nature of crop classification compared to general land cover classification. This difference presents unique challenges in identifying

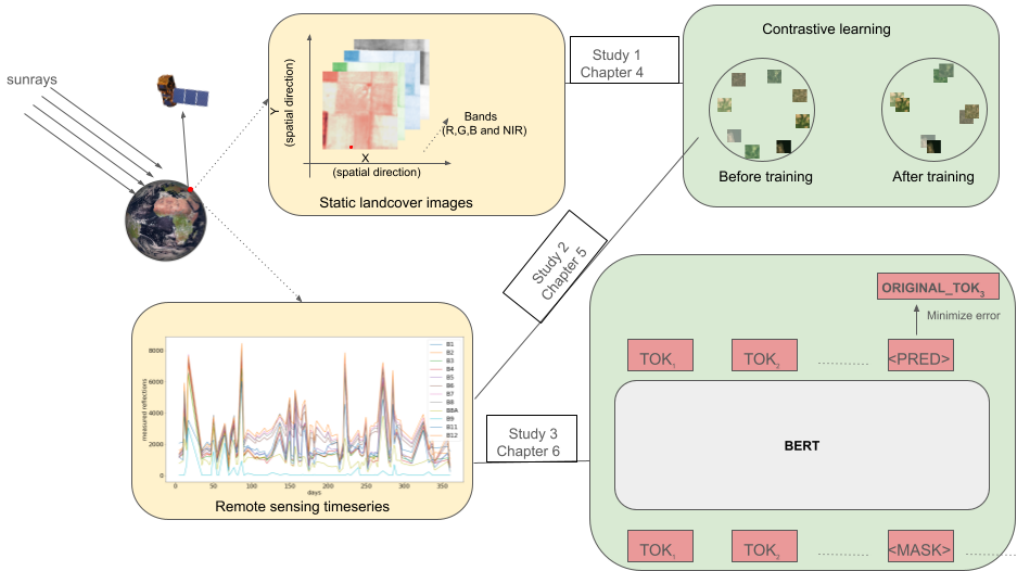


Figure 1.1: **Overview of the Research Contributions**. The data originates from the reflected radiation emitted by the land surface, which is measured and processed by optical remote sensing satellites. These processed satellite data are used for various applications. Contrastive self-supervised learning is employed for static land cover classification is discussed in Chapter 4. Chapter 5 focuses on the application of contrastive learning to time series data, with a specific emphasis on crop classification tasks. Chapter 6 explores the adaptation of the BERT masked language model, a self-supervised technique, for learning representations from remote sensing time series data.

meaningful transformations for tabular data that represent spectral measurements at the pixel level. To address this issue, a novel multi-modal self-supervised strategy for crop data has been developed. The major contributions resulting from this approach are thoroughly discussed in subsection 1.3.2.

The third research question underscores the importance of developing a self-supervised learning strategy for crop classification that can effectively handle both the spectral and temporal aspects of the data during the pre-training phase. In response to this challenge, an innovative BERT bi-modal model has been proposed. This model not only addresses the aforementioned issues but also leverages the advantages of multi-modality. The significant contributions from this part of the work are detailed in subsection 1.3.3.

To provide a comprehensive overview of this research, Figure 1.1 presents a graphical abstract that visually summarizes the key components and contributions of this thesis.

1.3.1 Atmospheric Transformation for Contrastive Learning for Static Landcover Classification

Common transformations have proved beneficial for natural images. Naively extending these transformations to all channels of multi-spectral remote sensing images can distort the physical information contained in bands beyond the visible spectrum. Since the focus is on vegetation analysis, the experiments incorporated the near-infrared (NIR) channel

along with the existing RGB channels. The results showed the benefits of adding an NIR channel for land cover classification. Then to address the challenge of meaningful transformation for remote sensing images, atmospheric transformations, a novel transformation technique is developed based on atmospheric correction algorithms. Chapter 4 thoroughly discusses the methodology and experimental setup, demonstrating the superiority of the proposed atmospheric transformations over conventional color-based transformations for contrastive self-supervised learning on remote sensing data.

1.3.2 Multi-Modal Contrastive Learning for Crop Classification

Applying contrastive learning to crop data, which comprises both spectral and temporal domains, presents significant challenges. Existing data transformation methods are not easily adaptable to this type of data. To address this, the multi-modal contrastive self-supervised learning approach is proposed. Multiple land cover satellite missions focus on the same region with different spatial, spectral, and temporal resolutions. In this work, a novel setup has been designed that leverages data from two satellite sources. Thorough evaluations have been conducted on three different time series networks using the representation from the pre-trained model for a comprehensive analysis. The results demonstrate the superiority of the multi-modal approach over the uni-modal approach employing random feature corruption.

1.3.3 Self-Supervised Learning with Time Series Networks for Crop Classification

The multi-modal approach proposed earlier only utilizes the spectral component of the two satellite sources. In this work, an innovative BERT bi-modal strategy has been introduced that leverages both spectral as well as the temporal component for pre-training of crop classification models. This approach enhances the model's ability to capture temporal complex patterns inhibited by the crops without using manual labels. Furthermore, the two novel auxiliary losses, the seasonal classifier and the cloud prediction loss, have been proposed that substantially improve the self-supervised learning process. The results are validated by conducting extensive experiments comparing the proposed bi-modal BERT method against the multi-modal self-supervised model. These comprehensive evaluations demonstrate the effectiveness of the BERT bi-modal approach in leveraging diverse data sources and temporal information for improved crop classification accuracy.

1.4 Peer-Reviewed Publications

This work produced three papers and one article in conference proceedings related to remote sensing. Following are the bibliographic details of the papers :

1.4.1 Journal Articles

- **Generating Views Using Atmospheric Transformation for Contrastive Self-Supervised Learning on Multi-Spectral Images**
  Ankit Patnala, Scarlet Stadtler, Martin G. Schultz, and Juergen Gall
  IEEE Geoscience and Remote Sensing Letters (Vol 20), 2023
  doi : https://doi.org/10.1109/LGRS.2023.3274493

- **Bi-modal contrastive learning for crop classification using Sentinel2 and Planetscope**
  Ankit Patnala, Scarlet Stadtler, Martin G. Schultz, and Juergen Gall
  Frontiers in Remote Sensing, Sec. Remote Sensing Time Series Analysis (Vol 5), 2024
  doi: https://doi.org/10.3389/frsen.2024.1480101

- **BERT Bi-Modal Self-Supervised Learning for Crop Classification Using Sentinel2 and Planetscope**
  Ankit Patnala, Martin G. Schultz, and Juergen Gall
  Frontiers in Remote Sensing, Sec. Remote Sensing Time Series Analysis (Vol 6), 2025
  doi: https://doi.org/10.3389/frsen.2025.1555887

1.4.2 Conference Presentations

- **Multi-Modal Self-Supervised Learning for Boosting Crop Classification Using Sentinel2 and Planetscope**
  Ankit Patnala, Scarlet Stadtler, Martin G. Schultz, and Juergen Gall
  IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium
  doi : https://doi.org/10.1109/IGARSS52108.2023.10282665

1.5 Structure of Thesis

This thesis explores the application of self-supervised learning in two domains of Earth observation: static land cover classification and crop classification. The focus of this work was on the classification of vegetation types, and more specifically crop types, using multi-spectral remote sensing data.

Chapter 2 discusses the theoretical foundations of both remote sensing and machine learning literature. The chapter briefly covers distributed learning in the context of handling large amounts of data.

Chapter 3 provides a comprehensive overview of different self-supervised techniques. The chapter categorizes these methods into four distinct algorithmic approaches: autoencoders, auxiliary tasks, instance-based discriminative methods, and BERT.

In **Chapter 4**, the paper "Generating Views Using Atmospheric Transformation for Contrastive Self-Supervised Learning for Multi-Spectral Images" is presented. This chapter was structured to cover land cover classification, an introduction to the topic, a section dedicated to datasets, a complete explanation of the proposed atmospheric transformation method, evaluation, results, and conclusions.

Chapter 5 presents the paper "Multi-modal Contrastive Learning for Crop Classification Using Sentinel2 and Planetscope". The chapter revisits the task of crop classification and emphasizes the importance of learning temporal patterns during the self-supervised pre-training stage. Further, it discusses the development of a dataset for conducting spectro-temporal self-supervised learning experiments. The chapter provides a detailed description of the methods, experimental setup, ablation studies, and results, concluding with several discussion points for the spectro-temporal self-supervised learning algorithm.

Chapter 6 presents the paper "BERT Bi-Modal Self-Supervised Learning for Crop Classification Using Sentinel2 and Planetscope". This chapter highlights the importance of learning temporal patterns in the self-supervised pre-training stage. It further discusses how to develop a dataset to conduct spectro-temporal self-supervised learning experiments. It provides a detailed description of methods, experimental setup, ablation studies, and results, and ends with a few discussion points.

Finally, **Chapter 7** conclude this thesis by summarizing the major findings and key takeaways from this work. The chapter addresses major challenges and provides an outlook for future directions.

CHAPTER 2

Preliminaries

This chapter covers the preliminaries required for both remote sensing and machine learning. The first part covers topics related to remote sensing.

It is organized as follows. Subsection 2.1.1 discusses Sentinel2, a satellite mission from ESA. This is followed by Subsection 2.1.2, which covers the different channels or bands of Sentinel2. Section 2.1.3 provides an overview of the conventional approach to land surface analysis using optical remote sensing satellites. The workflow of Google Earth Engine (GEE) for obtaining Sentinel2 images is described in Subsection 2.1.4. Finally, Subsection 2.1.5 introduces Planetscope, a constellation of small satellites operated by Planet Labs, and describes the procedure used to obtain Planetscope imagery

The second part will cover the basics of machine learning. Subsections 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5, and 2.2.6 provide an overview of neurons, loss functions, gradient descent, different types of machine learning, distributed machine learning, and different network architecture used in this work respectively. Finally, the concluding section highlights the intersection of remote sensing and machine learning fields.

Contents

2.1 Introduction to Remote Sensing

In the context of this work, remote sensing refers to the process of detecting and monitoring the physical properties of the Earth's surface by measuring the reflected and emitted radiation from satellites, drones, or aircraft. In remote sensing, sensors capture the reflected or emitted radiations across different wavelengths. These sensors record an electrical signal that is proportional to the received radiation. Each wavelength conveys unique information about the land surface characteristics. Remote sensing finds applications in a wide range of domains, including environmental monitoring, disaster prevention, land cover mapping, crop yield estimation, and others, analyzing this data is crucial for human understanding and decision-making.

This chapter focuses on land cover satellite images from various satellite missions that have been launched to date, for example, Landsat[1] and MODIS (ORNL DAAC 2018) from NASA, Sentinel satellite missions from ESA (ESA 2021), and commercial satellite missions like Planetscope[2]. These satellites typically use scanning techniques such as push-broom (using a line of detectors parallel to the satellite's motion) or side-looking scanners (scanning at an angle to the satellite's path) to capture images. Their wide swath widths enable broad coverage and frequent revisits. Once captured, the imagery is transmitted to ground stations for processing and distributed to end-users. For landcover monitoring, there are two main types of remote sensing images: optical remote sensing images and synthetic aperture radar (SAR) images. Optical remote sensing images are captured by sensors that record the reflected sunlight across various wavelengths of the electromagnetic spectrum. In contrast, SAR images use microwave signals instead of sunlight to monitor the Earth's surface. The sensors in SAR systems record backscattered radar signals to form images. While optical sensors provide information about the surface and cannot penetrate through clouds, vegetation, soil, and other obstacles, SAR can penetrate through these objects. Optical remote sensing is primarily used for applications such as land cover classification, vegetation analysis, and surface monitoring. On the other hand, SAR is used for tasks like terrain mapping, disaster monitoring, and others. While some researchers use SAR for vegetation analysis due to its resilience to cloud cover, however, this study focuses on optical remote sensing data.

2.1.1 Sentinel2

Sentinel2 is an advanced Earth observation satellite mission developed by ESA as a part of the Copernicus program[3]. It consists of a constellation of two satellites, Sentinel-2A and Sentinel-2B, orbiting the Earth in a sun-synchronous polar orbit. Launched in 2015, Sentinel2 has aided research and developments in the field of optical remote sensing applications due to its advanced imaging capabilities, high spatial resolution, and global coverage. The ability to produce images of the entire globe every 5-6 days has made

[1] https://www.usgs.gov/landsat-missions
[2] https://api.planet.com
[3] https://scihub.copernicus.eu/

Sentinel2 a valuable resource for various applications, including environmental monitoring, agriculture, change detection, forestry, urban planning, disaster management, and natural resource management. Since 2017, Sentinel2 has been providing atmospherically corrected and orthorectified images using data fusion techniques, enabling researchers to enhance their analyses using high-quality images. A detailed explanation of the atmospheric correction algorithm is given in Chapter 4.

2.1.2 Multi-Spectral Optical Remote Sensing Images

After providing a brief introduction to the benefits of optical remote sensing earlier, this section will elaborate further on multi-spectral remote sensing images, with a specific focus on Sentinel2. Optical imagery is collected by sensors that record electromagnetic radiation across a wide range of wavelengths, encompassing the visible, near-infrared (NIR), and shortwave infrared (SWIR) regions of the electromagnetic spectrum. Figure 2.1 illustrates the wavelength distribution of all the bands in the Sentinel2 satellite mission. Table 2.1 provides a comprehensive overview of Sentinel2's spectral capabilities, detailing each band's number, spectral channel, wavelength range, and spatial resolution. This table summarizes Sentinel2's wide range of spectral bands, encompassing coastal aerosol detection, visible channels, red-edge and multiple NIR bands, and multiple SWIR bands. Each of these bands provides distinct insights into various aspects of the Earth's surface properties.

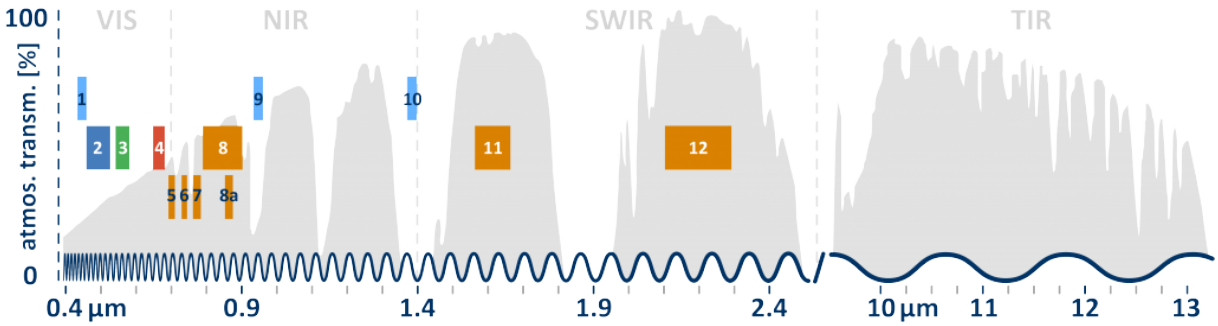


Figure 2.1: **Wavelength distribution of each Sentinel2 bands.** Figure taken from [a].
VIS = Visible bands
NIR = Near infrared bands
SWIR = Shortwave infrared
TIR = Thermal infrared
a= https://blogs.fu-berlin.de/reseda/sentinel-2

2.1.3 Land Surface Analysis from Optical Remote Sensing Satellites

The NIR band is of particular interest for this study due to its capability to capture crucial information about the health, composition, and characteristics of vegetation on the

Table 2.1: **Spectral and spatial resolution of each bands of Sentinel2**.

Band number	Spectral channel	Wavelength [µm]	Spatial resolution [m/px]
1	coastal aerosol	0.429-0.457	60
2	blue	0.451-0.539	10
3	green	0.538-0.585	10
4	red	0.641-0.689	20
5	red edge	0.695-0.715	20
6	red edge	0.731-0.749	20
7	red edge	0.769-0.797	20
8	NIR	0.784-0.900	10
8a	narrow NIR	0.855-0.875	20
9	water vapour	0.935-0.955	60
10	SWIR cirrus	1.365-1.385	60
11	SWIR	1.565-1.655	20
12	SWIR	2.100-2.280	20

Earth's surface. Healthy vegetation strongly reflects NIR light while absorbing red light, owing to the presence of chlorophyll in leaves. The relationship between these spectral bands and plant properties makes the NIR band invaluable for various applications, such as monitoring crops, forests, grasslands, and related tasks like detecting stress, diseases, and changes in growth patterns. The vegetation state of the surface is measured using vegetation indices, which are derived from the relationship between the NIR and red channels. The most widely used vegetation index is the Normalized Difference Vegetation Index (NDVI), calculated as:

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

Equation 2.1 highlights the contrasting reflectance properties of vegetation in the NIR and red regions of the electromagnetic spectrum, enabling the quantification of vegetation health and density.

The NIR channel's ability to capture vegetation characteristics makes it an important element for land cover classification tasks. Different land cover types exhibit distinct NIR reflectance properties - areas with dense vegetation like grasslands and forests strongly reflect NIR light due to the presence of chlorophyll in plants. In contrast, water bodies, drought-affected regions, industrial areas, and urban residential zones have relatively lower NIR reflectance. These contrasting NIR reflectance signatures across various land cover classes serve as a valuable discriminative feature for land cover classification models. The other applications where NIR plays an important role are assessing the extent and severity of burns caused by wildfires by detecting changes in vegetation health, estimating turbidity which is important for the water quality analysis, evaluating the distribution and health of green spaces in urban areas, aiding urban planning and infrastructure analysis,

and also for determining soil moisture content, which is vital for optimizing agricultural irrigation management practices. While these indices are traditionally derived from physical knowledge, modern machine learning models can now leverage the whole information available across all spectral channels for these vegetation related applications.

2.1.4 Google Earth Engine

After discussing Sentinel2, its properties, and the importance of the NIR channel, in this section, the ESA's open data policy is discussed with respect to Sentinel2 data and different ways to acquire Sentinel2 data. The Copernicus program officially provides a RESTful API, named Sentinel Hub API[4], as an interface to access various satellite imagery archives from Sentinel2. This API enables access to raw satellite data, rendered images, statistical analysis, and other functionalities related to Sentinel2 data.

Alternatively, one can use GEE (Gorelick et al. 2017b), a cloud-based platform created by Google for Earth observation data analysis. GEE has downloaded data from several Earth observation sources, including Sentinel2, and hosts them on their servers. By signing up with a Google account, users receive API credentials to access the digital platform and leverage its tools and libraries for geospatial analysis. Users can perform research directly on the GEE platform using its code editor and the built-in visualization tools for interactive maps, time-lapse animations, and more. GEE also provides a Python framework for local analysis. In this work, GEE is preferred due to its ease of use, although there are many other sources available.

The flowchart 2.2 illustrates the step-by-step process to obtain data from GEE:

2.1.5 Planetscope

Besides various national space agencies, there are also private companies conducting similar Earth observation satellite missions. Among these, Planetscope stands out as a constellation of more than 100 small satellites—known as Doves—developed and operated by Planet Labs, California, USA. This "swarm" configuration, with over one hundred satellites in low Earth orbit, allows Planetscope to achieve near-daily global coverage by revisiting the same locations at frequent intervals. The constellation is continuously expanding, with new launches further improving both coverage and revisit rates. Planetscope satellites have been providing data since 2018, and current models achieve a spatial resolution of approximately 3–4 meters, which is particularly advantageous for studies requiring detailed land surface information. These capabilities facilitate time-sensitive monitoring tasks such as tracking agricultural growth cycles, assessing land-cover changes, and supporting rapid response to natural disasters. Planet Labs preprocesses Planetscope imagery and delivers it projected onto the Universal Transverse Mercator (UTM) grid, allowing seamless integration with other remote sensing products such as Sentinel2. In this thesis, the Planetscope dataset was obtained from an existing dataset that provided these satellite readings, though Planet Labs also offers users a dedicated web platform

[4]https://documentation.dataspace.copernicus.eu/APIs/SentinelHub/ApiReference.html

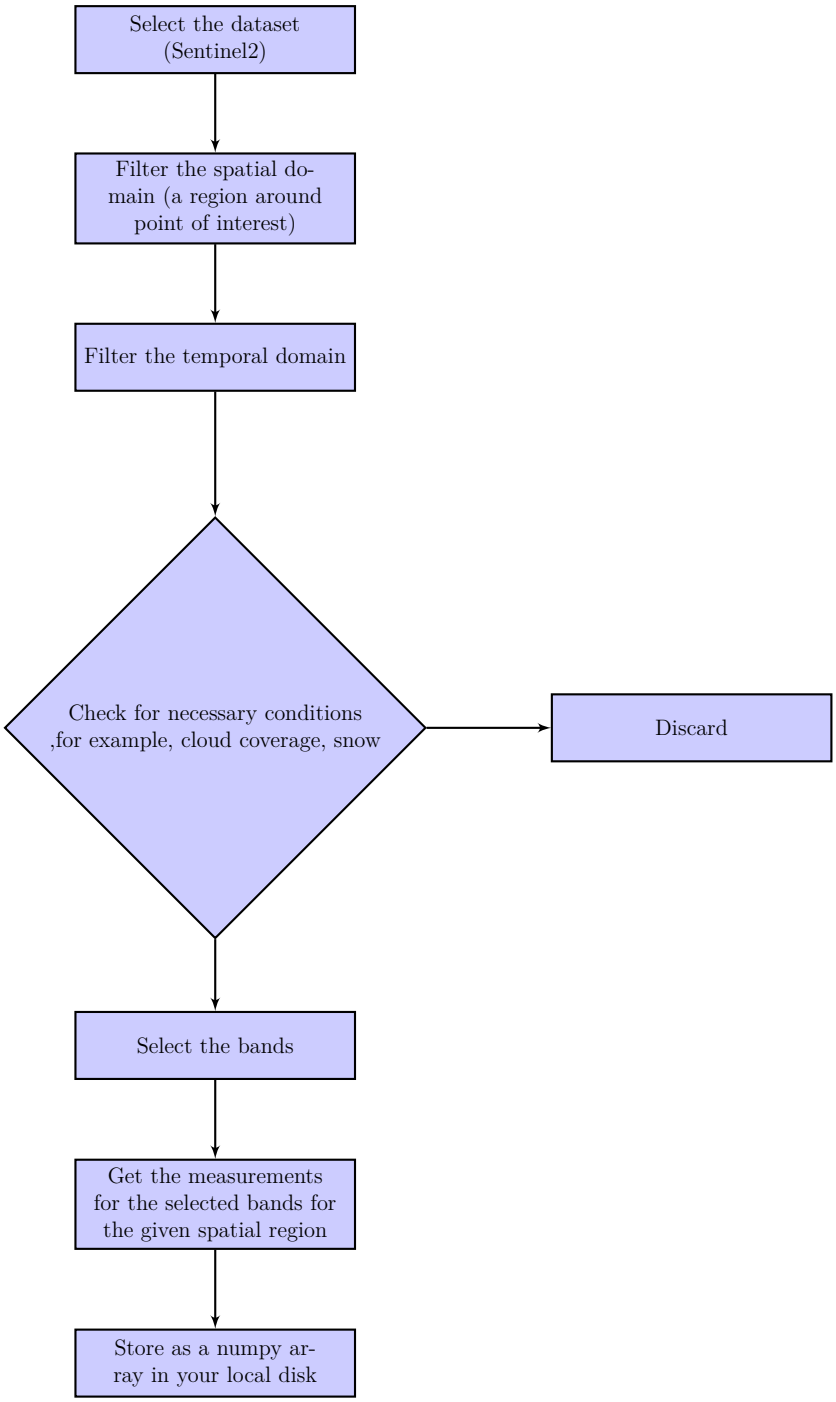


Figure 2.2: GEE workflow.

to search, order, and directly download imagery, following procedures outlined in their official documentation[5]. Access to Planetscope data is commercial, requiring licenses and quota for downloads. However, a downside of Planetscope is that it offers fewer spectral bands compared to some other missions. Furthermore, being a commercial venture, not all users may be able to access its advanced data due to cost considerations.

2.2 General Overview of Machine Learning

In this section, a general overview of the fundamental concepts of machine learning is provided, followed by a discussion on the applications of machine learning in remote sensing for various land cover analyses.

Machine learning has become a ubiquitous term across numerous fields. Traditional models often rely on sequential algorithms or numerical solutions based on mathematical formulations such as partial differential equations (PDEs) (Samuel 1959). These algorithms use explicit subroutines to produce desired outputs. In contrast, machine learning tunes the black box using a set of input-output pairs. To train a machine learning model, one must define a model architecture with variable parameters and a set of user-defined hyperparameters.

The Perceptron model (Rosenblatt 1958) was the first basic model developed in this field. Often compared to a biological model of a single neuron, the Perceptron is a single-layer network that processes multiple inputs to produce a single output, either a classification score or a regression value. The model's weights represent linear weights in a linear hyperspace. These weights are parameterized and optimized iteratively by minimizing a loss function, which penalizes predictions that deviate from the actual output. Since the Perceptron, ideas, and strategies have advanced significantly beyond its simple linear mapping capabilities. Most neural networks in use today comprise multiple layers to capture complex relationships. Additionally, the use of activation functions enables non-linear mapping from input to output, a capability the original Perceptron lacked. Figure 2.3 illustrates the difference in complexity between the Perceptron and a multilayer perception (MLP), a type of neural network. The architectures used in this work is discussed in Chapters 4, 5, and 6. These more advanced models build upon the foundational concepts introduced by the Perceptron, allowing for more sophisticated analysis and prediction in remote sensing applications.

2.2.1 Neuron

The neuron is the fundamental building block of neural networks. As illustrated in the right panel of Figure 2.3, each neuron is connected to other neurons in the previous layer receiving inputs from them, and is also wired to neurons in the subsequent layer, propagating its output. This connectivity pattern indicates that a neuron takes inputs from the preceding layer, applies a transformation within itself, and then passes the result

[5]https://docs.planet.com/platform/get-started/access-data/order-imagery/

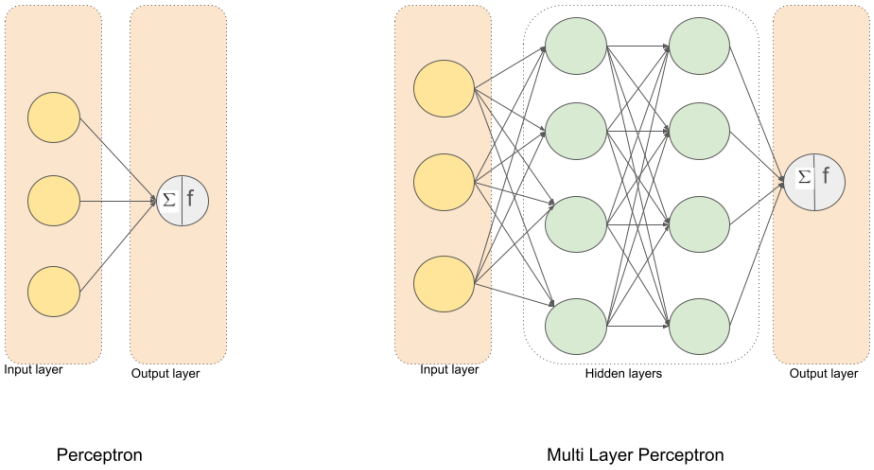


Figure 2.3: Difference in the complexity of Perceptron and Neural Network. "f" in the figure refers to any activation function (Figure 2.4)

to the next layer. Equation (2.2) gives the mathematical formulation of the transformation inside the neuron.

$$o_j^{(l)} = b_j^{(l)} + \sum_{i=1}^{K} w_{ij}^{(l)} a_i^{(l-1)} \tag{2.2a}$$

$$a_i^{(l-1)} = f(o_j^{(l-1)}) \tag{2.2b}$$

Where $o_j^{(l)}$ denotes the output of the j^{th} neuron in the l^{th} layer, $b_j^{(l)}$ is the bias weight for that neuron, $w_{ij}^{(l)}$ represents the weight associated with the connection from the i^{th} neuron in the previous layer to the j^{th} neuron in the current layer, and $a_i^{(l-1)}$ is the output of the i^{th} neuron in the previous layer passed through an activation function f as shown in the Equation (5.1c). The activation functions employed in this thesis include sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU), and leaky ReLU. These non-linear activation functions allow neural networks to learn and model complex, non-linear relationships in data. Without them, the network would only be capable of learning linear transformations. This further allows the stacking of multiple layers. Figure 2.4 presents plots illustrating the behavior of each of these activation functions.

The equations for the various activation functions employed in this thesis are given by Equation (2.3).

$$sigmoid(x) = \frac{\exp(x)}{1+\exp(x)} \tag{2.3a}$$

$$tanh(x) = \frac{\exp(x)-1}{1+\exp(x)} \tag{2.3b}$$

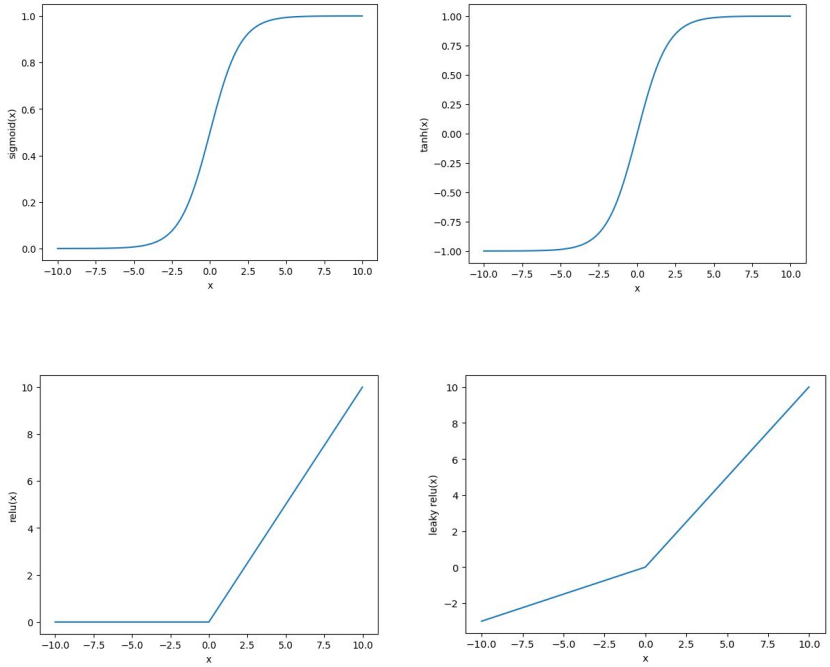


Figure 2.4: Different activation functions.

$$relu(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.3c}$$

$$leakyrelu(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{otherwise with } \alpha << 1.0 \end{cases} \tag{2.3d}$$

In addition to the activation functions discussed earlier, the softmax function is also used in this work for multi-class classification problems. The output layer contains neurons equal to the number of classes in the problem. The softmax activation aims to obtain a probability distribution over all the classes. Equation (2.4) formulates the softmax function mathematically:

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(xj)} \tag{2.4}$$

Where x_i represents the input to the i^{th} output neuron, and N is the total number of output neurons (classes). The softmax function normalizes the inputs such that the outputs sum to 1, effectively providing a probability distribution over the classes. Figure 2.5 provides a visual illustration of the softmax activation function. It depicts how the raw outputs from the previous layer are transformed into a probability distribution by the softmax, enabling the network to make probabilistic predictions for multi-class classification tasks.

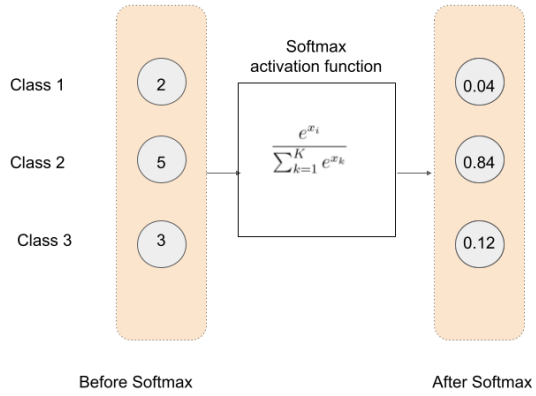


Figure 2.5: Softmax activation function.

2.2.2 Loss Functions

Loss functions, also known as objective functions or cost functions, serve the purpose of estimating the error that a model makes with its current weights. These functions determine how to update the network's weights in each iteration, with the goal of gradually reducing the error over successive iterations. The weight updating process is carried out using the gradient descent method, which will be discussed in the next section. The general loss functions employed in this thesis are MSE, binary cross-entropy, and cross-entropy. It is important to note that for binary cross-entropy, the activation function used in the final output layer is the sigmoid function, which predicts a probability between 0 and 1. On the other hand, for cross-entropy, the softmax activation function is used, ensuring that the sum of all probabilities is 1. The mathematical formulations for MSE, binary cross-entropy, and cross-entropy are given by Equations (2.5a), (2.5b) and (2.5c), respectively.

$$l_{mse} = \frac{1}{N}\sum_{n=1}^{N}(f_\theta(x_n) - y_n)^2 \tag{2.5a}$$

$$l_{bce} = \frac{1}{N}\sum_{n=1}^{N}(y_n log(f_\theta(x_n)) + (1-y_n)(1-log(f_\theta(x_n)))) \tag{2.5b}$$

$$l_{ce} = \frac{1}{N}\sum_{n=1}^{N}(\sum_{c=1}^{C} y_{nc} log(f_\theta(x_{nc}))) \tag{2.5c}$$

In these Equations (2.5), f_θ denotes the neural network with parameters denoted by θ. x_n represents the data sample, and y_n is the corresponding ground truth value for that data sample.

2.2.3 Gradient Descent

Gradient descent is an optimization algorithm that is used to update the weights in the machine learning models. The loss function, as discussed earlier, estimates the error in the current weights. Since the loss is a scalar-valued vector function, its derivative yields a gradient vector. The weights are then updated along the negative gradient direction using a small step size, as shown in Equation (2.6).

$$\theta_{i+1} = \theta_i - \eta \nabla \mathscr{L}(\theta_\rangle) \tag{2.6}$$

This step size is called the learning rate, denoted by η in the equation. The typical learning rate is $1e-2$ to $1e-3$. The terms θ_i and θ_{i+1} represent the old and new weight values, respectively.

This iterative approach adjusts the weights in small increments, gradually converging towards an optimal solution. A larger learning rate may cause divergence and yield suboptimal weights, while a smaller learning rate will require more iterations to converge. In complex and deep networks, a learning rate scheduler is often used to gradually decrease the learning rate as the number of steps increases. The gradient is typically calculated using the backpropagation algorithm (Rumelhart, Geoffrey E Hinton, and Williams 1986), where the derivatives are computed recursively layer by layer, starting from the final layer and propagating back to the input layer. While conventional gradient descent can be slow to converge, momentum-based optimizers like Adam (Kingma and Ba 2015) use the momentum of previous gradients to determine the new weight updates, potentially accelerating convergence.

2.2.4 Classification of Machine Learning

Supervised learning and unsupervised learning are two primary types of machine learning. Supervised learning involves training models using labeled data, exemplified by tasks like the ImageNet image classification (Russakovsky et al. 2015). In contrast, unsupervised learning operates without explicit labels, relying solely on the inherent structure of the data. This section only focuses on supervised and unsupervised learning to provide context for self-supervised learning. Supervised learning comprises various tasks, including multi-class classification, binary classification, and regression. This branch of machine learning has evolved to incorporate diverse architectures such as neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs). Unsupervised learning comprises algorithms like K-means clustering (MacQueen n.d.), Expectation-Maximization (Dempster, Laird, and Rubin 1977), K-Nearest Neighbors (Cover and Hart 1967), and DBSCAN (Ester et al. 1996). However, these traditional methods often face challenges when applied to high-dimensional data such as images or text. Self-supervised learning emerges as a distinct paradigm that, like unsupervised learning, does not require human-annotated data. However, it differs from both supervised and unsupervised approaches. Instead of using pre-defined labels or completely unlabeled data, self-supervised learning generates labels through heuristic algorithms during the learning process. This

approach bridges the gap between supervised and unsupervised methods, offering a unique solution to the challenges of data labeling and high-dimensional data analysis. This ability of self-supervised learning overcomes the limitation of both supervised and unsupervised learning. In practice, it faces on issue how to train such huge amount of data effeciently. Thus, distributed machine learning techniques progresses the field by facilitating the self-supervised learning algorithm to efficiently train on such large scale of data. The next subsection explains different types of distributed machine learning techniques.

2.2.5 Distributed Machine Learning

The increasing use of massive datasets, containing millions to billions of samples, has highlighted memory constraints that limit batch sizes and extend training times. Simultaneously, the trend towards extremely large machine learning models often exceeds available memory capacity, necessitating model distribution across multiple nodes. This approach, known as distributed machine learning, encompasses two main strategies: data parallelism and model parallelism. Data parallelism replicates the entire model across multiple computational nodes while distributing data shards, effectively increasing batch size. Conversely, model parallelism divides large models into chunks, distributing these across multiple nodes. In this approach, data batches flow sequentially through each model chunk, with outputs transferred between devices. Data parallelism further subdivided into two types: standard data parallelism and data-distributed parallelism, each optimized for different applications. A novel approach, Fully Sharded Data Parallelism (FSDP) (Y. Zhao et al. 2023), combines model parameters and optimizer sharding to achieve data parallelism benefits in a model-sharded setting with a slight overhead. Figure 2.6 illustrates the distinction between data and model parallelism (S. Li et al. 2020a).

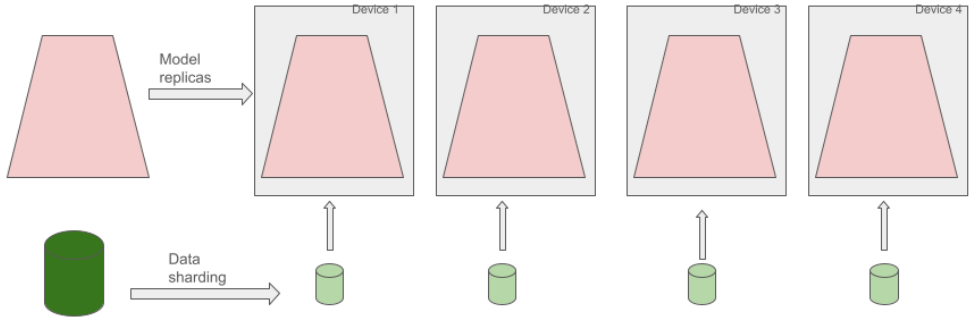
2.2.6 Network Architecture

2.2.6.1 Resnets

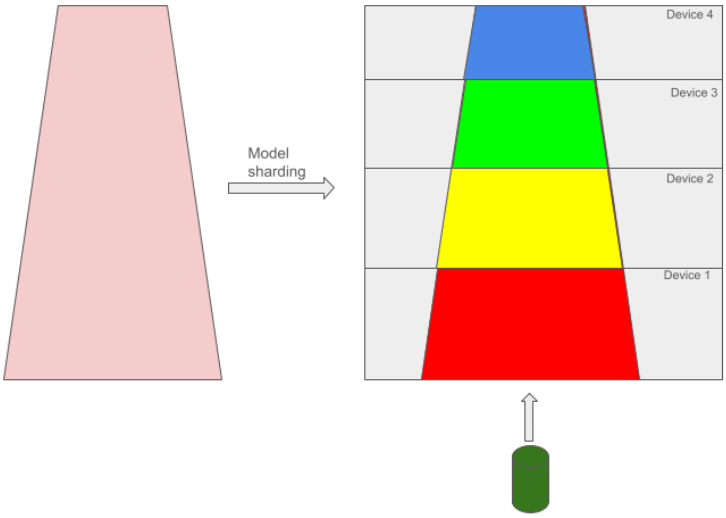
Resnets (He, X. Zhang, et al. 2015) employ a skip connection mechanism to conventional CNNs. Prior to Resnets in the category of convolution networks, LeNet (Lecun et al. 1998), and Alexnet (Krizhevsky, Sutskever, and Geoffrey E. Hinton 2012) were developed but these networks did not facilitate deeper layers. The skip connection connects the output of the convolution layer to its input with a parallel connection. This skip connection enables the deeper models to overcome the issue of vanishing and exploding gradients. The skip connection mechanism led to the rise of deeper Resnets with layers as much as 151. In our work, we used two variants of Resnets i.e., Resnet18 and Resnet50. The layer by layer connection is shown in Figure 2.7.

2.2.6.2 LSTM

LSTM, which stands for Long Short-Term Memory, is an autoregressive model introduced by (Hochreiter and Schmidhuber 1997). It was created as an improved variant



Data Parallelism



Model Parallelism

Figure 2.6: Data Parallelism and Model Parallelism.

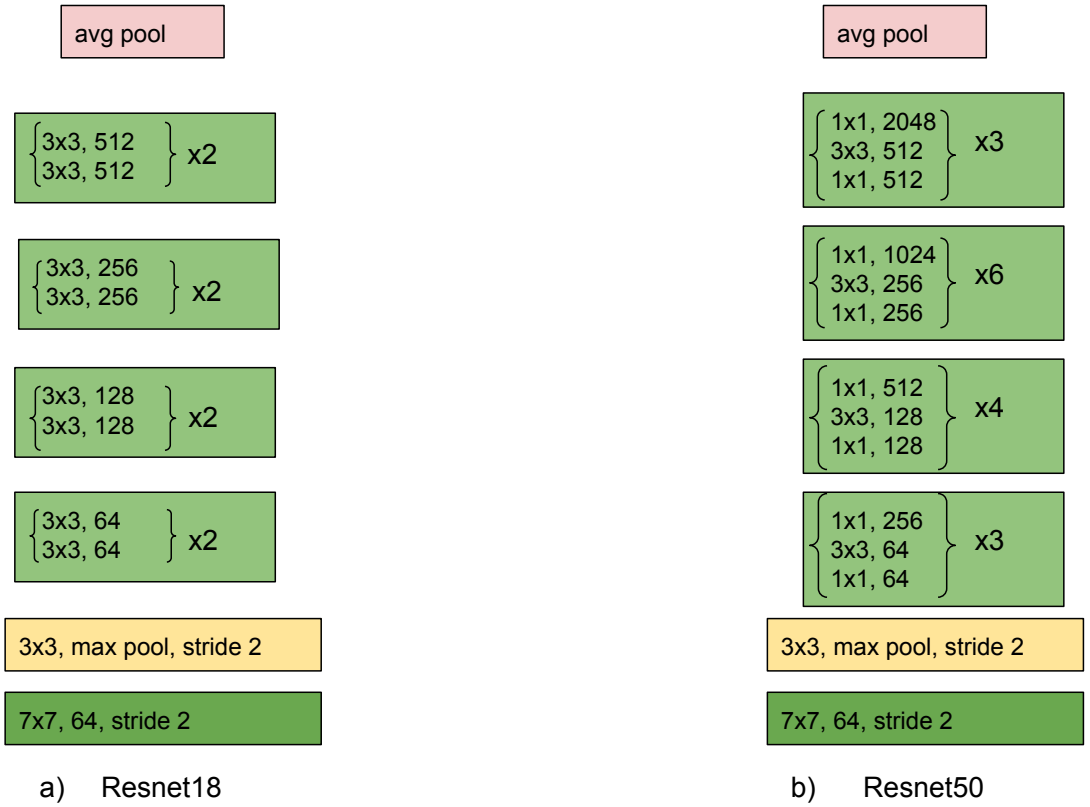


Figure 2.7: **Layer by layer connection of Resnet18 and Resnet50.** All the green boxes represent the convolution layer. Each cell is a residual block whose output is added to the output of the next residual block using a skip connection. Image is passed from bottom to top i.e. the first layer is 64 filters of 7 $\times$7 kernel size with stride 2 and the last layer is the average pooling.

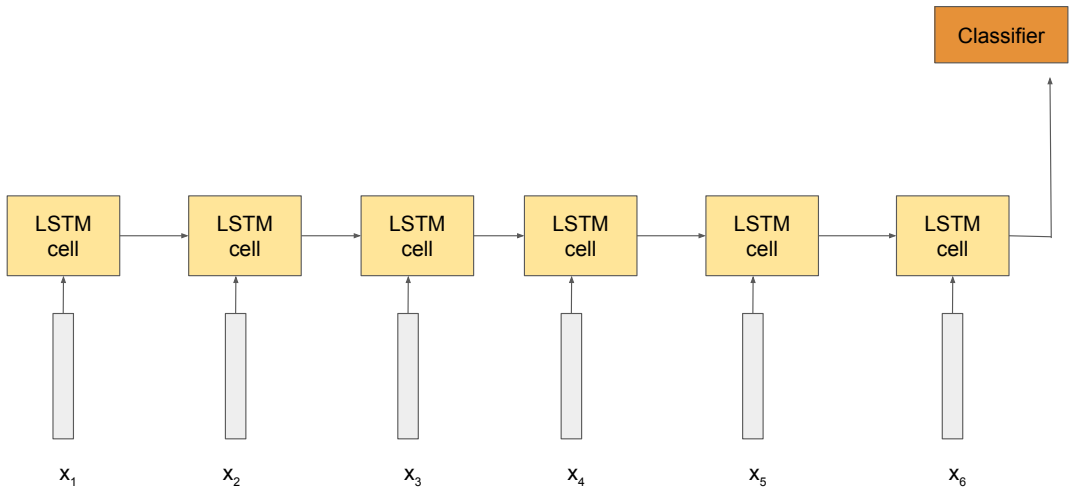


Figure 2.8: An LSTM network.

of the Recurrent Neural Network (RNN). Compared to standard RNNs, LSTM demonstrates superior capability in capturing long-term dependencies within sequences. The architecture of all recurrent networks, including LSTM, consists of a chain of repeating modules, as illustrated in Figure 2.8. Each module processes two inputs: the current word representation and a historical state representation that encapsulates information from all previously processed words. An LSTM cell uses this information to predict the subsequent word in the sequence. These LSTM cells maintain consistent weights throughout the network. The key advantage of LSTM over vanilla RNNs lies in its ability to retain relevant information over extended sequences, making it particularly effective for tasks involving long-range dependencies in data.

The structure of an LSTM cell incorporates several gates, as illustrated in Figure 2.9. These gates enable the LSTM to modify, augment, or eliminate information within the cell state. The input gate, utilizing a sigmoid function, regulates the extent to which new input is incorporated into the current state. Meanwhile, the output gate governs how the memory cell impacts the present time step. The mathematical operations occurring within each LSTM cell are represented by the set of equations shown in Equation (2.7).

$$\begin{aligned} F_t &= \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \\ I_t &= \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \\ O_t &= \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \\ C_t^{'} &= tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \\ C_t &= F_t \cdot C_{t-1} + I_t \cdot C_t^{'} \end{aligned} \tag{2.7}$$

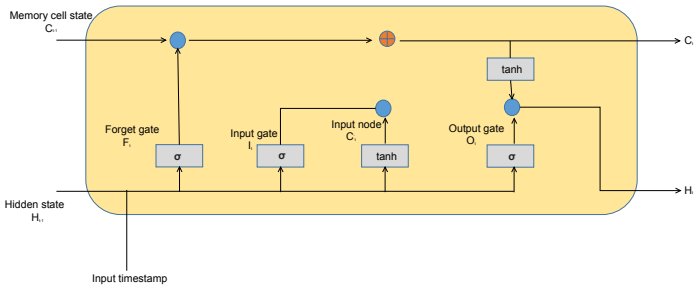


Figure 2.9: A LSTM cell

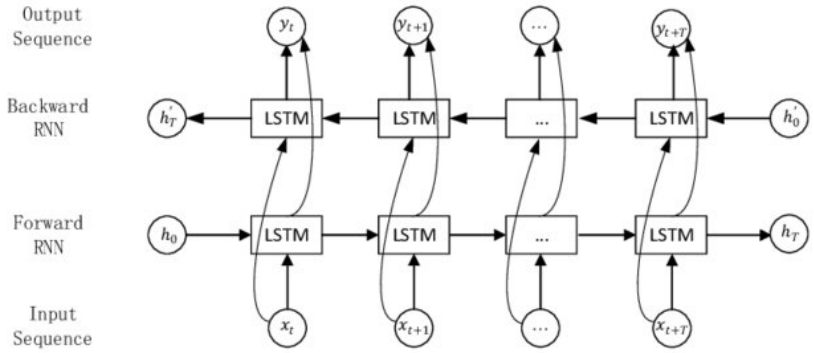


Figure 2.10: Bi-LSTM neural network structure (Xiang, Jinyong et al. 2020)

Despite their improved ability to capture long-term dependencies, LSTM cells still face challenges related to Backpropagation Through Time (BPTT). BPTT is the algorithm used to train recurrent neural networks, including LSTMs, by unrolling the network through time and applying the chain rule of derivatives. In sequential models like LSTMs, each word's representation depends on all preceding words. This sequential nature, combined with the shared weights across time steps, leads to the application of BPTT. To address these limitations, researchers developed the bi-directional LSTM architecture (Cornegruta et al. 2016). This approach processes sequences in parallel from both left-to-right and right-to-left directions, as illustrated in Figure 2.10. By considering both past and future context simultaneously, bi-directional LSTMs can mitigate the vanishing or exploding gradient problems associated with early words in the sequence. This bidirectional processing allows the model to capture more comprehensive contextual information, enhancing its ability to learn from long sequences and improving overall performance on various natural language processing tasks.

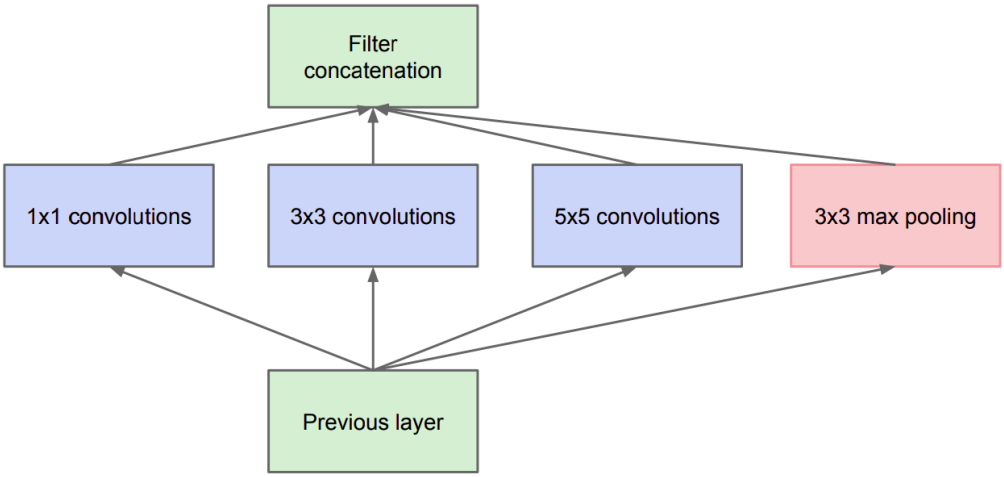


Figure 2.11: An inception module (Szegedy et al. 2014)

2.2.6.3 Inception Network

The inception network, introduced by Google (Szegedy et al. 2014), emerged as an innovative alternative to traditional convolutional networks. At the core of these networks are inception modules, which form their distinctive architecture. Figure 2.11 illustrates the structure of an inception module. In an inception module, input data is processed concurrently through three parallel convolutional layers, each with a different kernel size. This design aims to capture object features at various scales by employing both larger and smaller kernels simultaneously at the same network depth. A complete inception network is constructed by stacking multiple inception modules.

The inception-time network (Fawaz et al. 2019) adapts this concept for time series data analysis. It replaces the 2D convolutional layers of the original inception network with 1D convolutional layers. By utilizing three different kernel sizes, the network gains the ability to learn filters that capture both global and local patterns in the time series data. Figure 2.12 shows the overall architecture of the inception-time network.

2.2.6.4 Transformers

The transformer model, introduced by (Vaswani et al. 2017), was initially developed to address language translation challenges. A key component of these models is the attention mechanism, which operates as follows: Consider the sentence "Tom reads books". Each word is tokenized and represented by a continuous d-dimensional vector. For instance, "Tom" is represented as a vector in $\mathbb{R}^d$. The attention mechanism relies on three crucial elements: queries, keys, and values. The initial vector representation is transformed through different linear layers to obtain these elements for each word, as illustrated in Figure 2.13.

As depicted in Figure 2.13, the query of each word interacts with the keys of all words, including itself. This interaction computes the cosine similarity between words, quantify-

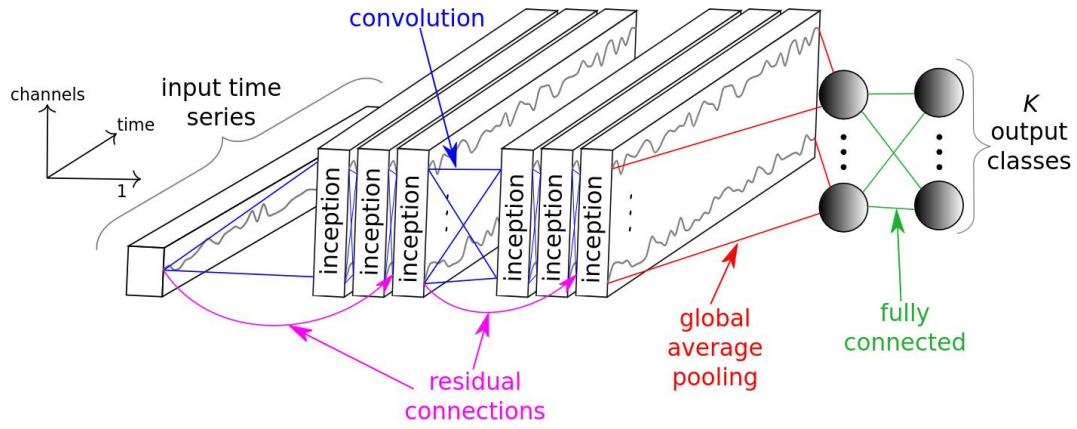


Figure 2.12: Inception time network (Fawaz et al. 2019).

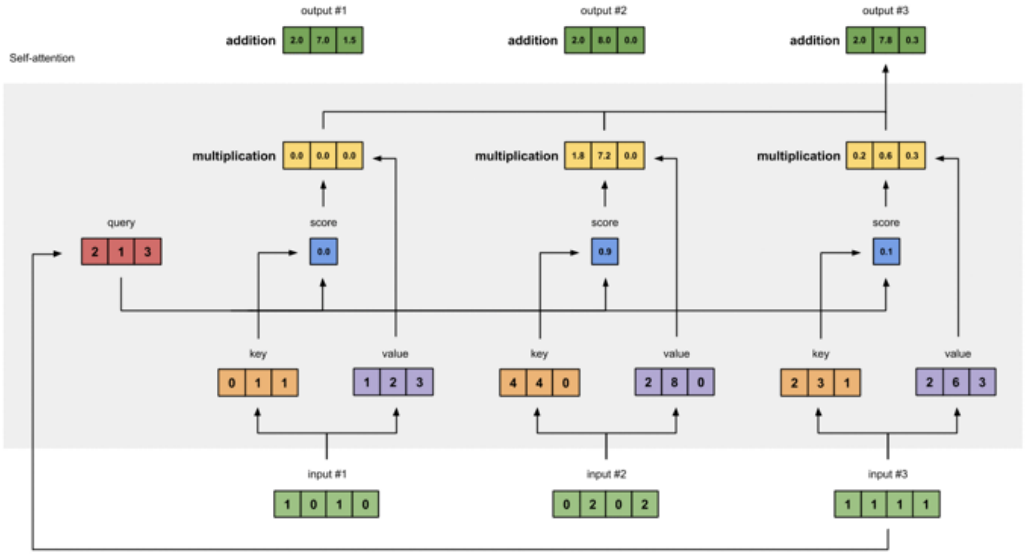


Figure 2.13: Schematic description of attention mechanism (Yan 2022).

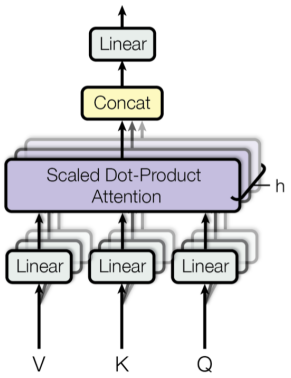


Figure 2.14: Multi-head self-attention (Vaswani et al. 2017).

ing their relatedness. The resulting cosine distances are then processed through a softmax activation function, yielding a probability distribution or weighted importance. The output for each word from an attention layer is obtained by multiplying these weights with the corresponding value vectors.

Transformer architectures typically employ multiple attention heads operating in parallel within each attention block. The features from these heads are concatenated and passed through a linear layer followed by a non-linear activation function, as shown in Figure 2.14

Our focus has been on classification tasks, so we utilized only the encoder component of the original transformer network. The following explores position-encoded transformers used as a base model for crop classification.

2.2.6.5 Position Encoded Transformer

We implemented our position encoded transformer on Sentinel2 i.e. time series data containing the measured reflection value of a pixel over a year. The attention mechanism is position invariant and position i.e. the sequence of events happening plays an important role, hence in an initial layer before passing it to the first attention block, the sinus and cosinus embedding equivalent to the shape of input are added. The sinus and cosinus embeddings are shown in the Equation (2.8).

$$\begin{aligned} PE_{(t,2i)} &= sin(t/10000^{2i/d_{model}}) \\ PE_{(t,2i+1)} &= cos(t/10000^{2i/d_{model}}) \end{aligned} \tag{2.8}$$

The output of the final attention block which contains embedding for each output is passed through a max pooling. This max pooled layer is fed to a linear layer predicting logits for a standard classification task. Figure 2.15 shows the overall architecture of the model we used for our classification task.

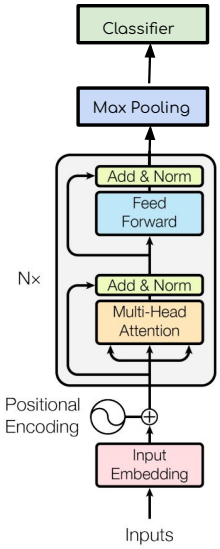


Figure 2.15: Overall architecture of the position encoded transformer network.
Figure adapted from (Vaswani et al. 2017)

2.3 Machine Learning for Earth Observation Applications

Machine learning techniques, particularly CNNs, have gained significant traction in remote sensing and Earth onservation applications (Bruzzone and B. Demir 2014; Hong et al. 2021; Diakogiannis et al. 2019). CNN models have the capability to autonomously extract significant features from remote sensing imagery, thereby superseding the traditional methodology that depends on domain-specific spectral indices for analysis. One of the primary applications of machine learning in this domain has been land use and land cover classification (Y. Chen et al. 2014). CNNs and recurrent neural networks (RNNs) have demonstrated impressive performance on hyper-spectral, multi-spectral, and SAR data for pixel-wise or object-based land use and land cover mapping (Hu et al. 2015; Makantasis et al. 2015; Mou, Ghamisi, and Zhu 2017). Object detection from aerial/satellite imagery is another area where machine learning models like Faster R-CNN, YOLO, and SSD have achieved promising results for detecting objects such as vehicles, buildings, and roads (Z. Zhao et al. 2018). Scene classification, involving categorization of the semantic content (urban, forest, agriculture, etc.), has also benefited from deep CNNs (Cheng and Han 2016). Deep learning has facilitated data fusion tasks like combining high spatial resolution panchromatic and low resolution multi-spectral imagery through pansharpening techniques (Huang et al. 2015; Q. Yuan et al. 2017; Wei et al. 2017). These models have outperformed traditional pansharpening methods. For image segmentation tasks (semantic and instance), deep models like fully convolutional networks (FCNs) (Shelhamer, Long, and Darrell 2016), SegNet (Badrinarayanan, Kendall, and Cipolla 2015), and U-Net have enabled precise delineation of object/region boundaries (Audebert, Saux, and

Lefèvre 2017). Some other emerging applications include time series analysis (Cai et al. 2018), precipitation data retrieval (Tao et al. 2016), and integration of machine learning with traditional object-based image analysis methods for urban land use mapping (Tong et al. 2018). While most studies focused on high-resolution imagery, machine learning models have been applied across different spatial resolutions, with urban areas and vegetated regions being the most commonly studied (L. Ma et al. 2019). Across various tasks, machine learning models achieved high median accuracies, with scene classification yielding around 95%, object detection around 92%, and land use and land cover classification around 90% (L. Ma et al. 2019). Current research efforts are focused on algorithmic advancements (He, X. Zhang, et al. 2016), addressing issues like lack of large annotated datasets, class imbalance, computational requirements, and integrating domain knowledge with data-driven machine learning approaches (L. Ma et al. 2019). The success of machine learning in Earth observation applications can be attributed to the powerful feature learning capabilities of deep neural networks, which distinguish them from conventional algorithms for remote sensing image analysis.

CHAPTER 3

Related Works

This chapter provides a comprehensive exploration of self-supervised learning techniques. As previously discussed, self-supervised learning serves as a crucial link between supervised and unsupervised learning paradigms, offering an unique approach to machine learning challenges encountered in both of these paradigms. To understand the power of self-supervised learning, it is important to know the concept of transfer learning. Transfer learning was originally developed to improve the performance of supervised models (Zhuang et al. 2019). Transfer learning is a technique that allows knowledge gained from solving one problem to be applied to a different but related problem. It has been instrumental in enhancing the performance of models trained on limited labeled data by leveraging knowledge from models trained on large datasets in a different domain. Transfer learning involves storing knowledge from a previous training phase, typically performed on a large labeled dataset. In many domains, abundant unlabeled data is available, and the idea behind self-supervised learning is to leverage this data to learn an initial state or representation.

In self-supervised learning, the training process without labels is called pre-training, and the task is referred to as a pretext task. The pretext task is a proxy task that has no real-world application but serves as a means to learn useful representations from the data. One such pretext task is Rotnet (Gidaris, Singh, and Komodakis 2018) where the image is rotated in one of four orientations (0°,90°,180°,270°) and the model is trained to predict the rotation angle as a four-class classification problem. These pretext tasks allow the model to learn from a large amount of available unlabeled data. Once pre-trained, the knowledge gained by the model can be transferred via transfer learning to learn a real-world task where only a limited amount of annotated data is available. The task for which the weights of the pre-trained models are used is called the downstream task. The performance of the model on these downstream tasks determines the quality of the representations learned during the self-supervised pre-training phase.

Contents

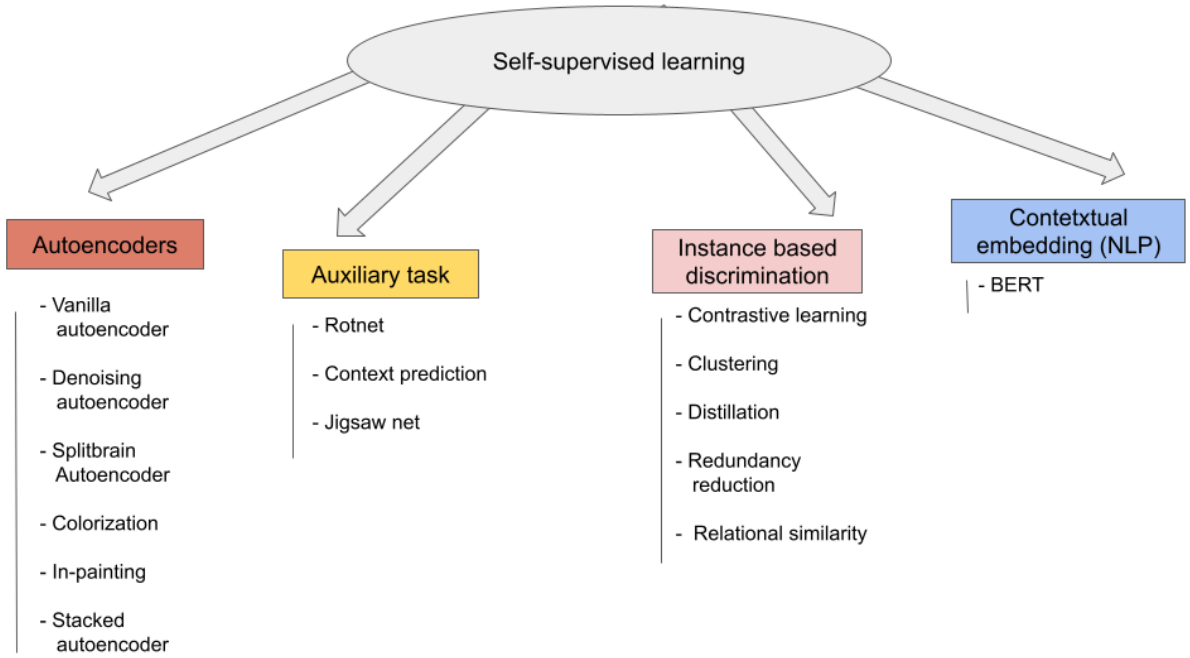


Figure 3.1: Taxonomy of self-supervised learning.

The field of self-supervised learning has witnessed several advancements. This chapter consists of 4 sections describing each type of self-supervised learning. The first section briefly highlights the role of autoencoders, which were among the initial self-supervised approaches. The second section covers the next generation of self-supervised learning methods that involves learning auxiliary tasks. The third section details pretext tasks based on instance discrimination loss. This instance discrimination-based self-supervised learning approach is primarily used in this work. The fourth section discusses BERT, an training technique developed from the NLP community, which is used for encoding word into a vector representation. Figure 3.1 presents a taxonomy of self-supervised learning methods. The last section briefly highlights the multi-modal contrastive learning, an extended variant of contrastive loss to a multi-modal setup.

3.1 Autoencoders

Autoencoders are a type of neural network designed to learn compact representations of input data in an unsupervised manner (Goodfellow, Bengio, and Courville 2016). The core idea behind autoencoders is to first encode or compress the high-dimensional input data into a lower-dimensional latent space through an encoder network. This compressed representation is then passed through a decoder network, which attempts to reconstruct the original input data from the encoded representation. In autoencoders, both the en-

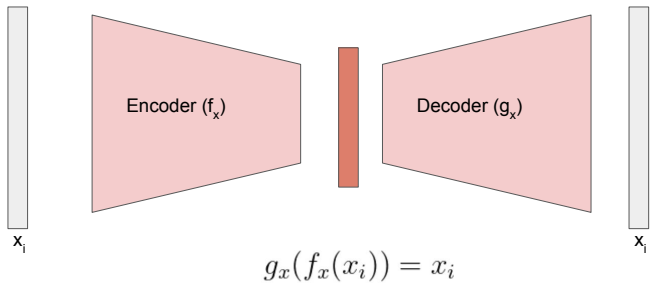


Figure 3.2: **General structure of autoencoders**. The encoders could be any network i.e. MLP, convolutional, or LSTM depending on the data. Here, the objective is to first compress the data and from the compressed data, the decoder reconstructs it.

coder and decoder are trained together to minimize the reconstruction error between the input and the reconstructed output, thereby learning an efficient encoding in the latent space that captures the most salient features of the data. Various neural network architectures can be employed as autoencoders, including standard feedforward networks (MLP) (Goodfellow, Bengio, and Courville 2016), CNN (Masci et al. 2011) suitable for image data, and recurrent networks like LSTMs for sequential data (Sagheer and Kotb 2019). Figure 3.2 illustrates the basic schematic of an autoencoder, consisting of an encoder and a decoder component.

As illustrated in Figure 3.2, autoencoders do not require annotated data, as the model's output is the reconstructed input data itself from a lower-dimensional latent space, similar to principal component analysis (PCA) (F.R.S. 1901). To ensure that the lower dimensional representation is meaningful, a decoder network is employed, to reconstruct the original data. The motivation behind learning a lower-dimensional space is to enable the use of conventional unsupervised algorithms, such as K-means clustering, on high-dimensional data such as images. Due to this compression capability, autoencoders are often referred to as non-linear PCA. The application of autoencoders for clustering land cover data in lower dimensions faces significant challenges. Firstly, autoencoders inherently underperform (Shwartz-Ziv and LeCun 2023), and secondly, the nature of remote sensing images differs substantially from that of natural images, as discussed in the previous chapter, makes it difficult to reconstruct. These complications have hindered the effective use of autoencoders for self-supervised learning on remote sensing images. Various autoencoder variants have been developed to enhance the performance of vanilla autoencoders, including denoising autoencoders (Vincent et al. 2010), split-brain autoencoders (R. Zhang, P. Isola, and Efros 2016b), colorization autoencoders (R. Zhang, P. Isola, and Efros 2016a), and in-painting autoencoders (Pathak et al. 2016). Denoising autoencoders are designed to take noisy version of the sample and reconstruct the original sample, thereby improving the model's robustness to noise. In the case of split-brain

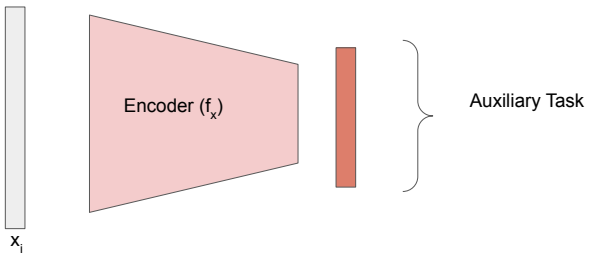


Figure 3.3: The general structure of all self-supervised learning approaches trained using auxiliary tasks.

autoencoders, an image is diagonally divided, and the model learns to reconstruct one half from the other. Colorization autoencoders take grayscale images as input and generate a possible color version, while inpainting autoencoders learn to fill in the randomly removed patch from an image. A significant challenge with autoencoders is the difficulty in training deeper layers, as vital information may be lost during the encoding process, potentially limiting the model's ability to accurately reconstruct inputs. To address this issue, stacked autoencoders (Vincent et al. 2010) were developed, which train each layer iteratively before fine-tuning the entire model. Despite all these advancements, autoencoders still lag behind alternative algorithms in some applications (Shwartz-Ziv and LeCun 2023).

3.2 Auxiliary Tasks

The second approach to self-supervised learning involves solving auxiliary tasks. Here, explanation of such methods are discussed in the context of computer vision. The core idea behind this type of self-supervised learning is to train the model without relying on a decoder to reconstruct the original data, as illustrated in Figure 3.3.

One such auxiliary task is RotNet (Gidaris, Singh, and Komodakis 2018), which was mentioned earlier. In this approach, the pretext task of predicting the rotation angle of an image does not constitute a meaningful application, but it does not require manual annotation. This allows for utilizing a large number of images to obtain a pre-trained model. Another widely-used algorithm in this category involves learning context prediction (Doersch, Gupta, and Efros 2016). In this task, the image is divided into nine (3×3) patches, and randomly selected patch, along with the central patch, is given as an input the model and model learns to identify one of the eight relative positions of the randomly selected patch with respect to the central patch. An extension of the context prediction task is solving jigsaw puzzles (Noroozi and Favaro 2017) created from the image. In this approach, the authors proposed an algorithm in which the image is divided into nine patches. However, instead of using eight configurations, this algorithm has a

pre-defined set of, say, 100 permutations of the nine patches stored with an index in a dictionary. These nine patches are fed to the convolutional network separately and merged at later stages. The pretext task is to identify the pre-defined index, i.e., 1 out of the 100 pre-defined permutation indices. This approach provides more robustness compared to the previous algorithm. While models pre-trained with these auxiliary tasks have shown promising results, the final layers of the network become specifically optimized for the pretext task. As a result, when these models are transferred to new tasks, they often fail to demonstrate comparable improvements (Misra and Maaten 2019).

3.3 Instance-Based Discriminative Methods

Instance-based discrimination approaches consider each individual sample in the dataset as a distinct class. To expand the dataset, each sample undergoes augmentation to create additional instances. However, this method presents a significant challenge: assigning a unique class label to every sample would result in an exceptionally large number of classes. Consequently, training a classification model with such an enormous number of classes becomes computationally intensive and impractical. To address this issue, a pretext task is crafted that preserves the similarity between the original data sample and its augmented version, in other words, the model is trained to learn the concepts that maximizes the mutual information between two views. There are different approaches to maximize this mutual information. These methods typically employ a Siamese-style network setup, consisting of two parallel networks, where the parameters may be shared or separated, depending on the specific approach. In order to achieve the desired objective, the research community has developed various loss functions to optimize. In summary, the key factors in instance discrimination methods are the type of loss function used and the transformations applied to generate augmented data samples. The loss functions aim to maximize the similarity between representations of an original sample and its augmented counterpart while pushing apart representations of different samples. The choice of data augmentation transformations is also crucial, as they should introduce meaningful variations that preserve the underlying content while creating diverse augmented views. By treating each sample as a separate class and optimizing for instance-level similarity, self-supervised methods learn rich and discriminative representations from unlabeled data, which were found beneficial for downstream tasks through transfer learning. The following sub-sections will discuss these different loss functions and general transformation used to generated an augmented sample. The taxonomy of these models has been slightly modified from the one mentioned in the lecture notes on instance-based self-supervised learning from New York University (NYU)[1].

[1] https://atcold.github.io/NYU-DLSP21/en/week10/10-1/

3.3.1 Loss Functions

3.3.1.1 Contrastive

For instance-based discrimination methods based on contrastive learning, the core idea is to align the representations of an original data sample and its augmented version, while simultaneously ensuring that the representations of all other samples have distinct embeddings. The InfoNCE (Oord, Y. Li, and Vinyals 2018) loss function is one of the first in category designed to learn non-trivial representation. Other notable algorithms developed under this category include SimCLR (T. Chen et al. 2020) and MoCo (He, Fan, et al. 2019). These contrastive methods aim to develop representations with two crucial properties: alignment, where representations of an original sample and its augmented counterpart should be similar in the embedding space, and uniformity, ensuring representations of different samples are uniformly distributed and well-separated in the embedding space. Replacing the specialized loss functions (like InfoNCE) with simpler alternatives such as MSE often leads to a trivial solution, where the model weights might optimized to the value of 0, resulting in a 0 vector for all samples in the embedding space. While this satisfies the alignment condition, but it violates uniformity, a phenomenon known as dimensional collapse, as demonstrated in (Tian et al. 2020). Most of the work in this thesis is based on contrastive learning, so all the loss functions in this category are thoroughly discussed.

InfoNCE

InfoNCE (Contrastive Predictive Coding) (Oord, Y. Li, and Vinyals 2018) is an unsupervised learning approach introduced to extract meaningful representations from high-dimensional data. This self-supervised method can be applied to various data types, including audio, images, and text. The core principle behind InfoNCE is the maximization of mutual information between representations. The InfoNCE formulation considers a set of N random samples, comprising one positive sample and $N-1$ negative samples. The objective is defined by Equation (3.1), where the positive sample $f_k(x_{t+k}, c_t)$ is derived from the joint density $p(x_{t+k}, c_t)$. This approach formed the basis for the initial application of Noise Contrastive Estimation (NCE) (Mnih and Kavukcuoglu 2013) to image data. f_k represents a neural network that takes inputs from both the context c_t and the complementary pair x. The network can be used for binary classification tasks. For instance, it may predict class 0 if the complementary pair is positive (i.e., from the same context) and class 1 if the sample is negative (i.e., from a different context).

$$L_N = -\mathbb{E}_x \left[log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \varepsilon X} f_k(x_j, c_t)} \right], X = \{x_1, ...x_n\} \tag{3.1}$$

SimCLR

SimCLR (T. Chen et al. 2020) extended the concept of InfoNCE by introducing a straightforward approach to contrastive learning for visual representations. Through extensive experimentation, the authors of SimCLR developed a set of data augmentation techniques to define an effective contrastive prediction task, resulting in robust representations. The method introduces a loss function called Normalized Cross-entropy, which combines contrastive cross-entropy with temperature hyperparameter to enhance representation learning. While the underlying principle is similar to InfoNCE, SimCLR differs as it uses the normalized dot product between two views of an image, rather than employing a neural network f_k as in the InfoNCE equation (Equation (3.1)).

Figure 3.4 provides a schematic illustration of the SimCLR algorithm. The process begins by passing an image through a stochastic transformation pipeline to generate two different but correlated views of the image. The loss function defined in Equation (3.2) is minimized to optimize the learning process. By leveraging these augmentation strategies and the normalized cross-entropy loss, SimCLR demonstrates the potential of simple contrastive learning approaches in capturing meaningful visual representations without relying on labeled data. This self-supervised method has shown impressive performance on various downstream tasks, particularly in scenarios where labeled data is scarce.

$$L_N = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1,k\neq j}^{2N} exp(sim(z_i, z_k)/\tau)} \tag{3.2}$$

The SimCLR algorithm uses a normalized dot product, denoted as "sim" in Equation (3.2), to measure the similarity between two views of the same image. In this equation, z denotes the encoded representation of the image. The equation includes a parameter τ, known as the temperature, which controls the sensitivity of the loss function. The denominator of the equation takes all negative samples from the batch. A key finding is that the generalization capability of the pre-trained model using SimCLR loss improves with the increase in number of negative samples. However, this characteristic makes such algorithms dependent on high computational memory resources. To address the issue of large memory requirements, the authors of MoCo (He, Fan, et al. 2019) developed an algorithm that offers a solution to this challenge.

MoCo

MoCo (Momentum Contrast) (He, Fan, et al. 2019) is an extension of the SimCLR algorithm. Unlike SimCLR, MoCo employs asymmetric networks, meaning the two networks do not share weights. These networks are referred to as the base (or online) network and the momentum (or target) network. During training, only the weights of the base network (θ_q) are updated directly, while the weights of the momentum network (θ_k) are updated using an Exponential Moving Average (EMA) of the base network's weights, as shown in Equation (3.3).

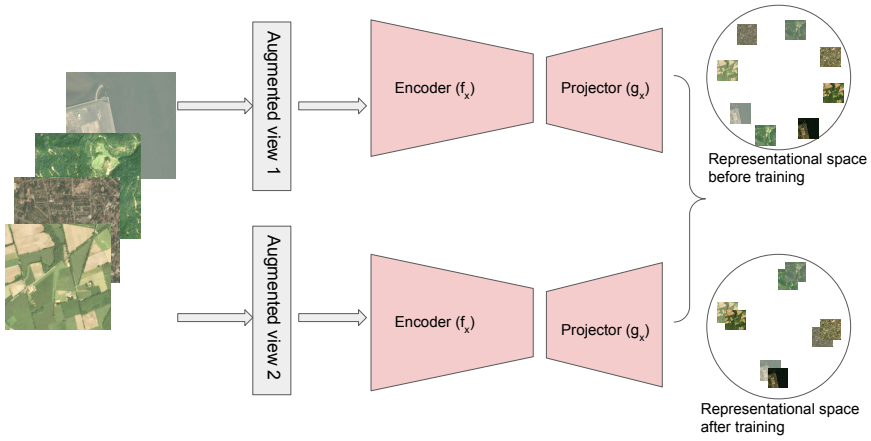


Figure 3.4: **General setup of instance based discrimination using contrastive learning**. The left part depicts a batch of data being passed through the model in an iteration. The contrastive loss function operates on the embedding space, considering all the data samples simultaneously. The right figure shows the distribution of representations in a hypersphere. Initially, before training, the representations are randomly distributed. However, after training with an appropriate contrastive loss objective, the representations of different views (augmented versions) of the same data sample become aligned, while the representations of different samples are uniformly distributed on the hypersphere surface.

$$\theta_k = m\theta_k + (1-m)\theta_q \tag{3.3}$$

MoCo uses a queue to increase the effective number of negative samples. This queue is a large array that stores embeddings of negative data and is updated dynamically using the First In First Out (FIFO) principle. At each iteration, new embeddings are added to the queue, and the oldest embeddings are removed. These stored embeddings are used as negative samples in the loss function, as defined in Equation (3.4). Here, q is the query representation, k_+ represents the embedding of the positive sample, and k_i used in the denominator refers to the embeddings of the negative samples stored in the queue. Figure 3.5 illustrates the general setup of the MoCo algorithm.

$$L_N = -log\frac{exp(q^T k_+)/\tau)}{\sum_{i=1}^{K}(exp(q^T k_i/\tau)} \tag{3.4}$$

In practice, the size of the queue is significantly larger than the batch size. For instance, in the original paper, with a minibatch size of 256 for their 8 GPU training setup, the queue size was set to 65,536, which is 64 times larger. The embeddings added to the queue must remain consistent with the weights of the current training iteration to ensure stability. The EMA update mechanism helps achieve this consistency by gradually updating the weights due to the momentum effect, providing a stable training platform with a larger number of negative samples from the queue. Figure 3.5 illustrates the experimental

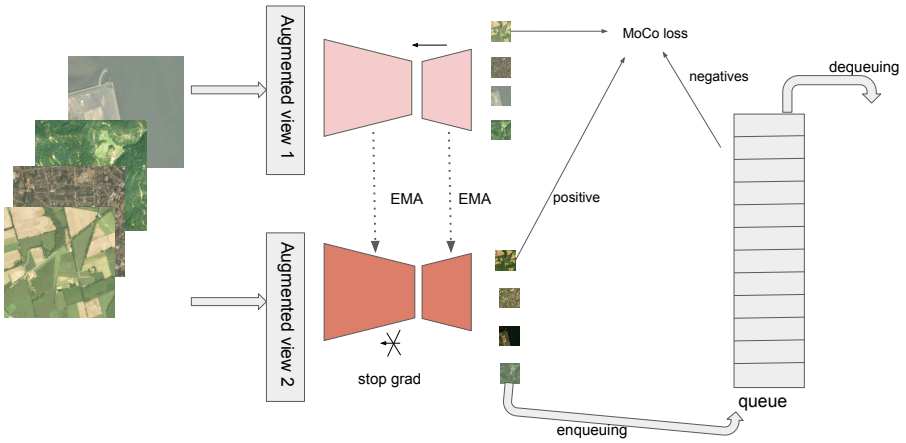


Figure 3.5: **General setup of MoCo algorithm.** The different colors indicate that the weights of the base and momentum networks are not shared. The momentum network is not directly trained; only the gradients of the base network are updated. The target outputs are obtained from the momentum network. To ensure stable training, the weights of the momentum network are updated using the EMA, as shown by the dotted arrow in the figure. Negative samples are drawn from the dynamic queue on the right side, that is updated after each iteration.

setup of MoCo algorithm.

LOOC

Leave-One-Out-Contrastive (LOOC) (Xiao et al. 2021) is a variant of contrastive learning that can be implemented along with existing contrastive self-supervised algorithms. In the original paper, it was implemented using the MoCo algorithm. The primary goal of LOOC is to develop a training strategy that allows the model to learn the nuances between each transformation separately. This approach is advantageous for broader downstream tasks, as some transformations may not be meaningful for certain tasks. LOOC is implemented by using a shared backbone network but different projector networks for each transformation. For example, given two transformations, $\mathbb{T}_1$ and $\mathbb{T}_2$, the first augmented view is a combination of both transformations and is passed to the projector 1. The second view is generated using only transformation $\mathbb{T}_1$ and is passed to projector 2, while the third view is obtained using transformation $\mathbb{T}_2$ and is passed to projector 3. Each projector has its own queue. For projector 1, the negatives are different data samples. For projector 2, the negatives include other data samples and views obtained from transformation $\mathbb{T}_2$. Similarly, for projector 3, the negatives include other data samples and views obtained from transformation $\mathbb{T}_1$. Each projector has its own contrastive loss, but they share a common backbone network. This setup allows the model to learn both general transformation properties and individual transformations during pre-training. Figure 3.6 (taken from the original paper) gives a visual description of the LOOC strategy.

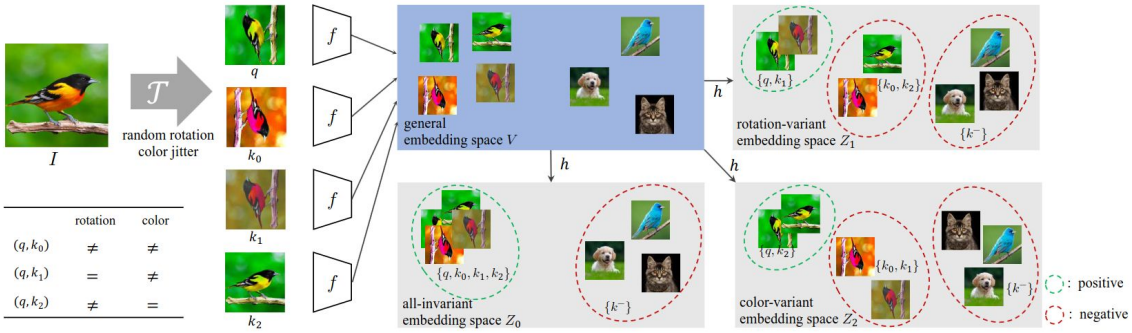


Figure 3.6: **Experimental setup of LOOC algorithm.**The figure shows an image of a parrot processed through the same backbone network with different transformations. On the right side, all the transformed images are passed through different projectors, denoted by h. For each embedding, the positive and negative queries are varied accordingly. This approach enhances robustness across various downstream tasks (Xiao et al. 2021).

The other instance-based discriminative loss functions mentioned subsequently were not employed in this research. Therefore, they are briefly discussed to offer readers an insight of other loss functions.

3.3.1.2 Clustering

This method aims to learn representations that capture instance-level similarity by assigning similar cluster assignments or scores to an original data sample and its augmented version. The approach involves passing a batch of data samples through a model to obtain representations of the samples in an embedding space. A clustering algorithm, such as K-means, is then applied to these representations to assign cluster labels or scores to each data point. These cluster assignments serve as pseudo-labels for the original data samples. Next, the augmented versions of the same data samples are passed through the model, and the model is optimized using a cross-entropy loss function, treating the cluster assignments from the previous step as targets for the augmented data. Algorithms like DeepCluster (Caron, Bojanowski, et al. 2018) and DeepClusterV2 (Caron, Misra, et al. 2020) follow this approach, where the cluster assignments are represented as one-hot encoded vectors corresponding to the cluster indices. An extension of this idea is to use soft cluster assignments instead of hard assignments. In this case, rather than assigning a single cluster index, each data point is associated with a distribution over multiple clusters. The Sinkhorn-Knopp algorithm is employed to obtain these soft cluster assignments or scores, which represent the degree of association between a data point and each cluster (Cuturi 2013). Examples of instance discrimination methods using soft cluster assignments include SeLa (YM., C., and A. 2020) and SwAV (Caron, Misra, et al. 2020). Unlike contrastive algorithms, this approach focuses on aligning cluster scores directly,

rather than aligning the representation vector in the projected space. Figure 3.7 illustrates the clustering based instance discriminative loss.

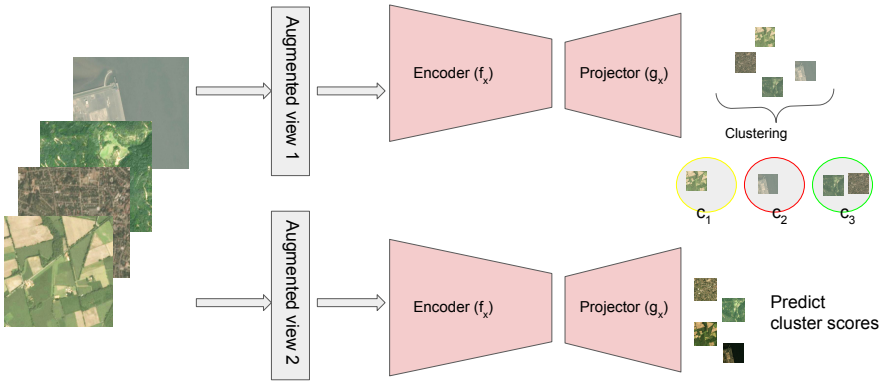


Figure 3.7: **General setup of clustering based instance based discrimination algorithm**. The top part of the figure depicts the process where all data samples are passed through the model to obtain representations in the embedding space. A clustering algorithm is then applied to these representations, assigning cluster labels or scores to each data point. The augmented versions of the data samples are also passed through the same model. The objective is for the model to predict the same cluster assignment for an original sample and its augmented counterpart. A cross-entropy loss function is employed, treating the cluster assignments from the original data as targets for the augmented samples.

3.3.1.3 Distillation

Distillation refers to the process of transferring knowledge from one model to another (G. Hinton, Vinyals, and Dean 2015). Distillation can also be employed as a form of instance discrimination self-supervised learning. This approach involves the use of two models: a *target* model and an *online* model. In some literature, these distillation model are also referred as a teacher-student model. The weights of the online model are trainable, while the weights of the target model are not directly trained, similar to the one used in MoCo. Instead, the target model's weights are updated using an EMA of the online model's weights. The objective of the online model is to produce an output representation that match those of the output representation of an augmented sample from the target model, effectively distilling the knowledge from the target model. This distillation process enables the online model to learn representations that capture instance-level similarity utilizing the positive pair of samples. Notable algorithms that fall under this category are SimSiam (X. Chen and He 2020), BYOL (Grill et al. 2020), and DiNo (Caron, Touvron, et al. 2021). The distillation based approach leverages the knowledge encoded in the target model's representations to guide the training of the online model, without requiring explicit labels or pretext tasks. Figure 3.8 illustrates the distillation based instance discrimination loss.

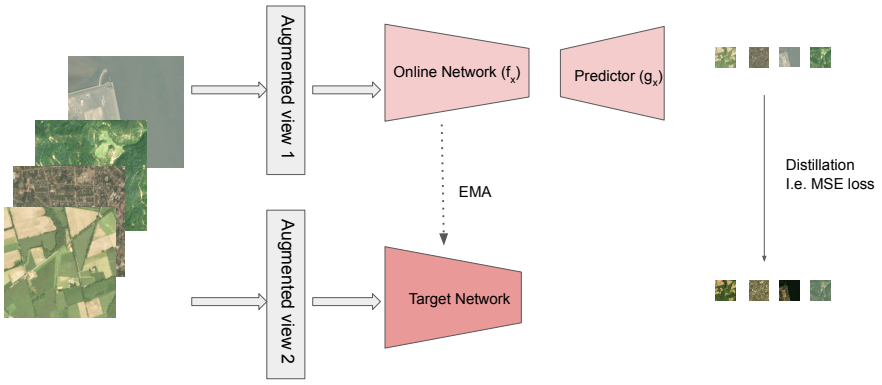


Figure 3.8: **General setup of instance-based discrimination algorithm using distillation.** The loss function used is a simple MSE loss between the output from the predictor head of the online network and the output of the augmented version of the same sample from the target network. This setup facilitates knowledge distillation from the target network to the online network. The asymmetric design is due to the presence of predictor head in the online network. The MSE loss between the outputs of the two asymmetric networks for the same augmented sample, enables the online network to learn from the target network's representations simultaneously ensuring it does not converges to a trivial solution.

3.3.1.4 Redundancy Reduction

The redundancy reduction variant draws inspiration from a principle proposed in neuroscience. (Barlow and Rosenblith 1961) mentions the hypothesis that the goal of sensory processing is to recode highly redundant sensory inputs into a factorial code. The principle underlying this approach is the development of an objective function designed to bring the cross-correlation matrix closer to an identity matrix, thereby encouraging decorrelated representations. To explain this concept more simply: In traditional methods, the normalized dot product of two output representations is typically constrained to be 1 for similar pairs. In contrast, this approach takes a different route. Here, the outer product of the two output representations, which results in a matrix, is encouraged to approximate an identity matrix. The Barlow Twins (Zbontar et al. 2021) algorithm falls under this category of instance-based discrimination methods. It is based on the idea of minimizing the redundancy between the components of the learned representations, effectively decorrelating them. Another algorithm in this category is VICReg (Bardes, Ponce, and LeCun 2021), which employs an ensemble of multiple loss functions. The "v" component enforces uniformity by minimizing the variance of the representations across the batch (Equation (3.5), where d refers to the dimension of the model output and n refers to the number of samples in the batch). The "i" component enforces alignment or similarity between representations of an instance and its augmented version (Equation (3.6)). The "c" component minimizes the cross-correlation between the components of the representations (Equation (3.7)). In the all the equations, z refers to representation of the image. Figure 3.9 illustrates such loss function.

$$v(Z) = \frac{1}{d}\sum_{j=1}^{d} max(0, 1 - \sqrt{Var(z) + \varepsilon}) \tag{3.5}$$

$$i(Z) = \frac{1}{n}\sum_{i} ||z_i - z_i'||_2^2 \tag{3.6}$$

$$c(Z) = \frac{1}{n-1}(z_i - \bar{z})(z_i - \bar{z})^T, \text{ where } \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{3.7}$$

The W-MSE (Whitening Mean Squared Error) (Ermolov et al. 2020) is a variant that combines elements from both the distillation and correlation-based approaches to self-supervised learning. It employs a whitening operation, which is a form of inter-positive distance correlation, to prevent degenerated solutions and encourage the learned representations to be uniformly distributed on a hypersphere. The whitening operation aims to decorrelate the components of the representations, ensuring that they are statistically independent and have unit variance. This decorrelation step helps to avoid the dimensional collapse issue, where all representations collapse to a single point or subspace, which is a trivial solution that satisfies the objective but fails to capture meaningful information.

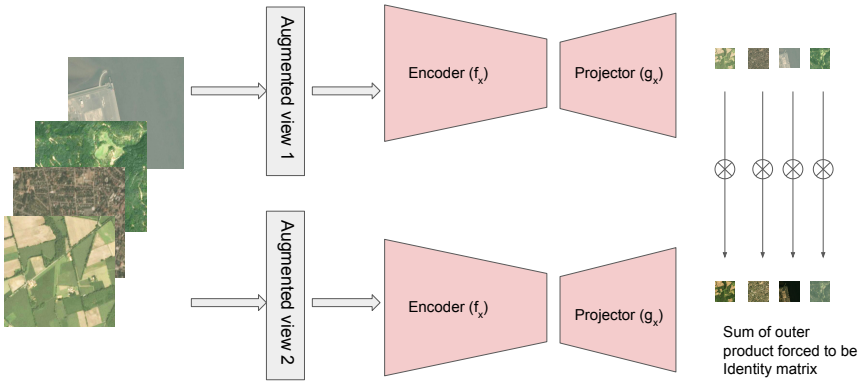


Figure 3.9: **General setup of instance based discrimination loss using redundancy reduction algorithm**. The loss function encourages the learned representations by approximating the sum of outer products between the representation of a data sample (z) and its augmented counterpart ($\tilde{z}$) to approximate an identity matrix. Mathematically, the objective is to minimize the difference between the sum of outer products $\sum_i z_i \otimes \tilde{z}_i$ and the identity matrix I, where $\otimes$ denotes the outer product operation.

3.3.1.5 Relational Similarity

This variant of instance discrimination self-supervised method aims to optimize the similarity between one data sample and the other samples within the same batch to that of its corresponding augmented samples as illustrated in Figure 3.10. In other words, the

objective is to match the probability distribution of similarity scores between the augmented sample and the other samples in the batch with the target similarity distribution computed from the original samples. The Kullback-Leibler (KL) divergence loss function is employed to minimize the difference between the target and online probability distributions, thereby optimizing for the desired similarity relationships. Algorithms such as Relational Self-Supervised Learning (ReSSL) (Zheng et al. 2021) and Iterative Similarity Distillation (ISD) (Tejankar, Koohpayegani, Pillai, et al. 2020) fall under this category of instance discrimination methods.

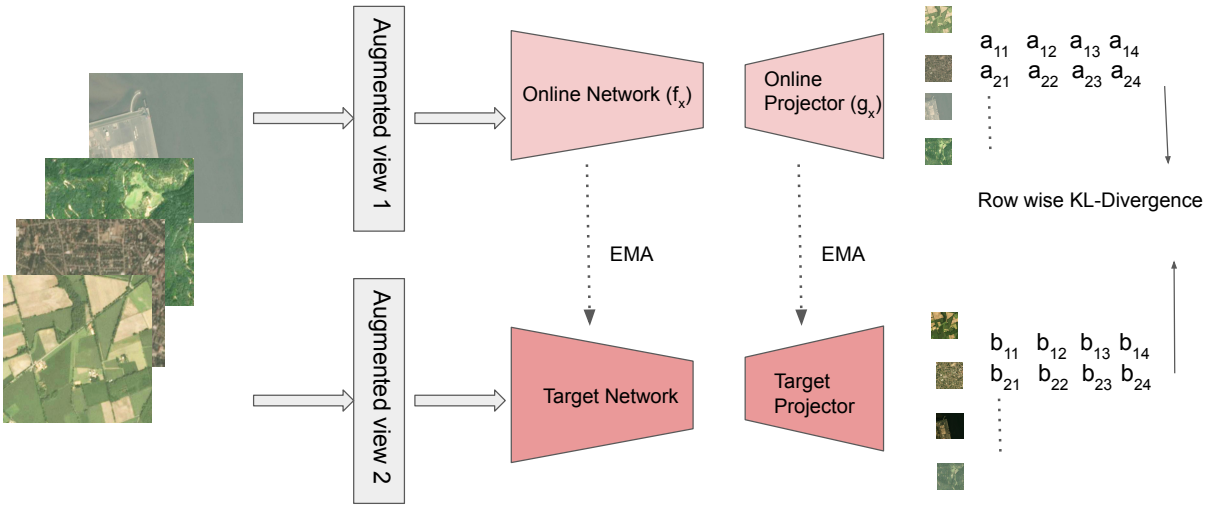


Figure 3.10: **General setup of instance-based discrimination loss using relational similarity algorithm**. In Relational Similarity instance discrimination method, instead of maximizing the similarity between a data sample and its augmented view, this approach focuses on the similarity relationships between one data sample and the other samples within the same batch. The objective is to match the similarity distribution of an augmented sample to the target similarity distribution computed from the original samples within the batch which is mentioned as Row-wise KL-Divergence in the figure.

3.3.2 General Transformations

As discussed before, the instance-based discrimination self-supervised methods also rely on transformations in order to generate a meaningful augmentation of the sample image which then the loss functions by maximizing the mutual information can learn meaningful representation. The majority of research in the field of instance discrimination type self-supervised learning has been implemented on natural images. This section concentrates on data augmentation techniques. Subsection 3.3.2.1 outlines standard transformations commonly applied to natural images, followed by a subsection discussing transformations specific to remote sensing applications.

3.3.2.1 Image Transformations

The instance level discrimination type of self-supervised learning relies on transformations to obtain augmented views. The significance of transformations has been highlighted in several other studies (Purushwalkam and Gupta 2020; Gansbeke et al. 2021). Transformations should be designed to distort the data while preserving its semantic information. In most remote sensing applications using multi-spectral imagery, the community utilizes transformation methods originally developed for natural images in computer vision. The computer vision community has largely standardized transformations for natural images. Some remote sensing studies have adopted these transformations for contrastive self-supervised learning in their applications. Different transformations used by the computer vision community are discussed in this subsection.

Random Cropping and Resizing

Random cropping is a transformation technique that, instead of distorting the image, uses a subset of the original image. Typically, implementations randomly cropping involves randomly selecting 20-100% of the original image area. After cropping, the selected portion is rescaled to match the original image dimensions using interpolation techniques. The purpose of this transformation is to enable the model to learn by identifying similarities between different crops of the same image in the embedding/representational space. This approach helps the model develop robustness to variations in scale and position, encouraging it to focus on salient features that persist across different views of the same image.

By presenting the model with various cropped versions of the same image during training, it learns to recognize the underlying semantic content regardless of its specific location or scale within the image. This contributes to the model's ability to generalize and perform well on downstream tasks, even when objects or features appear at different scales or positions.

Color Jittering and Grayscaling

This transformation aims to preserve the texture of an object by combining four image manipulation methods: contrast, saturation, brightness, and hue. The values for contrast, saturation, brightness, and hue are randomly selected within a specified range, ensuring that each time it produces a new view with a similar texture.

In the case of grayscaling, the grayscale version of the image is obtained using the formula in Equation (3.8). To ensure compatibility with networks that require three-channel RGB images, the grayscale image is duplicated across all three channels.

$$gray = (0.2989 \times red + 0.587 \times green + 0.114 \times blue) \tag{3.8}$$

This approach allows the model to learn by maintaining the texture information while varying other visual attributes, thereby enhancing its ability to generalize across different

visual conditions.

Horizontal Flipping

Horizontal flipping is a transformation technique that produces two versions of an image: The original, unaltered image and a horizontally mirrored version of the original image. This transformation is applied with a probability of 0.5, meaning there's an equal chance of the image remaining unchanged or being flipped horizontally. The flipping occurs along the vertical axis, effectively reversing the left-right orientation of the image. This simple yet effective augmentation technique helps the model learn features that are invariant to horizontal orientation.

Gaussian Blurring

In this transformation, the image is blurred using information from the surrounding pixels. The influence of these surrounding pixels is determined by a Gaussian distribution. A random value is selected to determine the radius (standard deviation parameter) of the Gaussian. The pixel value is then replaced by an aggregated value based on this distribution.

3.3.2.2 Standard Augmentation Pipeline

In the preceding sections, transformations developed by the computer vision community are being explored. These transformations have been adopted by researchers in remote sensing for a long time. This section will first examine the standard augmentation pipeline that incorporates the previously discussed transformations. Following that, two additional augmentation techniques specific to remote sensing images are covered. These are the nearest neighbor method and the use of time as an augmentation factor.

Standard Transformation Pipeline

A standard transformation pipeline is a dynamic process that generates diverse views of an image during each training epoch. This pipeline incorporates all the transformations discussed in the previous section on data augmentation techniques. The probability of applying each transformation is specified in Table 3.1, which outlines the standard augmentation parameters.

The standard augmentation pipeline begins with random cropping, where a portion of the image is selected at a random location, with the crop size ranging between 20% and 100% of the original image. This process introduces significant variability in each iteration. After cropping and resizing, the image has a 50% chance of undergoing color jittering with varying intensities. Subsequently, there's a 20% probability of converting the image to grayscale. Following this, the image may be horizontally flipped with a 50% probability, and then subjected to Gaussian blurring, also with a 50% probability. The combination and sequence of these random transformations significantly increase the stochasticity of

Table 3.1: The table outlines the transformations employed in the standard augmentation pipeline. The "Probability" column indicates the likelihood of each transformation being applied during the augmentation process. For certain transformations, additional parameters are specified in the factor column, providing more details on the value of the parameter yielding a range for distortion from the specific transformation.

Transformation	Factors	Probability
Random cropping and Resizing	crop factor :(20%-100%)	1.0
Color Jittering	range of (brightness, saturation, contrast and hue): (0.4, 0.4, 0.4, 0.1)	0.8
Grayscale		0.2
Horizontal Flip		0.5
Gaussian Blurring	radius of Gaussian kernel: (0.1, 2.0)	0.5

the augmentation process. Figure 3.11 illustrates various views of a sample image after applying this augmentation pipeline. This approach ensures that each training iteration presents the model with a unique variation of the input image, enhancing its ability to learn robust and generalizable features across different scales, colors, orientations, and levels of detail.

Figure 3.11: **Different view of an image generated by the augmentation pipeline**. The original image is shown on the left, while the four augmented versions of the same image, produced by the standard augmentation pipeline, are displayed on the right.

Nearest Neighbor View

For instance discrimination tasks, to obtain more diverse views, several studies use images that are closest to the original image in the latent space. The studies claim that images lying close in the latent space share more similar semantic information and are more likely to belong to the same class. Consequently, they use the nearest neighbor images along with the standard augmentation pipeline to generate additional views. Examples of works based on this approach include NNCLR (Dwibedi et al. 2021), NNMoCo (Wang

et al. 2023), and MeanShift (Tejankar, Koohpayegani, Navaneet, et al. 2021).

The Time-Dimension as a Transformation for Remote Sensing Data

Meaningful transformations are crucial for improving the representation learning in instance-based self-supervised learning. The importance of transformations for downstream tasks was demonstrated by the authors of (Purushwalkam and Gupta 2020). They found that certain pre-training transformations may lead to unsuitable representations for specific downstream tasks. For example, when pre-training on images of different rooms (e.g., kitchens and living rooms) and for a downstream task of identifying furniture and devices (e.g., sofas, dining tables, TVs), transformations like random cropping and resizing were not beneficial. This highlights the need to carefully select transformations that align with the intended downstream tasks. Remote sensing images differ from natural images in that they often include metadata such as time and geolocation. Researchers in the field have leveraged this unique aspect by using temporal differences as a form of augmentation. For instance, they use two images of the same location captured at different times, such as during different seasons, as augmented views of the image. This approach is effective because many land surface features change slowly over time, providing strong supervisory signals for instance-based discrimination in self-supervised learning. Figure 3.12 illustrates this concept. Several studies, including Geography-aware self-supervised learning (Ayush et al. 2020) and Seasonal contrast (Mañas et al. 2021), have incorporated this temporal transformation in their work. This approach takes advantage of the temporal dimension inherent in remote sensing data to enhance the learning of meaningful representations.

Figure 3.12: **Time used as a transformation method on a sample from the SeCo (Mañas et al. 2021) dataset.** It displays multiple images of the same land area captured at different times throughout the year, demonstrating the seasonal changes in the land surface. This approach leverages temporal variations as a form of data augmentation for remote sensing applications.

3.4 BERT

BERT (Bi-Directional Encoder Representations from Transformers) (Devlin et al. 2018) is a self-supervised technique used to encode words into a semantic vector space. Prior to

BERT, methods such as continuous bag of words (CBOW) and skip-gram were employed to obtain semantic representations of words from their one-hot encodings (Mikolov et al. 2013). However, these methods focused solely on individual words, failing to capture the contextual information. For instance, the word ”bank” can refer to a financial institution where people open accounts or a river bank, depending on the context. The same word carries different meanings in different contexts. BERT addresses this issue by utilizing the encoder architecture of the original transformer network (Vaswani et al. 2017). The attention mechanism employed in the transformer allows a word to interact with other words in the sentence or paragraph. Through stacked attention layers, complex relationships are captured, and each word is embedded semantically and contextually. Figure 3.13 illustrates the BERT architecture. There are two ways to train BERT models: Mask Language Modeling (MLM) and Next Sentence Prediction (NSP) approach. In MLM, a few tokens are masked, either replaced by random numbers, zeros, or another word. The masked sentence is passed through the network, and the network is trained to output the original word that was masked. In the NSP, two sentences are randomly selected from a text. If both sentences are consecutive in the text, it is classified as positive; otherwise, if they are not consecutive, it is classified as negative. By leveraging the transformer architecture and self-supervised training algorithms like MLM and NSP, BERT learns rich contextual representations of words, capturing the nuances of language and enabling better performance on various NLP tasks through transfer learning.

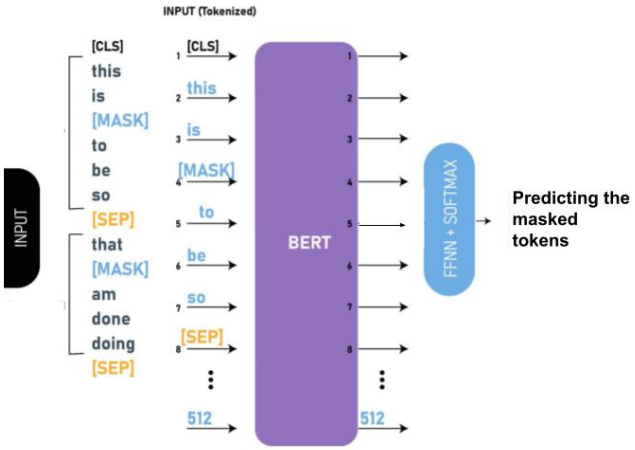


Figure 3.13: **The BERT algorithm**. On the left side, the original time series is shown. Prior to input into the transformer model, select tokens are masked. This masked time series is then fed into the model. The algorithm is designed to comprehend patterns even with masked elements, learning to predict accurate representations. The output is then passed through a single or shallow neural network to predict the original masked inputs. For the original task of word prediction in natural language, the BERT model typically employs a cross-entropy loss function. However, when adapting BERT for time series analysis, a conventional Mean Squared Error (MSE) loss function is more commonly used. Figure adapted from [a]

a=https://www.geeksforgeeks.org/understanding-bert-nlp

3.5 Multi-Modal Contrastive Self-Supervised Learning

Multi-modal contrastive self-supervised learning expands on traditional contrastive learning by incorporating multiple modes or sources of information. This approach draws inspiration from the human ability to perceive the world using various sensory inputs, such as visual information and verbal descriptions. Figure 3.14 provides an illustrative example of multi-modal learning, demonstrating how different types of data can be integrated to enhance the learning process.

Figure 3.14: **Different modes of accessing public bus information in Aachen.** On the left, a button is shown which, when pressed, provides an audio announcement of the scheduled arrival times for all buses at the stop. On the right, the same information is visually displayed on an electronic board. This example demonstrates how multi-modal systems can present identical information through different sensory channels - in this case, auditory and visual.

Figure 3.14 illustrates how different sensory modes can convey the same information. A deaf person can access bus schedule details visually from a display board, while a blind person can obtain the same information audibly by pressing a button. Although these are distinct methods, both sources provide identical information. This concept of complementing one information source with another has proven powerful in deep learning applications (S. Ma et al. 2020). It is currently being extensively used in the development of advanced models such as Stable Diffusion (Rombach et al. 2021), GPT-4 (OpenAI 2023), and Fromage (Koh, Salakhutdinov, and Fried 2023). These models leverage multiple modalities to enhance their understanding and generation capabilities. Figure 3.15 presents a schematic representation of multi-modal contrastive learning, demonstrating how different types of data can be integrated within a single learning framework. This approach aims to capture the complementary nature of various data modalities, potentially leading to more robust and comprehensive representations.

In multi-modal contrastive learning, multi-modal training data is required during the model's training phase. However, in the final application, the user does not necessarily need both data sources, as illustrated in Figure 3.15. In these methods, finding a suitable

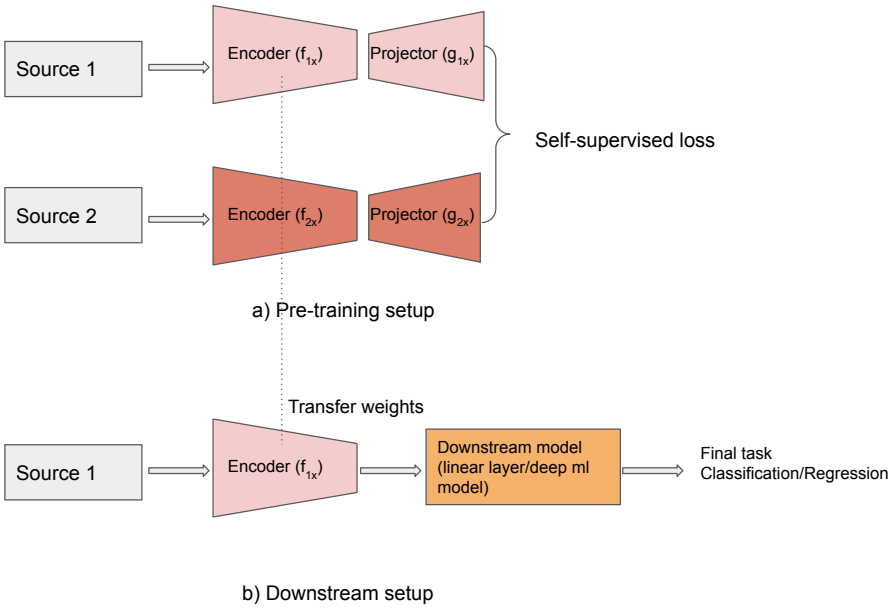


Figure 3.15: **The experimental setup for multi-modal contrastive learning**. The upper part depicts the pre-training phase, where two distinct data sources are processed through separate networks due to their differences in nature, shape, and size. This setup can incorporate any of the previously discussed conventional loss functions. The lower part shows that after pre-training, the model can operate using only one source (in this case, source 1), demonstrating the flexibility of the trained system to function with limited input modalities.

complementary source enhances the pre-training process. Essentially, in multi-modal self-supervised learning, the model corresponding to source 1 guides the model corresponding to source 2, and vice versa.

CHAPTER 4

Generating Views Using Atmospheric Transformation for Multi-Spectral Images

This chapter introduces our novel method called atmospheric transformation for generating augmented views of multi-spectral images for contrastive learning. This technique can be applied across all spectral channels of multi-spectral images while preserving the physical characteristics of land surface data. By replacing color jittering and grayscaling in the standard transformation pipeline with our proposed atmospheric transformation, contrastive self-supervised pre-trained models are better able to utilize the rich details present in multi-spectral images. Atmospheric transformation demonstrates the superiority over baseline methods through various downstream tasks.

Individual Contribution

The following chapter is based on the publication (Patnala et al. 2023)

Generating Views Using Atmospheric Transformation for Contrastive Self-Supervised Learning on Multi-Spectral Images
Ankit Patnala, Scarlet Stadtler, Martin G. Schultz, and Juergen Gall
IEEE Geoscience Remote Sensing

This study was a collaborative effort between Ankit Patnala and Scarlet Stadtler, with significant scientific insights and technical guidance provided by Martin G. Schultz and Juergen Gall. Their valuable input and feedback were instrumental in the smooth execution of the project. The initial concept originated from Ankit Patnala, inspired by discussions with colleagues in the remote sensing field at Forschungszentrum Juelich. This idea was further refined through brainstorming sessions involving Scarlet Stadtler, Martin G. Schultz, and Juergen Gall. Ankit Patnala took the lead in implementing the project and conducting the evaluation. Ankit Patnala authored the initial manuscript and Scarlet with her academic expertise aided in making it scientific enough by employing appropriate terminology and imbuing it with a nice scientific outlook. The successful

completion of this work was made possible by the combined efforts and expertise of all team members involved.

Contents

4.1 Introduction

Computer vision methods are crucial in the field of remote sensing, encompassing a wide range of applications (Mañas et al. 2021). Specifically, land cover classification aims at labeling contiguous regions of the Earth's surface based on characteristic surface reflectance patterns. Most land cover classification algorithms use information from various channels of the visible spectrum, as measured by satellite or airborne instruments. For several years machine learning has been explored in the remote sensing community for this task. The community has developed a small number of benchmark datasets for land cover classification, for example, small-sized labeled land cover datasets such as Sat-4 (Basu et al. 2015), Sat-6 (Basu et al. 2015), Eurosat (Helber et al. 2019), and medium-sized datasets like Bigearthnet (Sumbul et al. 2019), and Sen12ms (Schmitt et al. 2019). While these have proven their use, they are rather small compared to state-of-the-art benchmark datasets for image recognition or the petabytes of data publicly available from satellite missions.

Most previous machine learning studies on land cover classification used supervised training approaches where a neural network is trained on labeled data and then tested on another portion of the labeled dataset that was withheld from the model during training. Newer machine learning techniques allow the extraction of useful information from unlabelled data and thus open new possibilities to train machine learning models on much larger amounts of data than before. Due to the difference in nature of the multi-spectral image to that of the RGB image, in this study, we focus on contrastive learning as one fundamental approach to train on these unlabeled multi-spectral data.

In particular, multi-spectral remote sensing images provide different surface reflection properties useful for land cover classification. It would be a waste of information if their analysis is limited to the red, green, and blue (RGB) channels commonly found in natural images. Sentinel (ESA 2021), for example, has also bands such as red edge, near-infrared (NIR) and short-wave infrared. The near-infrared band is particularly useful for the classification of vegetation and the assessment of the healthiness of vegetation as it relates to the leaf area index (Vescovo et al. 2012). Figure 4.1 shows a box plot of NIR radiations for different classes of the Eurosat dataset. Land cover classes such as pasture lands, crop fields, and forests reflect significantly more NIR radiation than classes like industrial buildings, residential places, vegetation-lacking urban land, and water body land cover types such as rivers, oceans, and lakes. Thus, adding an NIR channel to the input data stream of a deep learning model will allow the model to learn more distinguishable features of land cover types and improve scores on land cover classification tasks, especially for vegetated land surfaces.

In order to classify land cover and specifically vegetated land cover from petabytes of unlabeled Sentinel2 remote sensing data, we use a self-supervised contrastive deep learning model. Such contrastive learning approaches rely on the type of transformation used in creating augmented data, i.e. the controlled creation of variants of the original data samples. Augmentations are task-specific and must be curated accordingly (Xiao et al. 2021; Purushwalkam and Gupta 2020). While remote sensing data can in principle be considered as images, they usually come with more spectral channels. Therefore, the transformations applied to remote sensing data should be able to take these into account. So far, the remote sensing community has adopted the same transformations that are used in natural image processing. However, especially the widely used color jittering can only be applied to RGB images. Naively extending color jittering to other spectral channels like NIR leads to a deterioration of physical information. For example, brighter signals in the NIR channel imply healthier plants. Therefore, we set out to develop a new transformation pipeline based on atmospheric correction, which can be applied to all channels of remote sensing images and preserve physical consistency.

Atmospheric correction is a process to remove atmospheric effects on the spectral signature of the reflected light (Baetens, Desjardins, and Hagolle 2019). Sentinel2 measures the solar radiation that is reflected at the Earth's surface. In the atmosphere, light is absorbed and scattered and this must be corrected to retrieve the true spectral signature of the objects on the Earth's surface. Absorption reduces the intensity of pixels, causing haziness, whereas scattering will affect the readings at neighboring pixels. Atmospheric correction affects the spectral reading of all bands and hence plays an important role in land cover classification (Rumora, Miler, and Medak 2020). Currently, there are many atmospheric correction algorithms in use to convert Top of Atmosphere (TOA) images to Bottom of Atmosphere (BOA) reflections. In this study, we used Sen2Cor as it is readily available with GEE. A more detailed explanation of Sen2Cor is described in subsection 4.1.1. For our contrastive learning approach, we exploit the fact that atmospheric correction produces pairs of uncorrected and corrected images. Furthermore, atmospheric correction is applied to all image channels, so that this "augmentation" can be used for multi-spectral

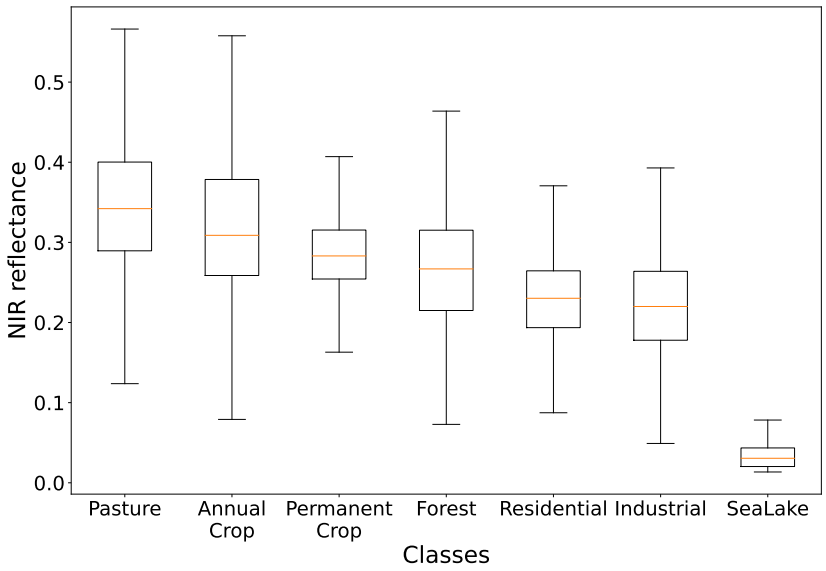


Figure 4.1: **Boxplot of NIR reflectance of different land cover types randomly sampled from Eurosat dataset**. The plot shows the differences between densely vegetated and non-vegetated land cover types. This plot is drawn from 500 images from each land cover type of which 30 pixels were randomly selected.

data and not only for visible RGB images as done in most prior methods (T. Chen et al. 2020). The pre-trained model obtained using our atmospheric correction based transformation as an alternative to color jittering yields better scores on two different land cover classification tasks.

4.1.1 Sen2Cor

Sen2Cor is an atmospheric correction tool that processes Sentinel2 L1C images to produce Sentinel2 L2A images. It comprises two main components: the Scene Classification (SCL) module and the Atmospheric Correction (AC) module. The SCL algorithm identifies clouds, their shadows, and snow, generating a classification map. This map is then utilized by the AC module to differentiate between water, cloudy, and clear pixels. The AC module employs a set of lookup tables, which vary based on the geographic location and climatology of the scene. To retrieve the Aerosol Optical Thickness (AOT) map, Sen2Cor uses Sentinel2 bands 12 (SWIR), 4(red), and 2(blue). The AOT map indicates the visual transparency of the atmosphere. Bands 8*A* (narrow NIR), and 9 (water vapour) are used to retrieve the water vapor map, which determines atmospheric absorption. These parameters are crucial in characterizing the state of the atmosphere. By using these parameters along with the radiative transfer look-up table, Sen2cor removes atmospheric effects and generate BOA images. Sen2Cor also provides additional corrections, such as cirrus and

adjacency corrections. Figure 4.2, taken from the original Sen2Cor paper (Main-Knorn et al. 2017), illustrates a schematic diagram of the Sen2Cor processor.

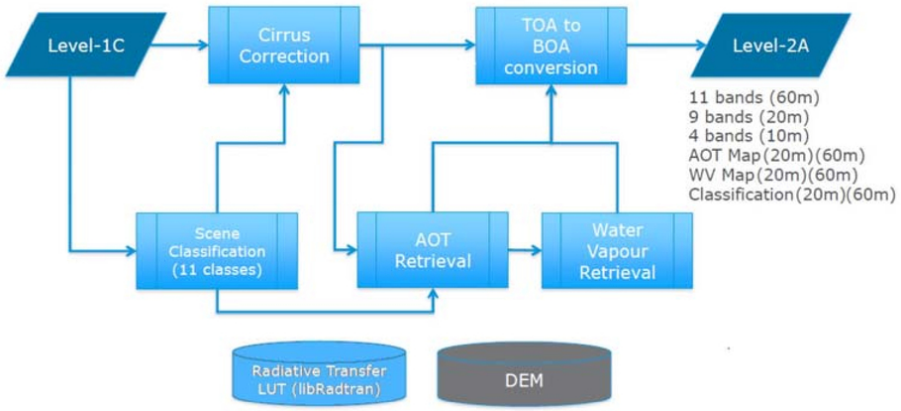


Figure 4.2: The flow chart of all the subroutines used in Sen2Cor(Main-Knorn et al. 2017).
TOA - Top of Atmosphere
BOA - Bottom of Atmosphere
AOT - Aerosol Optical Thickness
DEM - Digital Elevation Model

4.2 Methods

For contrastive learning, the type of contrastive learning setup and transformations used in obtaining augmented views play an important role. For our experiments, we use variants of MoCo (He, Fan, et al. 2019) and to obtain augmented views specific to multi-spectral images, we use an atmospheric transformation instead of color jittering. MoCo and augmented view generation using atmospheric transformation are explained in subsection 4.2.1 and 4.2.2 correspondingly.

4.2.1 MoCo Experiment Setup

As common to contrastive learning, MoCo consists of two networks. However, in MoCo, these networks are asymmetric. One is the trainable base and the other one is the non-trainable momentum network. The idea behind the momentum network is to obtain more negative samples to increase the generalizability of the pre-trained model. During training MoCo, the embeddings of the data obtained in previous iterations are appended to the queue to augment the number of negative samples. The negatives are updated in the queue dynamically, i.e. by flushing the old embeddings and filling the queue with new embeddings. That leads to the challenge of ensuring consistency between the embeddings of the previous iterations and the current training step. Therefore, the momentum network is updated slowly using the weights of the base network as shown in Equation (4.1).

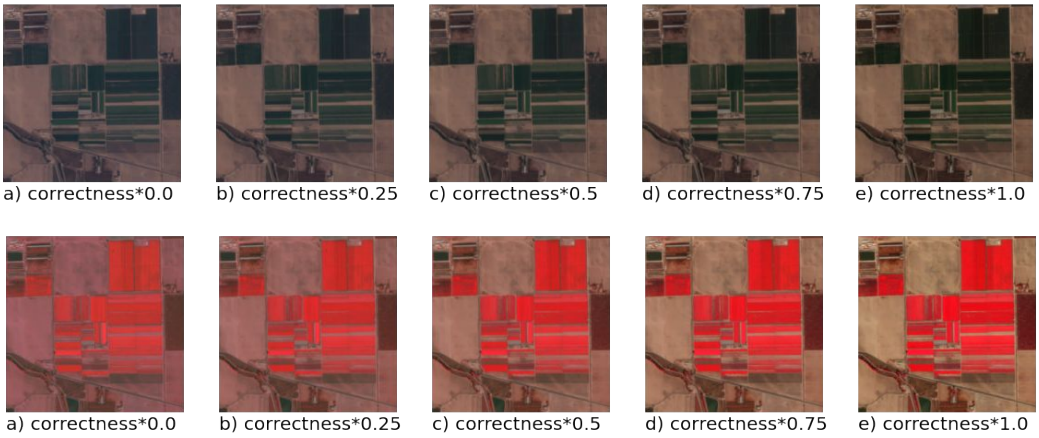


Figure 4.3: **Different degrees of atmospheric correction on a sample image**. In the top row, images are visualized in RGB scale. In the bottom row, images are visualized in the (NIR)RG scale. The first image on the left of column a) contains an atmospheric uncorrected image. The last image on the right i.e. column e) contains the atmospheric corrected version of the same image. Based on Algorithm 1, partially corrected images are generated which can be seen in the middle columns b)-d). As we move from left to right, the features become clearer, i.e. haziness in the top row decreases while the redness increases in the bottom row. These image variants can be used to build similar pairs for contrastive self-supervised learning.

$$\Theta_m = m\Theta_m + (1-m)\Theta_b \tag{4.1}$$

4.2.2 View Generation using Atmospheric Transformation

In self-supervised learning on natural images, color jittering is applied. In this transformation, the user defines the range of color distortion to produce new views of the same image. During each epoch, the image is distorted with different levels of brightness, saturation, contrast, and hue. Extending the algorithm to jitter other channels would distort the valuable physical information in those channels. Therefore, we use atmospheric correction to perturb the channels of multi-spectral images without distorting the physical meaning of the land cover information. In this study, we use four channels: RGB and NIR. The approach is summarized by Algorithm 1. Figure 4.3 shows a visualization of images obtained via the Algorithm 1 for one sample image of the SeCo dataset.

Algorithm 1 Algorithm to generate atmospheric transformed samples.

GIVEN Sentinel1C image (**s1c**), Sentinel2A image (**s2a**)
OUTPUT Atmospheric transformed image (**atc_image**)
$atmospheric\_differences(d) = s2a - s1c$
$uniform\_sampler(u) = \mathbb{U}(0,1)$
$atc\_image = s1c + u * d$
return atc_image

Figure 4.4 shows a schematic comparison of the transformation pipeline that we used in our work to integrate four channels to the existing baseline transformation pipeline.

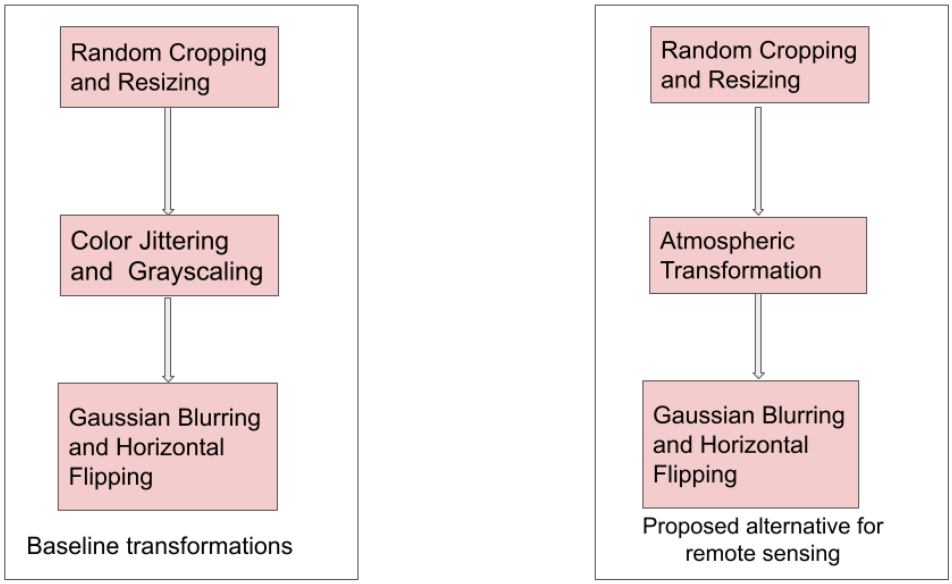


Figure 4.4: **Transformation pipelines**. The left flow diagram shows the baseline transformation pipeline, the flow diagram in the right shows our pipeline. In contrast to color based transformations, the atmospheric transformations are constrained by physics and preserve land cover cues.

4.3 Datasets

In our work, we used all the benchmark datasets which are already published. For self-supervised learning, two different categories of datasets are required, pre-training and downstream datasets. Generally, for the pre-training dataset, there is no requirement of labels, hence it mitigates the huge expenses and time on annotation as well as compensation for skilled human resources. For the static land cover classification tasks, we used the SeCo (Mañas et al. 2021) dataset released by authors of the SeCo algorithm for pre-training and for downstream tasks, Bigearthnet and EuroSAT datasets are used. Sections 4.3.1, 4.3.2, 4.3.3 give a detailed description of SeCo, Bigearthnet, and EuroSAT datasets, respectively.

4.3.1 SeCo dataset

The dataset released by the authors of SeCo was intended to do scalable self-supervised learning on remote sensing images. They used GEE and a distribution strategy to get as many images as required covering various land cover types.

Flowchart 4.5 shows the strategy on how the images were obtained.

4.3.2 Bigearthnet

The creators of Bigearthnet (Sumbul et al. 2019) have released a dataset comprising 590,326 complementary patches from Sentinel1 and Sentinel2 satellites, enabling large-scale deep learning experiments on multi-spectral remote sensing images. This dataset covers approximately 10 European countries, including Austria, Belgium, Finland, Kosovo, Lithuania, Luxembourg, Portugal, Serbia, and Switzerland. The authors filtered out 125 Sentinel patches with less than 1% cloud cover. All patches were collected between June 2017 and May 2018, and the Sentinel2 tiles are Level 2A products, meaning they are orthorectified and have undergone atmospheric correction.

The BigEarthNet dataset exhibits uniform seasonal variability, with 143,557 patches from autumn, 72,877 from winter, 175,937 from spring, and 126,913 from summer. 70,987 images were excluded due to the presence of seasonal snow, clouds, or cloud shadows. Bigearthnet was designed for multi-label classification tasks, where each image is associated with one or more labels. The labeling process utilized the 2018 Corine Land Cover (CLC) maps, which produce harmonized land cover and land use maps in vector format for all European Union member states, with a reported accuracy of approximately 85%. While the original CLC nomenclature includes 44 classes, the Bigearthnet authors modified this classification system based on their understanding of relevant applications. They ultimately used 19 labels, 10 of which were directly derived from the CLC map. The remaining labels were created by grouping 22 CLC labels, while some CLC labels were discarded from the dataset.

4.3.3 Eurosat

The second downstream task in our work focuses on static land cover classification, utilizing the Eurosat dataset. Compared to Bigearthnet, Eurosat is a smaller dataset comprising 27,000 labeled and geo-referenced images. It was originally developed to assess the performance of deep convolutional networks on multi-spectral images. The primary motivation was to evaluate the efficiency of these networks on images with different textures and multiple channels, in contrast to natural images like those in Imagenet (Russakovsky et al. 2015). Eurosat consists of 64×64 pixel images uniformly distributed across 10 classes, covering approximately 34 European countries. Similar to Bigearthnet, only satellite images with low cloud coverage are included. The dataset creators claim a minimal correlation between the 10 classes. Images are uniformly distributed throughout the year to ensure balanced seasonal variability. Unlike Bigearthnet, Eurosat is a single-label, multi-class dataset, meaning each image is associated with only one annotated label. The

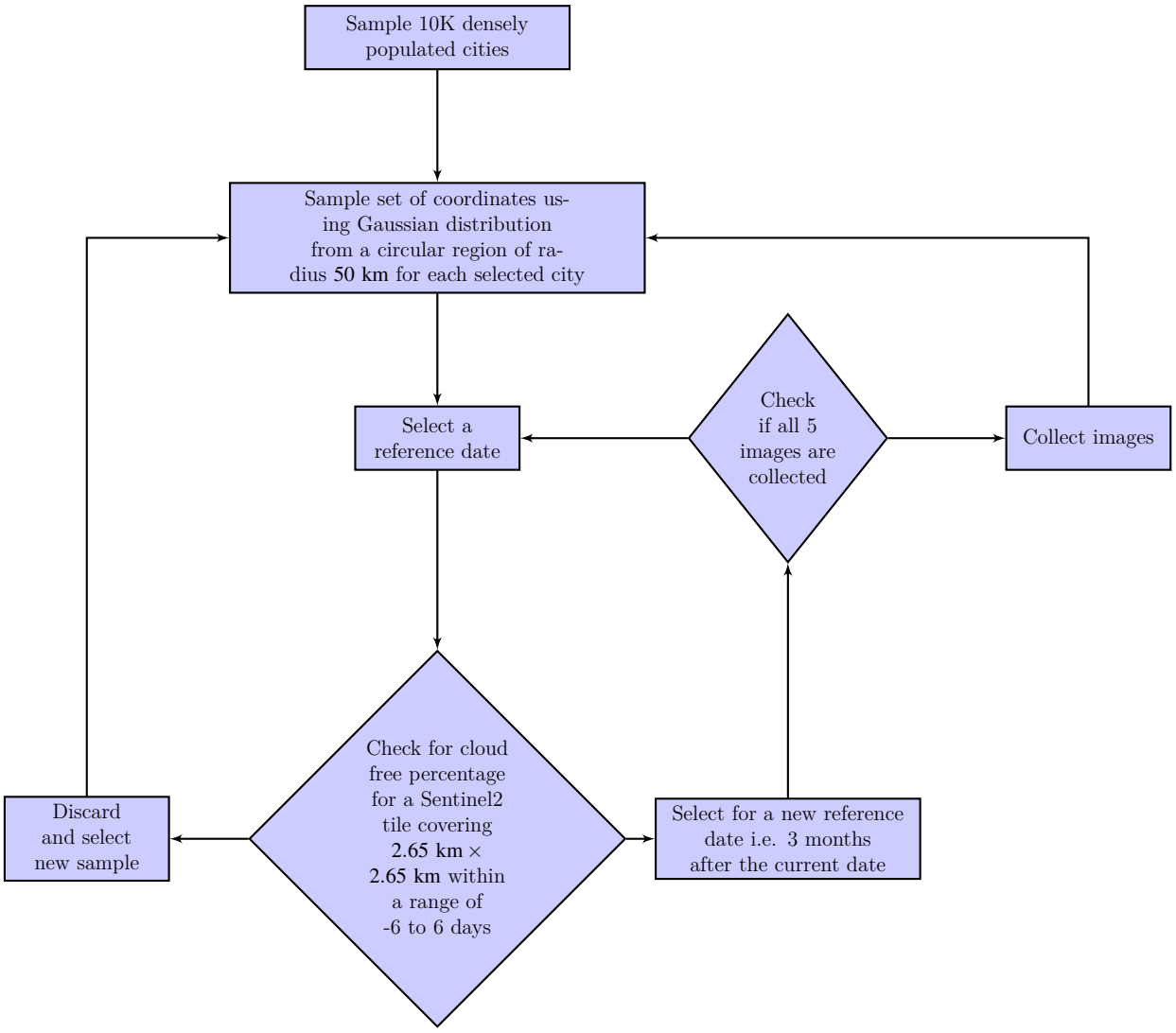


Figure 4.5: SeCo dataset collection algorithm.

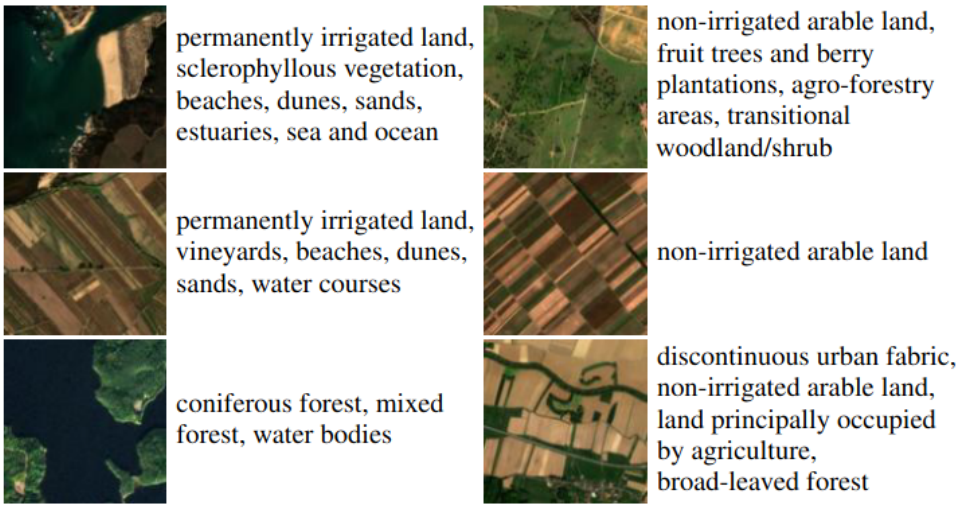


Figure 4.6: Bigearthnet dataset (Sumbul et al. 2019).

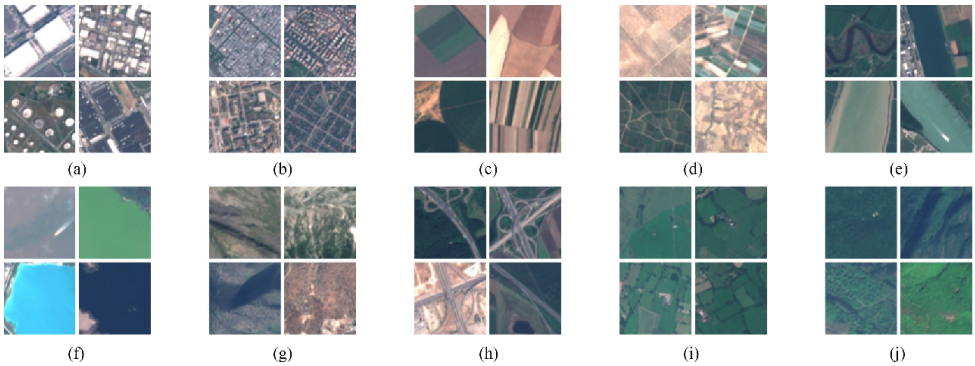


Figure 4.7: Eurosat dataset (Helber et al. 2019).

dataset underwent manual inspection, with images containing significant snow and ice coverage being discarded. It's worth noting that Sentinel2 bands have varying spatial resolutions (10 m/px, 20 m/px, and 60 m/px). To standardize the dataset, all bands with resolutions coarser than 10 m/px were interpolated to match the 10 m/px resolution. The models developed using the Eurosat dataset aim to understand physical land types and land use patterns.

4.4 Experiments

We conducted two sets of experiments. The first set of experiments was designed to show the added value of the NIR channel in land cover classification. The second set of experiments includes the pre-training setup using our proposed transformation pipeline to compare it with transformations used in MoCo-v2 (He, Fan, et al. 2019). Subsections 4.4.1 and 4.4.2 give detailed descriptions of the conducted experiments.

4.4.1 Randomly Initialized Linear Classifier Experiments

The experiments show the relevance of the NIR channel for land cover classification. We conducted the experiments with both the Eurosat and Bigearthnet datasets to evaluate the pre-trained model's performance on single-label and multi-label classification tasks respectively. Both of these are multi-spectral datasets containing the NIR channel and the three RGB channels. The experiment setup and hyperparameters used for both Eurosat and Bigearthnet are kept similar to the randomly initialized setup used in (Mañas et al. 2021). In our four channel setup, the first convolution layer is changed from three to four channels. This change increases the number of non-trainable parameters from $3 \times 64 \times 7 \times 7$ to $4 \times 64 \times 7 \times 7$. As we trained a linear classifier by freezing the weights of the backbone, the number of training parameters remained the same. Hence, the comparison of results is a fair evaluation.

Quantitative Results Table 4.1 shows the difference in the results with three and four channels for Eurosat (single-label classification) and Bigearthnet (multi-label classification).

We found an increase of 3.35% and 3.1% for Eurosat and Bigearthnet, respectively. By only training the linear classifier, these experiments show the inherent relevance of the NIR channel.

Table 4.1: Results of Experiment 4.4.1.

	Backbone	Accuracy
Eurosat (3 channels)	Resnet18	64.06
Eurosat (4 channels)		67.41
Bigearthnet (3 channels)	Resnet18	41.21
Bigearthnet (4 channels)		44.31

4.4.2 Self-supervised Learning with Atmospheric Correction

The pre-training experiments are conducted to show the quality of the representation learned by the pre-trained model. We conducted experiments on all pre-training setups mentioned in (Mañas et al. 2021) on the SeCo dataset, and compared them with our implementation of the baseline experiments. For the evaluation of the pre-training experiments, we used the same hyperparameter settings on the downstream tasks. Subsections 4.4.2.1 and 4.4.2.2 give a detailed description of the experimental settings for pre-training and downstream tasks correspondingly.

4.4.2.1 Pre-training experiments

We conducted three sets of pre-training experiments. In all experiment settings, we used the same hyperparameters adopted from (Mañas et al. 2021). We train the network for 1000 epochs.

MoCo-v2

In this experiment setup, we used a vanilla MoCo-v2 approach over images in the SeCo dataset. Baseline experiments are with MoCo-v2 and our setup is use of our proposed transformation pipeline instead of the MoCo-v2 pipeline. We used the same transformation pipeline for the query and key part of the image.

MoCo-v2 + TP In this experiment setup, we used two images of the same location. Baseline experiments are with MoCo-v2 and our setup uses our proposed transformation pipeline instead of the MoCo-v2 pipeline.

SeCo

In this experiment setup, three different embeddings are used to optimize the self-supervised loss. LOOC algorithm (Xiao et al. 2021) is applied. The first embedding uses MoCo-v2 transformation on a query image as its augmented pair. The second embedding is the same image taken at a different timestamp and doesn't undergo MoCo-v2 transformation. The third embedding is the image taken at another timestamp and is also transformed via MoCo-v2 transformation. Baseline experiments are with MoCo-v2 transformation pipeline and our setup is to replace it with our proposed transformation pipeline. Figure 4.8 illustrates the SeCo experimental setup (Mañas et al. 2021).

4.4.2.2 Downstream experiments

The experiments were conducted on two datasets. For Eurosat, a linear classifier is used on the pre-trained Resnet18 (He, X. Zhang, et al. 2015). Whereas for Bigearthnet, we evaluated our approach for both linear classifier and finetuning using two pre-trained models, i.e. Resnet18 and Resnet50 (He, X. Zhang, et al. 2015). The training-validation split and hyperparameters for downstream experiments are adopted from (Mañas et al. 2021).

Table 4.2: Classification accuracy on Eurosat dataset. The backbone is kept frozen and we only train the linear classifier.

Pre-training	Backbone	Accuracy
MoCo-v2	Resnet18	81.4
MoCo-v2 (Atmospheric)		**82.4**
MoCo-v2 + TP	Resnet18	86.2
MoCo-v2 + TP (Atmospheric)		**89.1**
SeCo	Resnet18	90.5
SeCo (Atmospheric)		**90.8**

Quantitative results Table 4.2 shows the accuracy obtained by each pre-trained model on the Eurosat dataset. The accuracy for all three experiments shows the new transfor-

Table 4.3: Mean average accuracy scores on the Bigearthnet land cover classification task.

Pre-training	Backbone	Linear Classifier		Finetuning	
		(10%)	(100%)	(10%)	(100%)
MoCo-v2	Resnet18	67.01	68.12	75.48	83.38
MoCo-v2 (Atmospheric)		**67.49**	**68.76**	**79.41**	**85.87**
MoCo-v2 (4 channels without atmospheric)		66.12	66.39	-	-
SimCLR	Resnet18	65.48	66.39	-	-
SimCLR (Atmospheric)		**72.83**	**75.19**	-	-
MoCo-v2 + TP	Resnet18	66.90	68.92	76.23	84.15
MoCo-v2 + TP (Atmospheric)		**70.58**	**73.02**	**80.72**	**86.47**
SeCo	Resnet18	68.02	70.46	77.52	84.91
SeCo (Atmospheric)		**74.40**	**75.02**	**83.00**	**87.27**
MoCo-v2	Resnet50	67.77	71.87	77.79	84.88
MoCo-v2 (Atmospheric)		**73.71**	**77.68**	**81.82**	**87.27**
MoCo-v2 + TP	Resnet50	69.87	72.21	78.27	85.13
MoCo-v2 + TP (Atmospheric)		**72.92**	**76.17**	**82.58**	**87.53**
MoCo-v2 (4 channels without atmospheric)		69.41	71.39	-	-
SeCo	Resnet50	70.10	73.26	80.46	86.51
SeCo (Atmospheric)		**76.02**	**78.60**	**83.00**	**88.03**

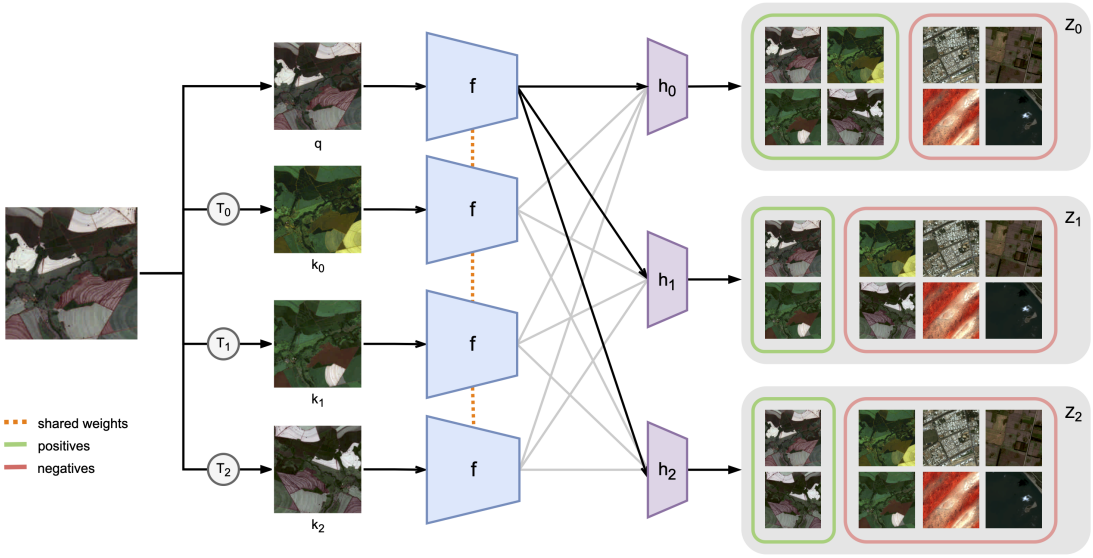


Figure 4.8: SeCo experiment setup

mation pipeline surpasses the baseline transformation. We achieved an improvement of 1.0% for MoCo-v2, 2.9% for MoCo-v2 + TP and 0.3% for SeCo.

Table 4.3 shows the accuracy obtained on Bigearthnet. We found our novel transformation to perform better than the baseline transformation irrespective of the backbone network and of the task type. We found an increase in the score for MoCo-v2 from 2.39% to 5.94%. In case of MoCo-v2 + TP, the accuracy gain lies in the range of 2.4% and 4.31%. For SeCo, it is between 1.52% and 6.38%. With our transformation, we even improved performance using the SimCLR loss (T. Chen et al. 2020). If we remove atmospheric transformation from our proposed transformation pipeline, mean average accuracy for MoCo-v2 on Bigearthnet (linear classifier) was degraded to 66.12% for 10% data and 66.39% for 100% data on the Resnet-18 backbone. With similar experiments on Resnet-50, we found the mean average accuracy degraded to 69.41% for 10% data and 71.39% for 100% of Bigearthnet data.

4.5 Conclusion

We showed a new transformation based on atmospheric correction that can be used as an alternative to color jittering in a MoCo-v2 style transformation pipeline for land cover classification based on Sentinel2 images. With this new transformation, it is possible to exploit multi-spectral bands of images and to generate physically consistent augmented samples. As we do not explicitly use the information on temporal change, this novel transformation pipeline can be extended to spatio-temporal land cover self-supervised learning applications.

From our experiments, we found that our new transformation pipeline outperforms the baseline in all the above test cases. Thus, we conclude that using atmospheric correction

as a substitute for color jittering is more effective for self-supervised pre-training on land cover images.

CHAPTER 5

Bi-Modal Contrastive Learning for Crop Classification Using Sentinel2 and Planetscope

This chapter introduces a novel approach that leverages the advantages of contrastive learning for crop classification. By employing a bi-modal contrastive learning method, we address the challenge of applying meaningful transformations to tabular data, such as pixel reflectance measurements. This bi-modal approach has gained popularity in image-text pair domains and has also shown promising results for land cover classification in remote sensing when combining optical and radar imagery. Our research demonstrates the effectiveness of using two distinct sources as different modes to measure reflection at the same location and time, each with unique properties. This approach provides diverse views for contrastive learning. The proposed method is evaluated against SCARF, a widely recognized baseline method for generating augmented views of tabular data in the machine learning community. This study includes the design of a comprehensive setup and procedure to implement an end-to-end workflow. The results show that the proposed approach outperformed the supervised and uni-modal self-supervised baseline in 8 out of 9 test cases, highlighting its efficacy in crop classification tasks.

Individual Contribution

The following chapter is based on our (Patnala et al. 2024).

Bi-modal Contrastive Learning for Crop Classification Using Sentinel2 and Planetscope
Ankit Patnala, Scarlet Stadtler, Martin G. Schultz, and Juergen Gall
Frontiers in Remote Sensing

Ankit Patnala authored the initial draft of this manuscript, which was subsequently proofread and refined by Scarlet Stadtler. Martin G. Schultz and Juergen Gall provided valuable insights through scientific discussion sessions. The initial concept originated from Ankit Patnala following discussions with external colleagues at the Institute of Geography, University of Cologne. The proposed plan was refined and polished through collaborative efforts involving Scarlet Stadtler, Martin G. Schultz, and Juergen Gall. Ankit Patnala was responsible for implementing the project and conducting the evaluation. The successful

completion of this work was made possible by the combined efforts and expertise of all team members involved.

Contents

5.1 Introduction

Crop classification is a method of identifying the type of agricultural plants at a particular location. In remote sensing, researchers typically use information from various public landcover satellite missions such as Sentinel2 (ESA 2021) and Landsat[1] which cover the entire globe at a regular time interval over many years. Crops exhibit clear temporal signatures due to phenological traits, i.e. the pattern of their growth stages from seed to sprout through budding, growing, and then ripening (Meier et al. 2009).

The availability of extensive satellite data facilitates large-scale crop mapping suitable for machine learning applications. However, traditional supervised learning methods face significant challenges. Labeling crops is time-consuming and requires skilled human effort, often limiting studies to small regions with few crop fields. Conventional methods like random forests generate good results for specific fields but fail to generalize across different geographical properties (Račič et al. 2020) or at different time periods (Hütt, Waldhoff, and Bareth 2020a). Unsupervised learning algorithms such as K-means clustering and K-Nearest Neighbor (KNN) do not require labels but are only effective for low-dimensional data, making them less suitable for high-dimensional remote sensing time series. This

[1] https://landsat.gsfc.nasa.gov/appendix/references/

limitation has led to the development of advanced deep learning techniques, particularly self-supervised learning.

Self-supervised learning enables pre-training of models using large amounts of unlabeled data, with subsequent transfer learning for related tasks with limited annotations. This approach has shown improvements over randomly initialized models (Yang et al. 2020). Among self-supervised methods, contrastive learning (X. Liu et al. 2021) has demonstrated promising results. Contrastive learning aims to align outputs from different viewpoints of the data sample while pushing away outputs from other data samples. It typically uses alternative loss functions like InfoNCE (Oord, Y. Li, and Vinyals 2018) to avoid trivial solutions. The method relies on data augmentation to maximize mutual information shared between a sample and its augmented version.

However, applying contrastive learning to remote sensing time series data poses unique challenges. Standard image transformations assume static images covering large spatial neighborhoods, which is unsuitable for crop analysis characterized by small field sizes and significant temporal changes. Moreover, the lack of field boundary information makes it harder to adapt the existing self-supervised approach for crops. This can be overcome by using spectral information of individual pixels instead of relying on crop field boundaries. This approach is justified because the variance of spectral patterns among pixels belonging to one field is quite low. The variance distribution plots of spectral measurements for four field parcels can be found in the Figure 5.1.

To address the challenges of augmentation for remote sensing time-series data, we propose a novel bi-modal contrastive learning approach (X. Yuan et al. 2021). Instead of relying on standard augmentation techniques, our method obtains the augmented version of Sentinel2 data directly from another source, specifically Planetscope. This innovative strategy serves a dual purpose: it not only provides an alternative to traditional augmentation but also combines the complementary strengths of both data sources—Sentinel2's superior spectral resolution and Planetscope's finer spatial resolution. Further, it enables the development of a bi-modal self-supervised pre-trained model that can be applied even when only one data source is available for downstream tasks.

In this work, we designed a strategy to develop a bi-modal self-supervised pre-trained model, thus combining the higher spectral resolution of Sentinel2 with the finer spatial resolution of Planetscope. To evaluate this, we demonstrate crop classification using only Sentinel2 data as our downstream tasks. In this work, we demonstrate with our experiment setup that the proposed bi-modal contrastive self-supervised pre-training improves crop classification accuracy compared to the uni-modal contrastive self-supervised model. In the following Section 5.2, the images of a few crops and their corresponding ndvi time series plot is shown.

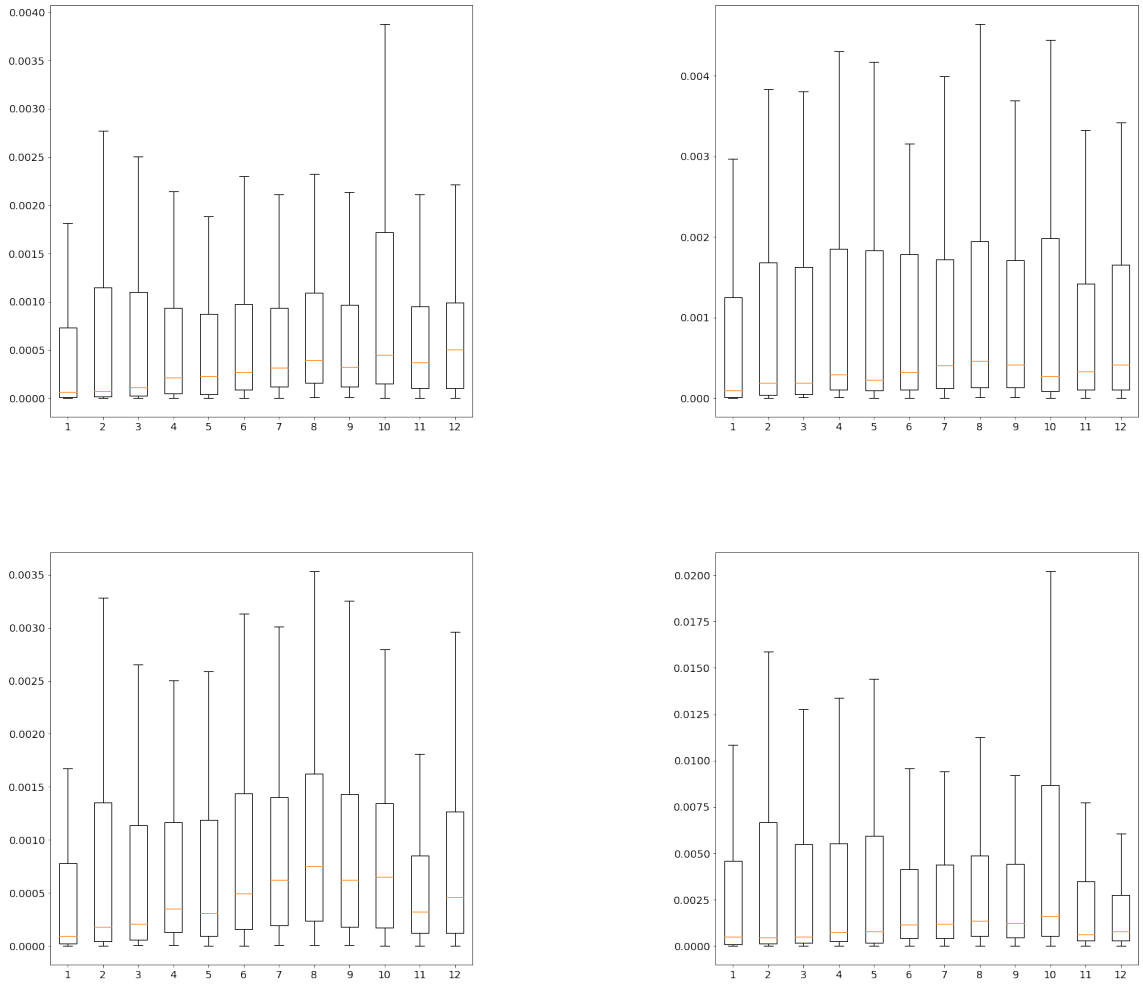


Figure 5.1: **Variance plot within a field parcel.** Four randomly selected field parcels are evaluated. The spectral measurements ranging from 0 to 1 are analyzed across 144 Sentinel2 time stamps. The box plot illustrates the variance distribution per spectral band for each field parcel for the 144 time stamps. A similar pattern is observed in other field parcels. The mean of the distribution falls within an acceptable range, indicating lower spatial variability.

5.2 Crop Images and NDVI Plots

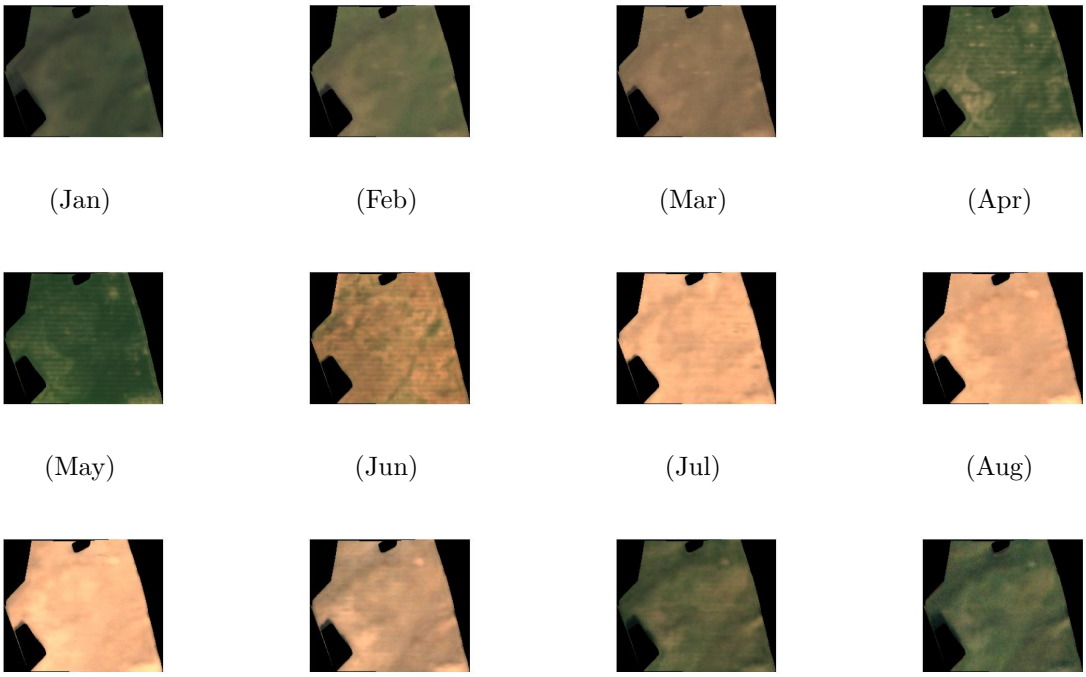


Figure 5.2: Images of a sample barley crop field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

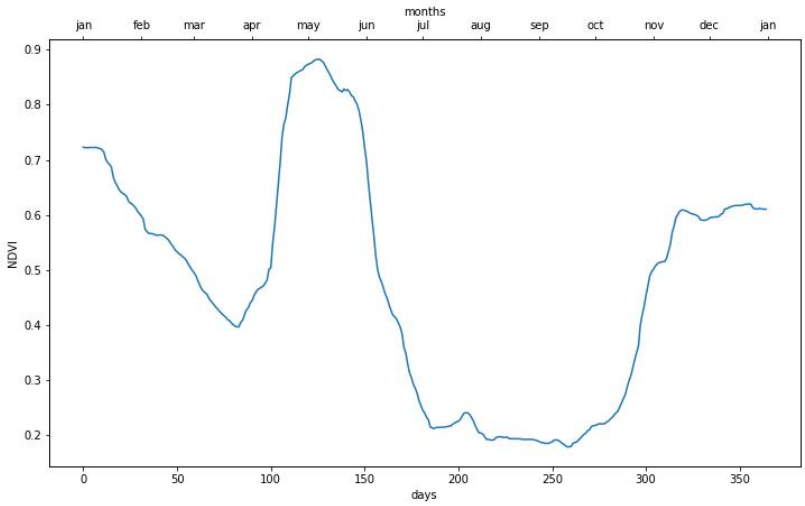


Figure 5.3: Ndvi plot of the sample barley crop field.

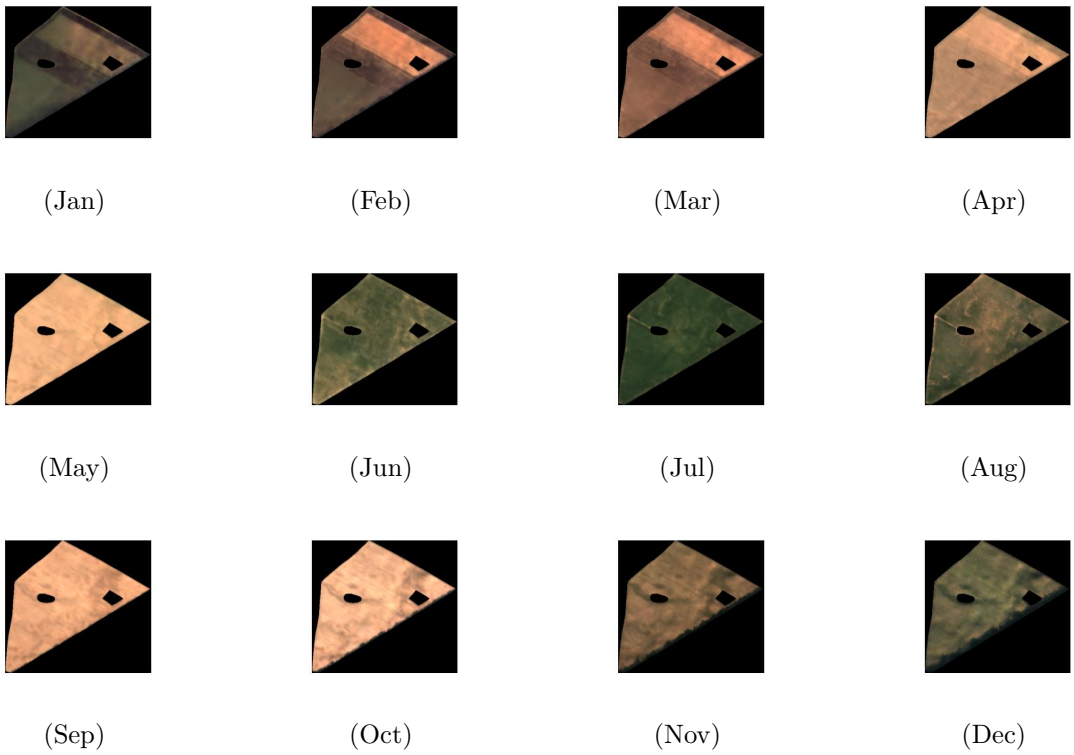


Figure 5.4: Images of a sample corn crop field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

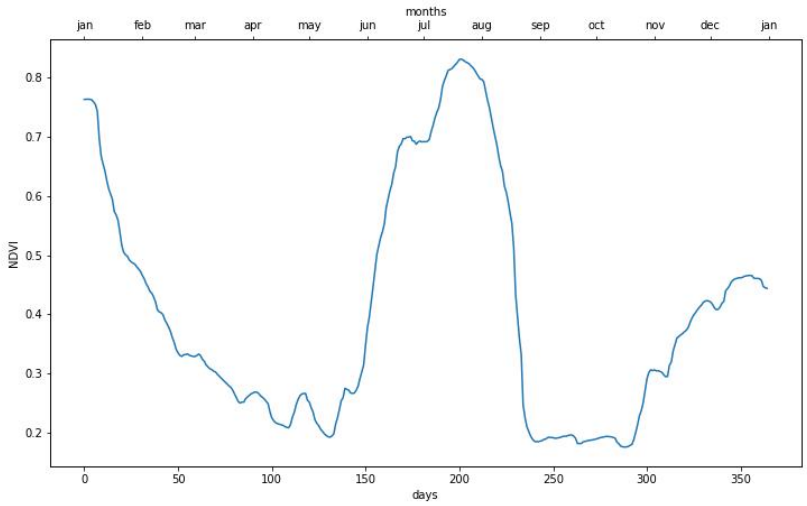


Figure 5.5: Ndvi plot of the sample corn crop field.

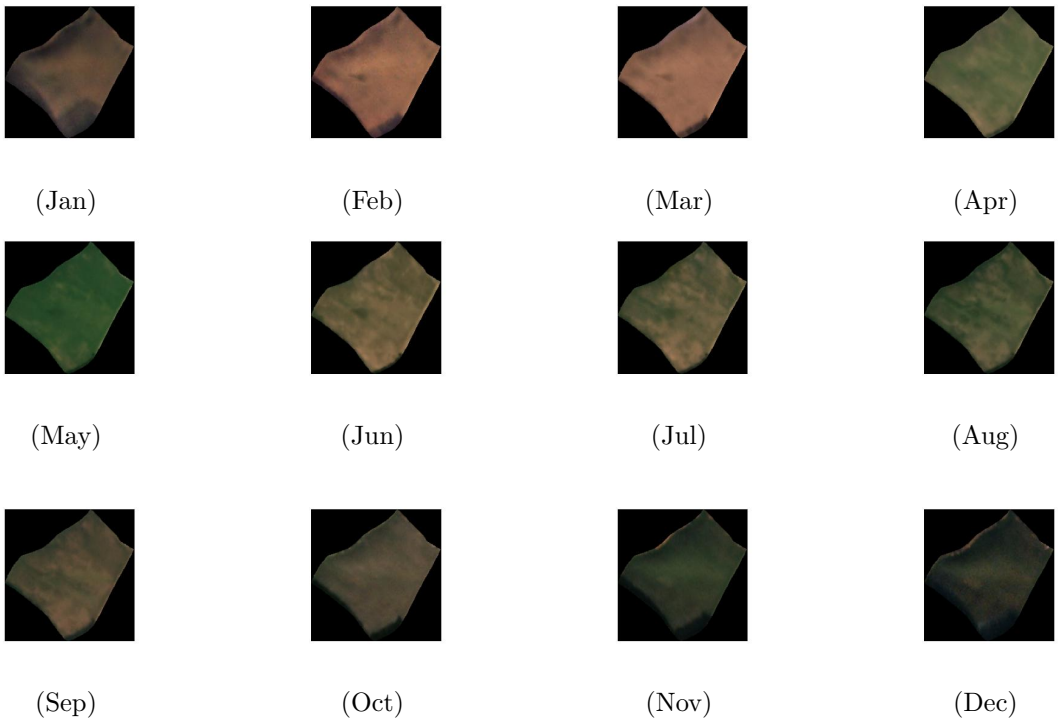


Figure 5.6: Images of a sample forage crops field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

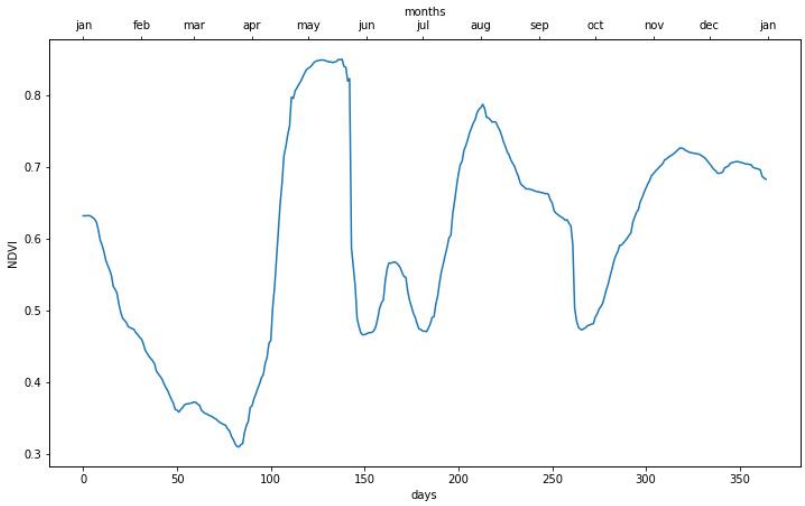


Figure 5.7: Ndvi plot of the sample forage crops field.

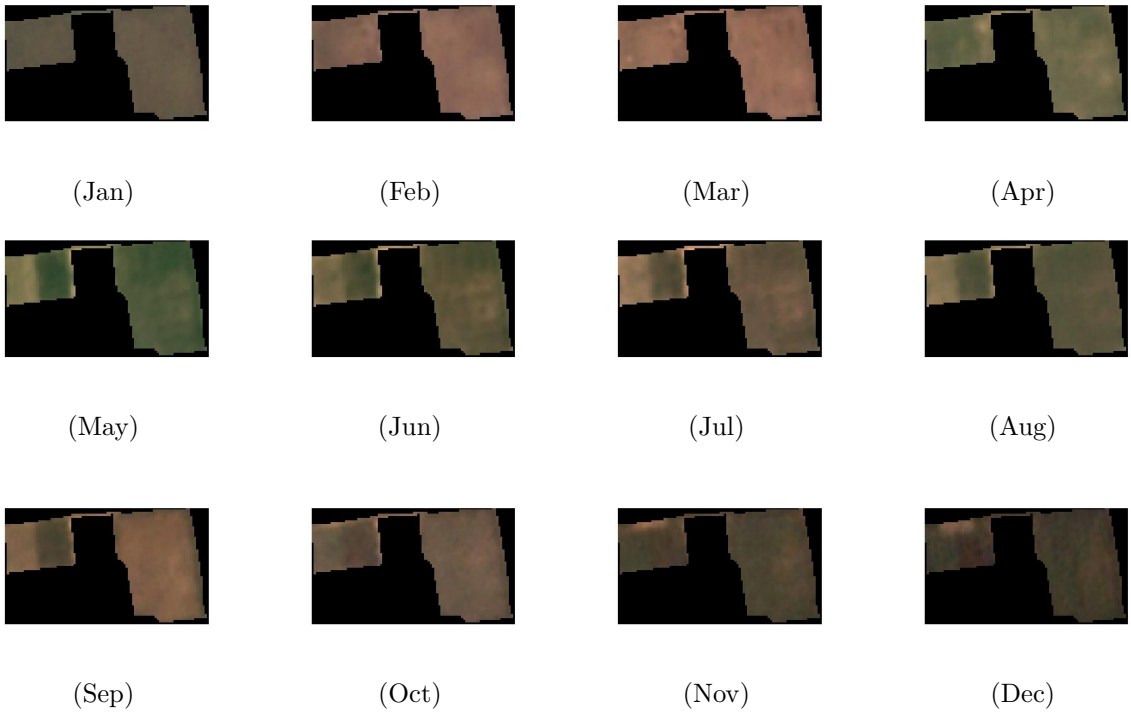


Figure 5.8: Images of a sample meadows field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

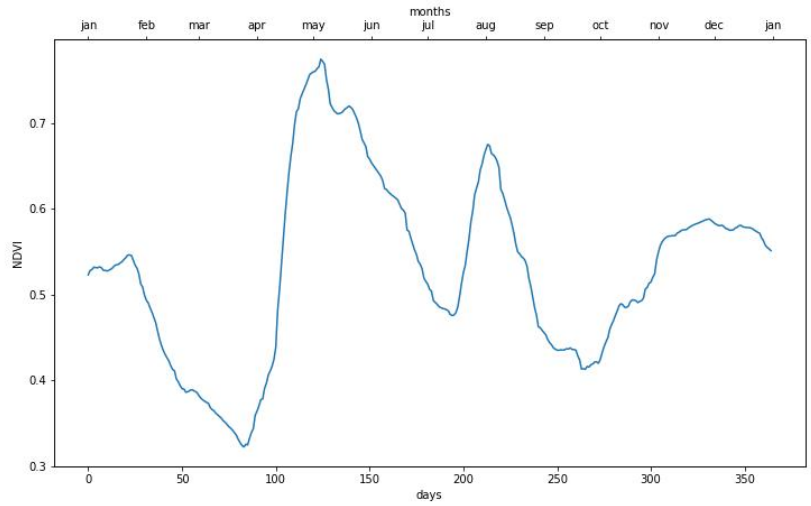


Figure 5.9: Ndvi plot of the sample meadows field.

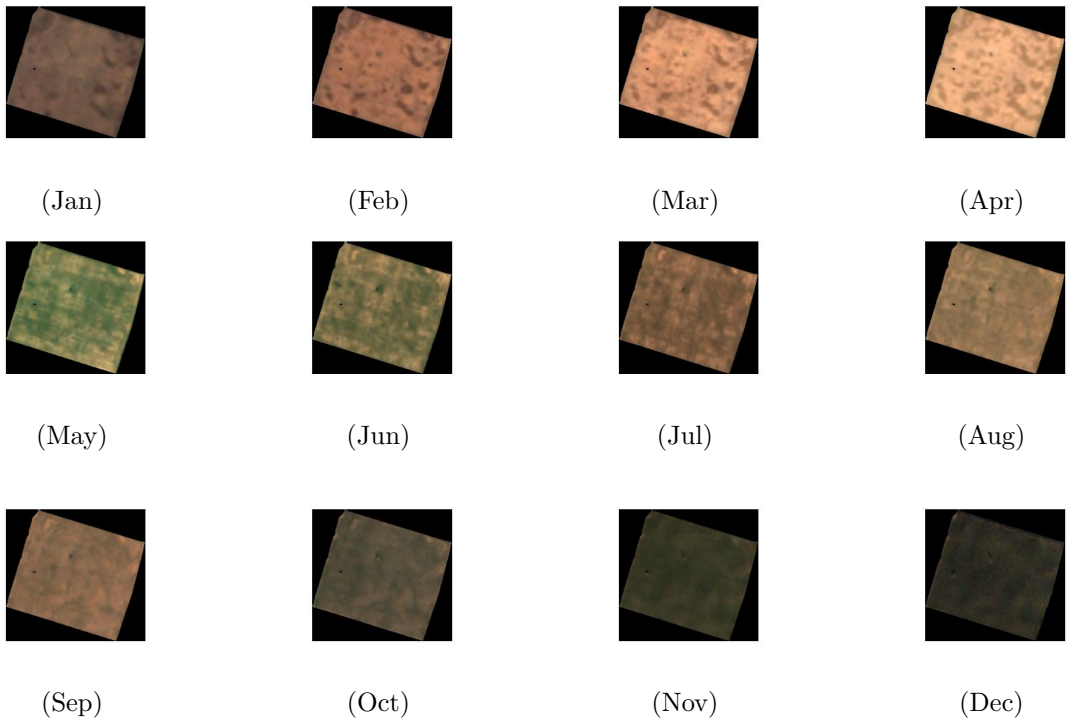


Figure 5.10: Images of a sample oats field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

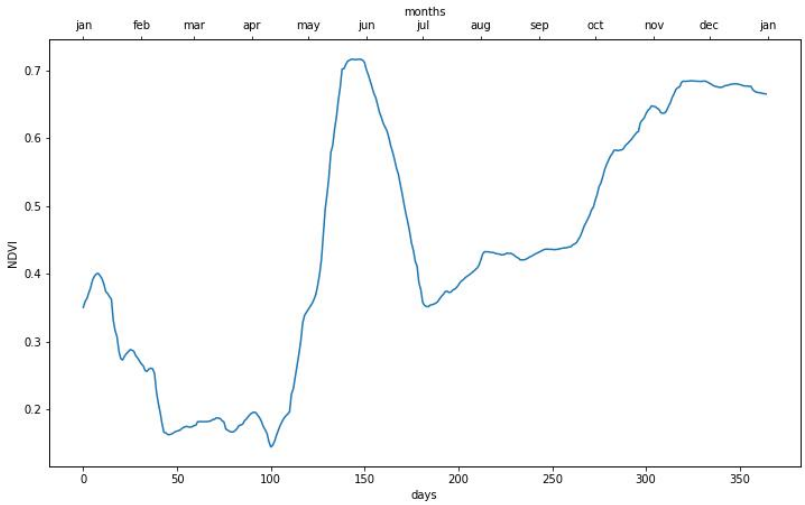


Figure 5.11: Ndvi plot of the sample oats field.

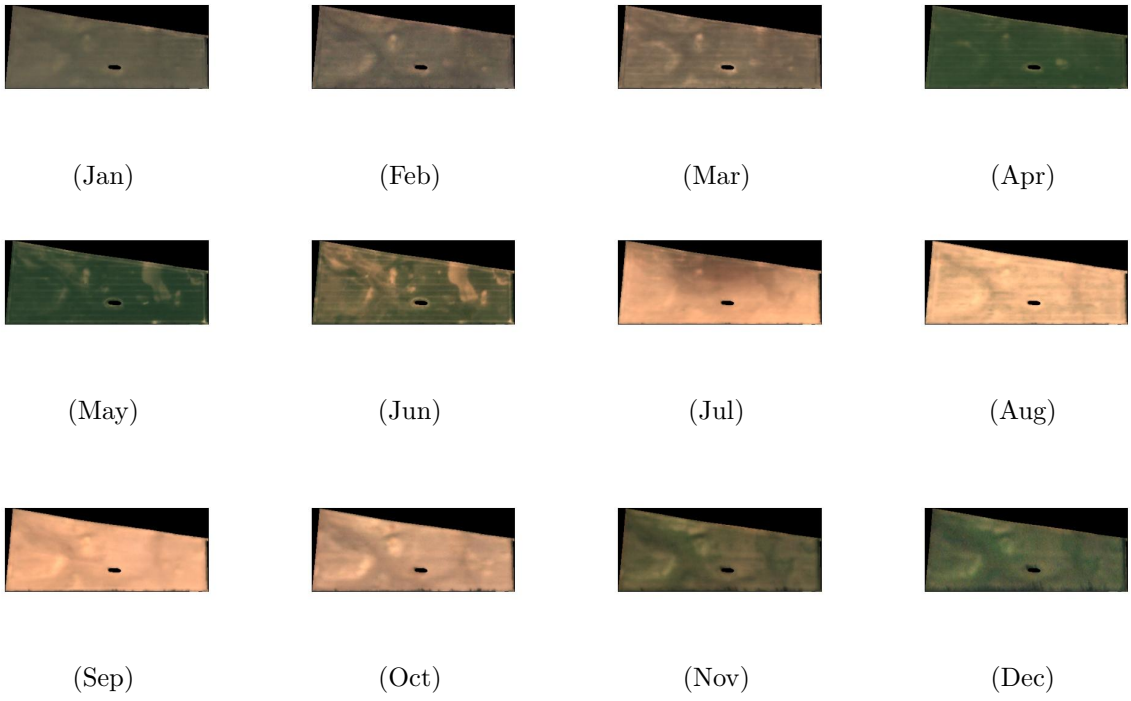


Figure 5.12: Images of a sample rye field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

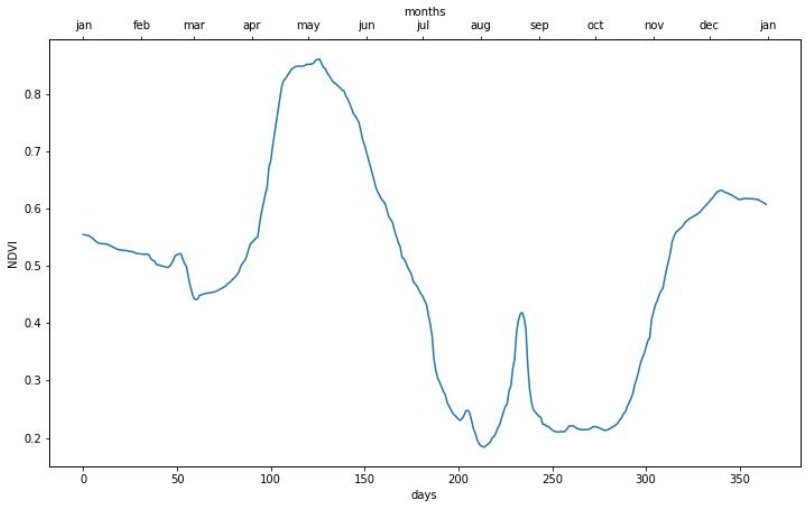


Figure 5.13: Ndvi plot of the sample rye field.

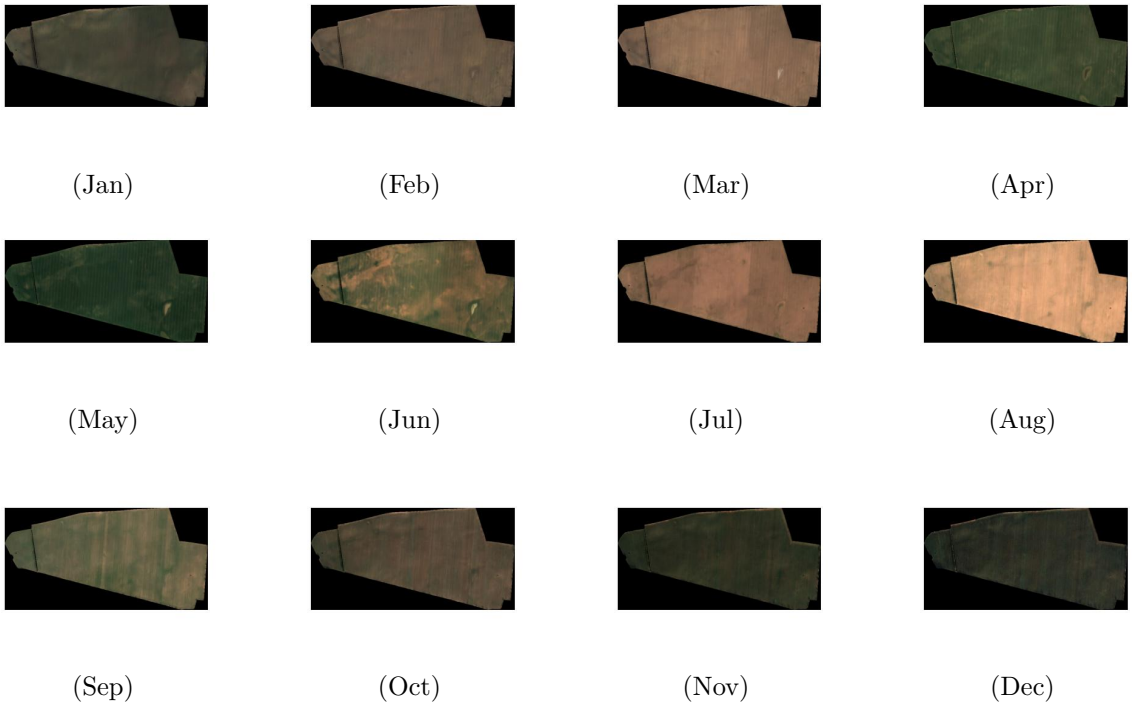


Figure 5.14: Images of a sample wheat field taken from Planetscope satellite missions. Images are taken on the 15th day of each month.

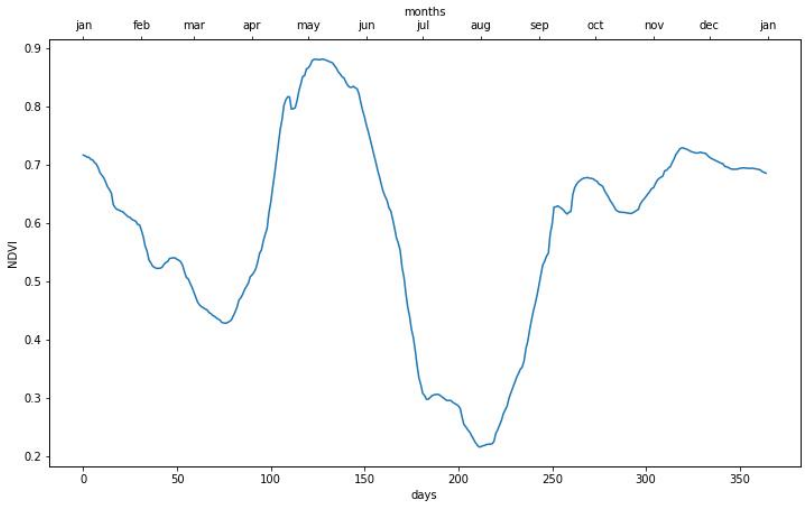


Figure 5.15: Ndvi plot of the sample wheat field.

5.3 Methods

5.3.1 Bi-modal Self-Supervised Learning

Figure 5.16 illustrates the experiment setup of our bi-modal contrastive learning method. In this approach, the networks are not shared between the two modalities due to the different input dimensions of both sources. Here, the backbone network is denoted by $E_s : \mathbb{R}^{12} \to \mathbb{R}^{256}$ and $E_p : \mathbb{R}^{36} \to \mathbb{R}^{256}$ for data from Sentinel2 and Planetscope, respectively. Similarly, the projector network is denoted by $P_s : \mathbb{R}^{256} \to \mathbb{R}^{256}$ and $P_p : \mathbb{R}^{256} \to \mathbb{R}^{256}$ for Sentinel2 and Planetscope, respectively. Here, 12 refers to 12 spectral bands of Sentinel2, and 36 refers to 4 spectral bands of 3×3 Planetscope pixels flattened to a 36-dimensional vector. Equation (5.1) provides a mathematical formulation of the SimCLR loss function (T. Chen et al. 2020) used in our work. For the pre-training, we adapt the SimCLR loss to our bi-modal setup:

$$l_{x_{is},x_{ip}} = -\log \frac{r_{iisp}}{\sum_{k=1,k\neq i}^{N} r_{ikss} + \sum_{m=1,m\neq i}^{N} r_{imsp}} \tag{5.1a}$$

where

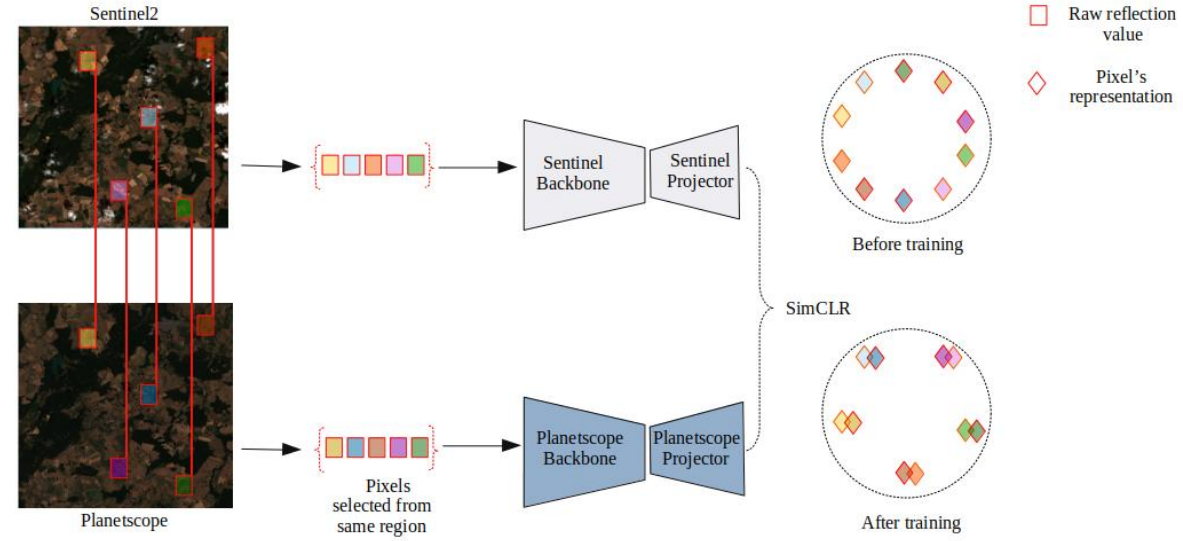
$$r_{ijsp} = \exp\left(\frac{sim(z_{is}, z_{jp})}{\tau}\right) \tag{5.1b}$$

and

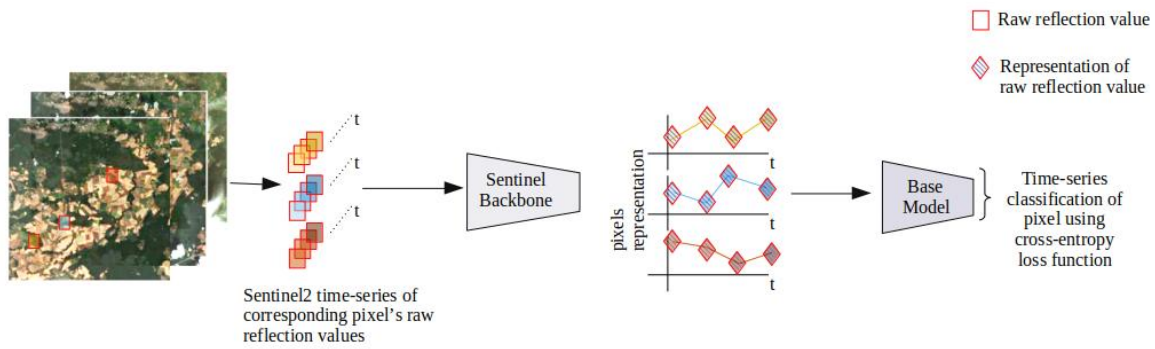
$$sim(z_{is}, z_{jp}) = \frac{z_{is}^T z_{jp}}{\|z_{is}\| \left\|z_{jp}\right\|}. \tag{5.1c}$$

Here, x_{is} represents the i^{th} Sentinel2 data sample, and z_{is} represents the output obtained after passing through the encoder and projector components of the Sentinel2 network. Similarly, x_{ip} represents the i^{th} Planetscope data sample, and z_{ip} represents the output obtained after passing through the encoder and projector components of the Planetscope network. The parameter τ denotes the temperature that controls the sensitivity of the loss function. In the original SimCLR equation (T. Chen et al. 2020), there is only one network and two augmented views share the same model. In contrast, in our bi-modal case, there are separate networks for different views. The term r_{ikss} in the denominator of Equation (5.1a) denotes the cosine distance between a Sentinel2 data sample to other Sentinel2 data samples in the batch, while r_{imsp} denotes the cosine distance between the Sentinel2 data sample and the other Planetscope data samples in the batch. Figure 5.16b illustrates how to use the pre-trained Sentinel2 backbone for the downstream task of crop classification. For each pixel, 144 timestamps are passed through the pre-trained model to obtain an abstract pixel representation. The time series formed with the representations of each timestamp serves as an input to the base models.

As bi-modal pre-training implicitly learns a mapping from Planetscope to Sentinel2 data, it is sufficient to input only Sentinel2 data into the model for the downstream



(a)



(b)

Figure 5.16: **Schematic setup of our bi-modal self-supervised experiment.** a) The top part illustrates the bi-modal setup, where corresponding pixels indicated by red bordered colored boxes are randomly selected. Sentinel2 and Planetscope data are passed through their respective backbone and projector networks. The outputs are aligned by optimizing the contrastive loss to attract similar and repel dissimilar pairs. Two spherical diagrams on the right side show how the projection of the data is randomly distributed before and after training. The similar pairs get aligned simultaneously maintaining uniformity in the latent hypersphere space. A sample of pixel collection is shown for one timestamp as an example. Similarly, data for pre-training are collected from other timestamps. b) The bottom part shows the training of a downstream task, where raw Sentinel2 data is fed through the pre-trained Sentinel2 backbone. The output is then fed to different base models for conventional supervised learning, optimizing the standard cross entropy loss for multi-class classification.

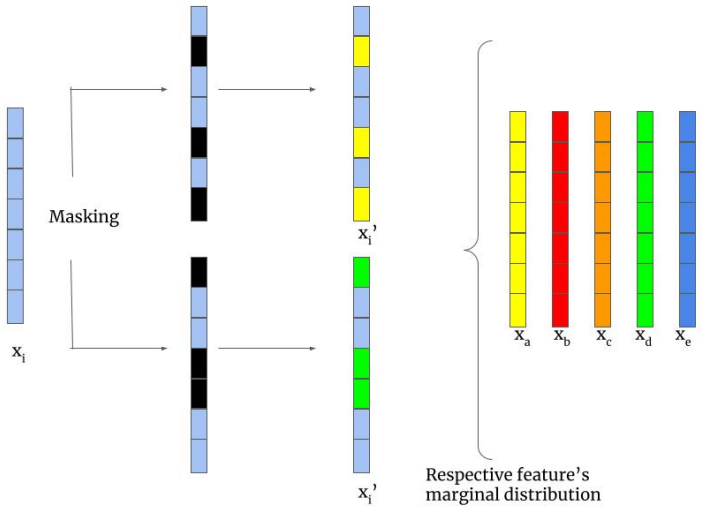


Figure 5.17: **Random feature corruption mechanism in SCARF** (Bahri et al. 2021). Entries of a feature vector x_i are masked, indicated by black cells. These masked features are then replaced by features from other samples in the batch, denoted by x_a, ..., x_e. This results in two corrupted feature vectors (x_i'). The two corrupted feature vectors build a positive pair.

classification task. Thereby, users can implicitly take advantage of Planetscope's finer spatial resolution.

We employ the random feature corruption technique from SCARF (Bahri et al. 2021) as a transformation on both sources in our bi-modal self-supervised learning setup, illustrated in Figure 5.16a. In random feature corruption, with a given corruption rate c, randomly c% of the features in the data are replaced by the empirical marginal distribution of the corresponding features. Figure 5.17 provides a schematic diagram of the random feature corruption technique.

For tabular data, MLPs are the most common choice. We enhance this approach by implementing ResMLP as our backbone, inspired by the skip connection mechanism introduced in ResNets (He, X. Zhang, et al. 2015). These skip connections facilitate deeper gradient flow, addressing vanishing and exploding gradient issues. Figure 5.20 illustrates the structural differences between a conventional MLP and our ResMLP. In the ResMLP architecture, each block incorporates a skip connection (depicted by the orange overhead arrow) that links the non-linear output back to its input, enabling more efficient information flow through the network. Our implementation consists of an 8-layer network with approximately 550K parameters.

5.3.2 Downstream Tasks

For the crop classification downstream tasks, we use the representation obtained from the pre-trained model. The downstream tasks, each slightly different from the others, allow us to test the generalizability of the pre-trained model. The first task involves data from the same region (Brandenburg) as the pre-training data. The second task uses data from

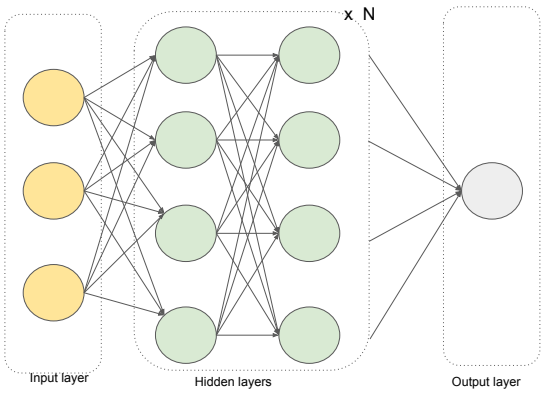


Figure 5.18: MLP architecture

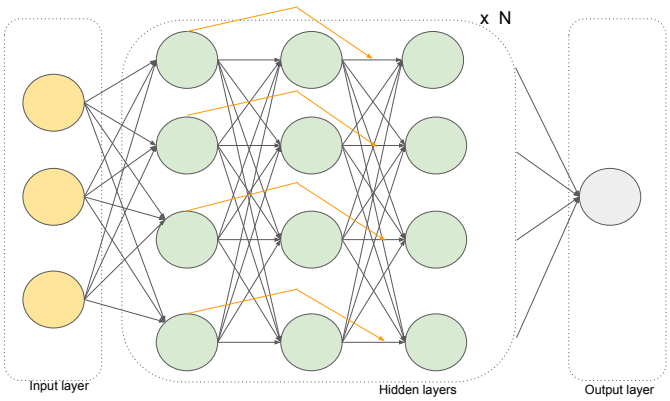


Figure 5.19: ResMLP architecture

Figure 5.20: Difference in the network architecture of MLP and ResMLP.

a different area in Brandenburg but from a distinct year. The third task encompasses data from the Brittany region in France. In these downstream tasks, the time series of pre-trained features is fed to a temporal network for classification. We use 3 standard deep learning architectures: Bi-LSTM (Cornegruta et al. 2016), inception time (Fawaz et al. 2019), and position encoded transformers (Vaswani et al. 2017).

5.4 Datasets

In this work, we use Sentinel2 and Planetscope as two different sources for bi-modal contrastive learning. Sentinel2 is an ESA satellite mission. Its multi-spectral instrument (MSI) consists of 12 bands, spanning from visible to thermal and infrared bands (400 nm to 2190 nm). For Sentinel2 with a spatial resolution of 10 m, each pixel covers an area of 100 m^2. Data are publicly available and can be accessed either through the Copernicus API or Google Earth Engine (Gorelick et al. 2017a). Despite the availability of cloud masks, obtaining cloud-free images for a particular region can be challenging. In contrast, Planetscope, a commercial satellite mission, provides higher pixel resolution at 3 m/px. The instrument takes multiple snapshots of a particular region and uses the "best scene on top" algorithm[2]. Planetscope ensures images with minimal cloud, haze, and other disturbances. However, it has a limitation in spectral resolution, providing only 4 channels (R, G, B, and NIR).

In this work, we utilize the training and validation sts of the DENETHOR dataset (Kondmann et al. 2021) to create a custom dataset for bi-modal self-supervised learning experiments. DENETHOR is a publicly available crop type classification dataset that provides high-resolution remote sensing data from Planetscope, Sentinel2, and Sentinel1. It is developed for near real-time monitoring of agricultural growth cycles in Northern Germany. By leveraging multiple satellite sources, DENETHOR enhances the data for accurate crop classification. DENETHOR provides both training and validation sets. The dataset provides both training and validation sets, with the latter being at different temporal and geographical location, thereby allowing the development of models that are robust to such temporal and spatial variations.

We use the training and validation sets of the DENETHOR dataset (Kondmann et al. 2021) to construct a custom dataset for conducting bi-modal self-supervised learning experiments.

DENETHOR's training dataset covers a 24×24 km^2 region in the state of Brandenburg, Germany. The dataset includes data from both Sentinel2 and Planetscope. The training data covers the entire year 2018. As the pixel resolution of Sentinel2 is 10 m/px and 3 m/px for Planetscope, the dimensions of the measurement scenes are represented as 2400×2400 and 8000×8000, respectively. For both Sentinel2 and Planetscope, we use the same 144 timestamps for a given year. DENETHOR's validation dataset also covers a 24×24 km^2 in Brandenburg, but from a different region. Furthermore, the validation dataset is from 2019.

[2]https://developers.Planet.com/docs/data/visual-basemaps/

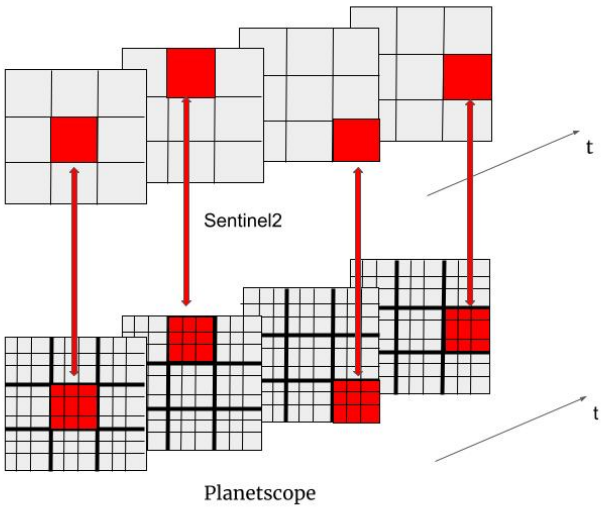


Figure 5.21: **Description of data alignment for bi-modal self-supervised learning.** The top and the bottom rows display a spatial region of the same geographical area captured by Sentinel2 and Planetscope, respectively. For each timestamp, pixels are randomly selected from Sentinel2. The pixel data is aligned to 3×3 pixels from the same corresponding region for the same timestamp from Planetscope.

The training set comprises 2534 field parcels, while the validation set comprises 2064 field parcels. They are distributed across 9 different crop types. In both sets, there are locations where no crops are grown, and the pixels associated with these locations are masked.

The use of multiple downstream tasks serves to evaluate the generalizability of a pre-trained model. This is done by evaluation of test data from a different time and region. Figure 5.22 shows a visual representation of our splitting strategy. Subsection 5.4.1 details the generation of the pre-training data and Subsection 5.4.2 offers an overview of the data generated for the downstream tasks.

5.4.1 Data for Pre-training

For pre-training, we utilize unlabeled data. To acquire this, we use 70% of the random crop fields following the 70-21-9 split. We do not use the crop labels. For our pre-training dataset, we iterate through each of the 144 Sentinel2 timestamps and randomly select $100,000$ pixels from the 70% split, resulting in $14,400,000$ data samples for our bi-modal self-supervised experiment. Since the samples are randomly chosen, the pre-training dataset is not balanced. The $14,400,000$ data samples are independent. So, the negatives for a Sentinel2 pixel include other pixels as well as the same pixel in the same location at a different time stamp. Given one pixel of Sentinel2 covers $100\ \mathrm{m}^2$ while a Planetscope pixel covers $9\ \mathrm{m}^2$, we align a Sentinel2 pixel to 3×3 pixels of Planetscope, as illustrated in Figure 5.21.

5.4.2 Data for Downstream Tasks

The pre-trained model is tested on three different sets of Sentinel2 data (two from DENETHOR and one from Breizhcrop) to assess its generalizability. We establish two crop classification downstream tasks using DENETHOR's training and validation dataset. For downstream task 1, we use 21% and 9% of the data, as per the given 70-21-9 split of DENETHOR's training dataset, to separate training and validation field parcels. To ensure a balanced dataset, 5000 pixels are randomly selected for each of the 9 crop types from their field parcels. Similarly, for the validation set of downstream task 1, we create a balanced dataset by randomly selecting 1000 pixels for each crop from the validation field parcels. A 70-30 split is applied on DENETHOR's validation set to separate training and validation field parcels for downstream task 2. We follow a similar procedure to generate a balanced dataset for our second crop classification downstream task. Figure 5.22 visually illustrates the two downstream tasks.

The downstream task 3 is added to assess the performance of the pre-trained model in a region located further away from the region used for pre-training. We use a subset of the Breizhcrop dataset (Rußwurm, Lefèvre, and Körner 2019), containing 2018 data from the Brittany region in France. The dataset provides aggregated spectral measurements per field parcel. We specifically use data from field parcels with spectral measurements for more than 142 time stamps. There are 9 crop types in the original dataset (permanent meadows, temporary meadows, corn, wheat, rapeseed, barley, orchards, sunflower, and nuts). We discard data from orchards, sunflowers, and nuts as there are fewer field parcels for these crop types. We create a balanced dataset from the remaining crop. For training subsets, we collect 9000 data samples from each of the six crop types, and for the validation subset, we collect 1000 data samples. The final task is crop classification with 54000 training samples and 6000 validation samples.

5.5 Experiments

To evaluate the performance of our new bi-modal, self-supervised, contrastive method, we conducted supervised experiments and uni-modal self-supervised experiments as competitors. All experiments were performed on a 32GB NVIDIA Tesla V100 GPU.

5.5.1 Supervised Experiments

We use 10 different models for each category of base models. To obtain these 10 models, we define the range or definite sets for each hyperparameter. We randomly generated 10 different models for each category. This random generation of 10 models for each category is done using Optuna hyperparameter tuner (Akiba et al. 2019) on a defined hyperparameter grid. It is important to note that, in this case, we do not use the tuner to find the best model rather we use all 10 models for our analysis.

For bi-directional LSTM, the hyperparameter space is defined as follows: dimensions of the hidden layer as one of [32,64,128,256], number of layers between 2 and 6, and learning

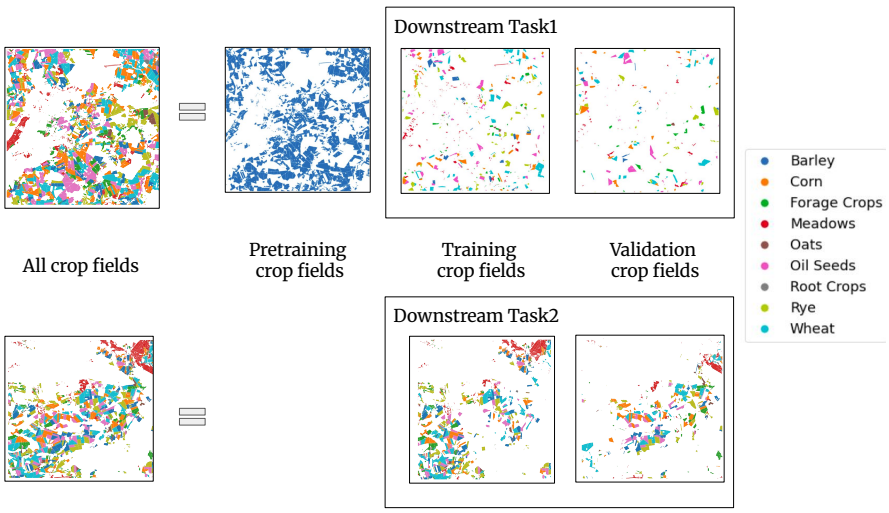


Figure 5.22: **Dataset for the bi-modal self-supervised learning experiment.** The top row of the figure illustrates the splitting applied to DENETHOR's training dataset. The top image shows all crop fields in the training dataset, divided into three parts. The first part is used for pre-training and is represented by blue crop fields. The blue color represents unlabeled data. The other two parts are training and validation data for our downstream task 1. The bottom row shows the validation dataset for DENETHOR. In the case of downstream task 2, we split the validation data into training and validation for our downstream task 2. The downstream task 2 is trained using the pre-trained model obtained from the same pre-training dataset, allowing us to assess its performance when region and time period are changed. We refer to (Rußwurm, Lefèvre, and Körner 2019) for the dataset of downstream task 3.

rate in the range from 10^{-5} to 10^{-3}. For inception time, the hyperparameter space is specified as follows: number of layers as either 2, 4, or 8, dimension of hidden layer as one of [128,256,512,1024], kernel size as one of [40,80,120,136], and learning rate between 10^{-5} and 10^{-3}. The hyperparameter space for transformers is defined as follows: the dimension of the model is either 32, 64, or 128, the number of attention heads as one of [2,4,8], the number of layers between 2 and 6, and the learning rate ranges between 10^{-5} and 10^{-3}. In all supervised experiments, we train the network for 20 epochs. We use the initial learning rate of 10^{-3} with the linear scheduler.

5.5.2 Uni-Modal Self-Supervised Experiments

This is our second set of experiments. With these experiments, we intend to compare our proposed bi-modal self-supervised models to other self-supervised model.

In this experiment setup, we use uni-modal contrastive learning, employing only Sentinel2 data during pre-training. The absence of transformation processes such as cropping, and color jittering for tabular data makes it difficult to obtain augmented samples for Sentinel2. Therefore, we use the random feature corruption technique SCARF (Bahri et al. 2021) to facilitate contrastive learning for tabular data with a single source. The experiment setup is illustrated in the Figure 5.23. In our uni-modal self-supervised experiment

setup, we obtain the pre-trained model by applying contrastive learning on pre-training data. We run the pre-training for 100 epochs. We use a SimCLR loss function with a temperature of 0.07. The learning rate is set to 10^{-3}. Given that a contrastive loss requires a higher batch size to generalize effectively, we opt for a batch size of 2048. We obtain two pre-trained models, one with a random feature corruption rate of 20% and the other with 60%.

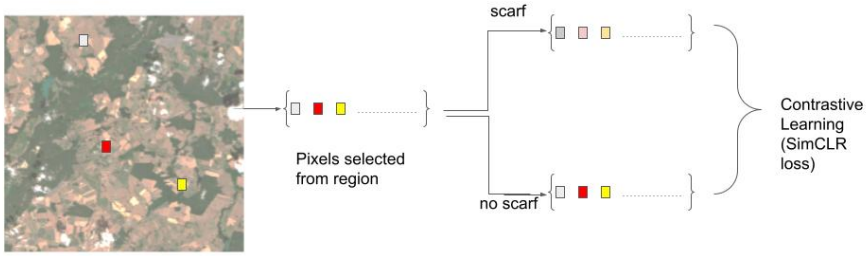


Figure 5.23: Uni-modal contrastive learning experiment setup.

5.5.3 Bi-Modal Self-Supervised Experiments

This is the new experimental setup proposed in this study. In contrast to the uni-modal setup, the bi-modal self-supervised model uses data from two sources i.e. Sentinel2 and Planetscope, to obtain pairs of matching samples. The Sentinel2 and Planetscope data are processed with different networks for pre-training and the pretext task consists of aligning the representation obtained from spectral signatures of both input data streams. We run the pre-training for 100 epochs. Similar to the uni-modal self-supervised experiment setup, we set the initial learning to 10^{-3}. The temperature parameter for the bi-modal SimCLR loss is set to 0.07. In addition, we use the random feature corruption for each source. Similar to the uni-modal setting, we pre-train two models, one with a corruption rate of 20% and another one with 60%. We also run experiments without corruption, but the models with feature corruption yielded better results. Therefore, we are reporting the results for models pre-trained with 20% and 60% corruption rates.

5.6 Results

We adopt the evaluation protocol from SCARF (Bahri et al. 2021), which involves a win-matrix plot and a box plot to compare different models on a number of test datasets. In the win-matrix plot, each cell's value represents the ratio of experiments mentioned in the row outperforming the one in the column, as formulated in Equation (5.2); where i and j are competing methods, and N is the total number of experiments.

$$W_{ij} = \frac{\sum_{i=1}^{N} \mathbb{I}(val\_acc_i > val\_acc_j)}{N} \tag{5.2}$$

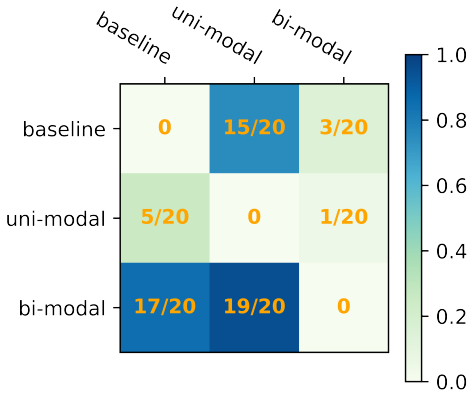
We provide separate results for each downstream task, as well as a distinct evaluation for the three base models. To evaluate the performance, we compare bi-modal self-supervised against uni-modal pre-trained models on the same corruption rate with corruption rates of 20% and 60%, respectively.

For each downstream task, we use 10 different models from each category with varying hyperparameters. For supervised learning, we report the mean scores for these 10 models. To evaluate the pre-trained model, we fed the representation obtained from our pre-trained model to these 10 models with the same hyperparameters. We report the mean relative gain of the self-supervised model over the corresponding supervised model with the same architecture and training parameters. We show our win-matrix and relative gain plot for 20 scores (10 results each for corruption coefficient of 20% and 60%, respectively). We report the mean gain (min and max value inside the parenthesis) for the self-supervised models. In some cases, the uni-modal self-supervised models show inferior results compared to the supervised and bi-modal methods. These are shown with negative values.

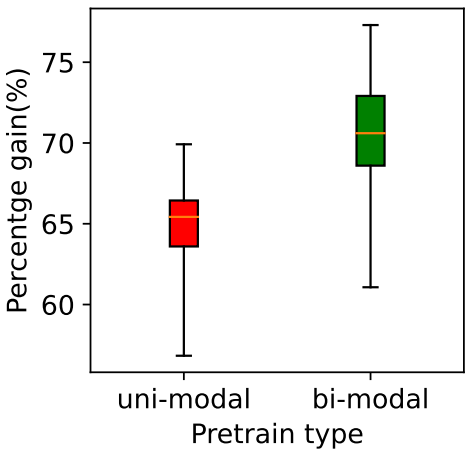
Figure 5.24 shows the win-matrix and relative gain box plot for ResMLP pre-trained models on downstream task 1. The bi-modal self-supervised model outperforms uni-modal self-supervised and supervised models. The random feature corruption technique, which demonstrated improved results on OPENML tabular data (Bischl et al. 2021) in the case of uni-modal contrastive learning pre-trained models, does not yield promising results for time-series crop classification data. Upon comparing the bi-modal self-supervised ResMLP model with the supervised setup for downstream task 1, the win-ratios are 17/20, 19/20, and 20/20 for LSTM, inception, and transformer, respectively. This indicates that bi-modal self-supervised learning during the pre-training stage gains knowledge about crops. On comparing to the uni-modal self-supervised ResMLP model, the bi-modal self-supervised win-ratios are 19/20 for LSTM and 20/20 for the other base models.

The mean classification accuracies of supervised models are $66.7\% \pm 2.53\%$, $25.84\% \pm 4.65\%$, and $71.39\% \pm 4.54\%$ for LSTM, inception, and transformer, respectively. The corresponding box plot shows the range of gain over the supervised setup. For LSTM, the mean gain over the supervised experiment is -2.26% (min: -16.33%, max: 3.87%) for uni-modal self-supervised and 3.75% (-9.38%, 9.14%) for bi-modal self-supervised. In the case of inception, the mean gain is -10.43% (-14.8%, -2.87%) for uni-modal self-supervised, while for bi-modal self-supervised, the mean gain is 8.92% (-2.26%, 19.02%). For transformers, the mean gain is 3.36% (-18.81%, 11.76%) for uni-modal self-supervised, whereas for bi-modal, the mean gain is 8.78% (0.66%, 17.68%).

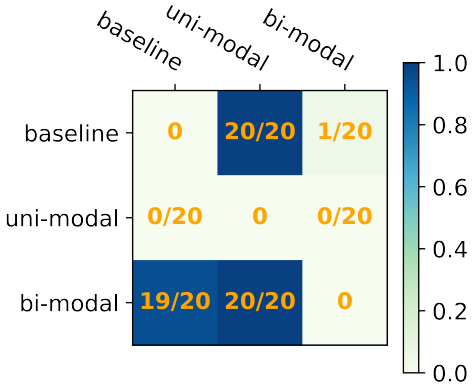
Figure 5.25 presents the results for downstream task 2. The objective of downstream task 2 is to assess how the models behave when they are applied to data from a different year and at a different geographical region with relatively similar characteristics compared to the region used for training. We find that the uni-modal self-supervised model's performance is inferior for all three baseline models. Similar to downstream task



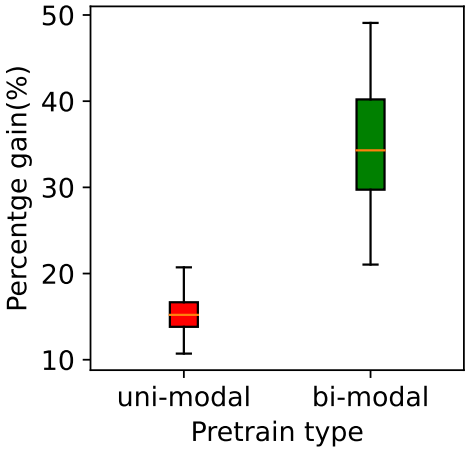
(a)



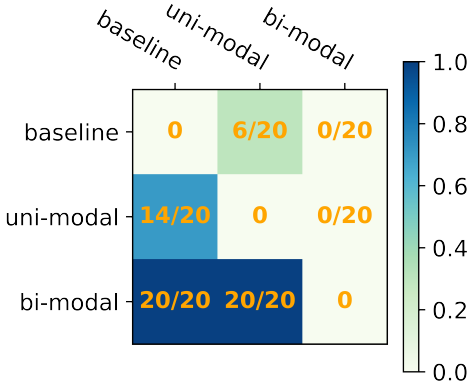
(b)



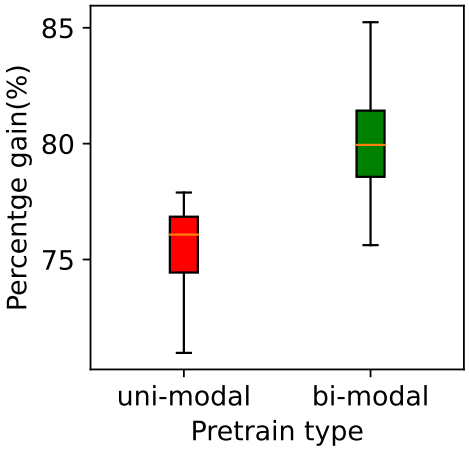
(c)



(d)

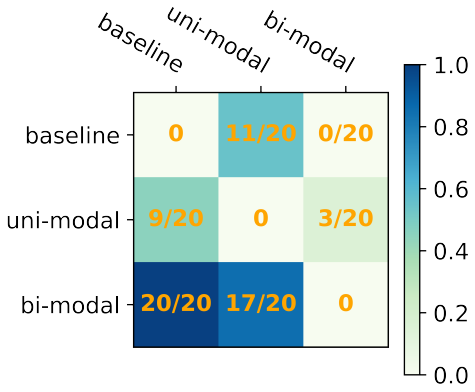


(e)

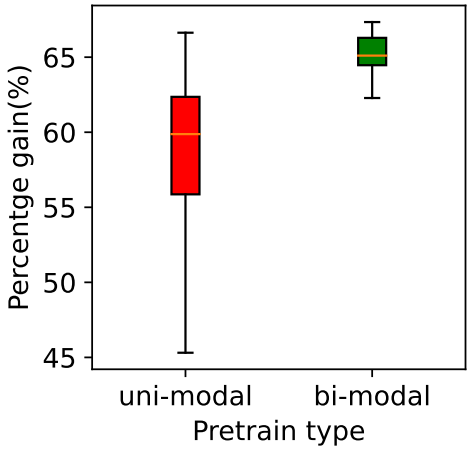


(f)

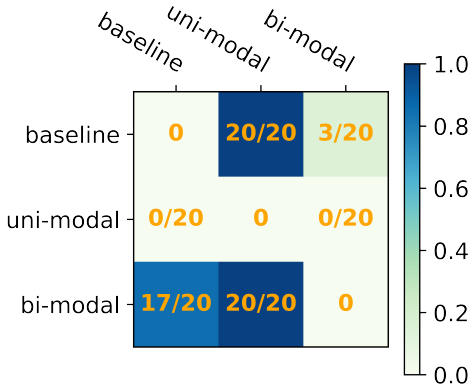
Figure 5.24: **Win-matrix and box plot for ResMLP backbone model on downstream task 1.** Plots (a), (c), and (e) show the win-matrix for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing a relative gain for both uni-modal and bi-modal self-supervised compared to the supervised experiments for LSTM, inception, and transformer, respectively.



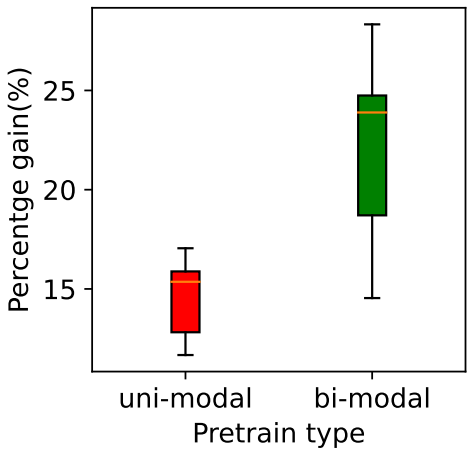
(a)



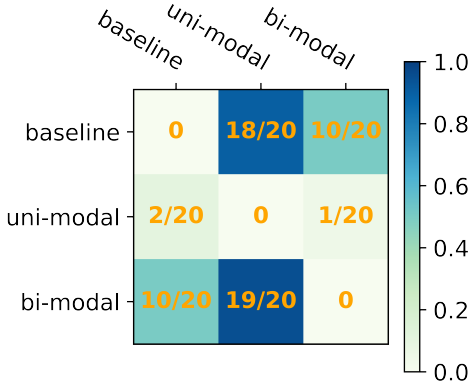
(b)



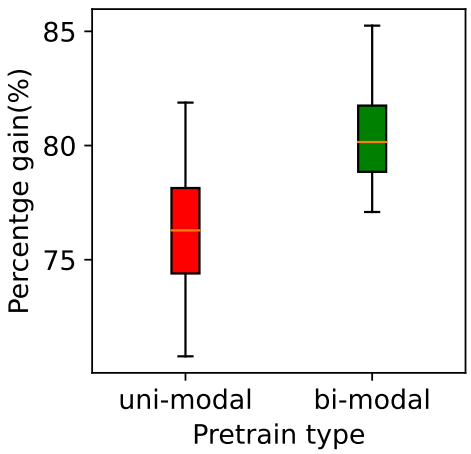
(c)



(d)



(e)



(f)

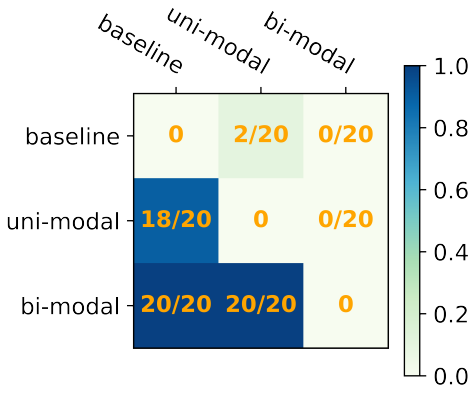
Figure 5.25: **Win-matrix and box plot for ResMLP backbone model on downstream task 2.** Plots (a), (c), and (e) show the win-matrix for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing a relative gain for both uni-modal and bi-modal self-supervised compared to the supervised experiments for LSTM, inception, and transformer, respectively.

1, the bi-modal self-supervised model outperforms the uni-model self-supervised model in all experiments. When comparing the bi-modal self-supervised ResMLP model with the supervised base model, the win-ratios are 20/20, 17/20, and 10/20 for LSTM, inception, and transformer, respectively. In comparison with the uni-modal self-supervised model, the win-ratios of bi-modal self-supervised are 17/20, 20/20, and 19/20 for LSTM, inception, and transformer, respectively. The mean classification accuracies of the supervised models are $59.31\% \pm 5.75\%$, $20.43\% \pm 3.98\%$, and $80.83\% \pm 2.69\%$ for LSTM, inception, and transformer, respectively. The box plots in Figure 5.25 show the range of gain over the supervised experiment. For LSTM, the mean gain is -0.11% (min: -10.27%, max: 12.12%) for uni-modal self-supervised, and the mean gain is 5.62% (0.46%, 19.18%) for bi-modal self-supervised. In the case of inception, the mean gain is -5.78% (-9.75%, -1.01%) for uni-modal self-supervised, whereas the mean gain is 1.77% (-1.39%, 5.62%) for bi-modal self-supervised. For transformers, the mean gain is -4.63% (-11.56%, 3.37%) for uni-modal self-supervised, and the mean gain is -0.25% (-6.76%, 5.82%) for bi-modal self-supervised.

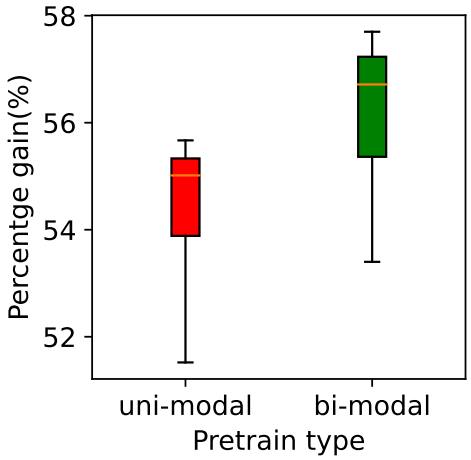
Figure 5.26 presents the results for downstream task 3. The objective of the downstream task 3 is to assess how the models behave when they are tested on data from a region that is far away from the Brandenburg region of Germany, from where our pre-training data is taken. The test data of this experiment is from Brittany, France. Consistent with the results from the previous two downstream tasks, the bi-modal self-supervised model outperforms the uni-model self-supervised model across all experiment setups. When comparing the bi-modal self-supervised ResMLP model with the base model, the win-ratios are 20/20, 9/20, and 18/20 for LSTM, inception, and transformer, respectively. When bi-modal is compared with uni-modal, the win-ratios are 20/20, 15/20, and 19/20 for LSTM, inception, and transformer, respectively. The classification accuracies for supervised models are $33.19\% \pm 5.19\%$, $16.15\% \pm 3.38\%$, and $19.92\% \pm 4.46\%$ for LSTM, inception, and transformer, respectively. In the case of LSTM, the mean gain is 1.4% (min: -0.17%, max: 2.78%) for uni-modal self-supervised, and for bi-modal self-supervised, the mean gain is 3.17% (1.82%, 4.6%). For inception, the mean gain is -3.2% (-23.59%, 11.73%) for uni-modal self-supervised, whereas for bi-modal self-supervised, the mean gain is 0.42% (-8.88%, 11.35%). For transformers, the mean gain is -2.09% (-10.04%, 1.47%) for uni-modal self-supervised and 1.56% (-1.70%, 3.55%) for bi-modal self-supervised.

Table 6.5 shows the supervised accuracy and gain for both uni-modal and bi-modal self-supervised learning for all downstream tasks.

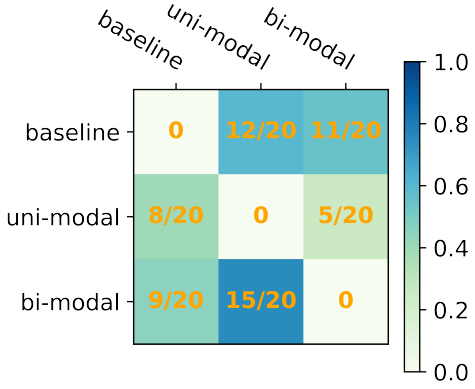
Inspired by our success with atmospheric transformation in static landcover classification as shown in Chapter 4, we conducted series of additional experiments to compare the performance of our proposed bi-modal contrastive learning to the uni-modal setup employing atmospheric transformation to generate an augmented data. In this approach, we used atmospherically corrected and uncorrected spectral measurements as different views of the same sample. Despite the potential of this method, our results showed that the bi-modal self-supervised contrastive approach still outperformed this uni-modal setup employing atmospheric transformation.



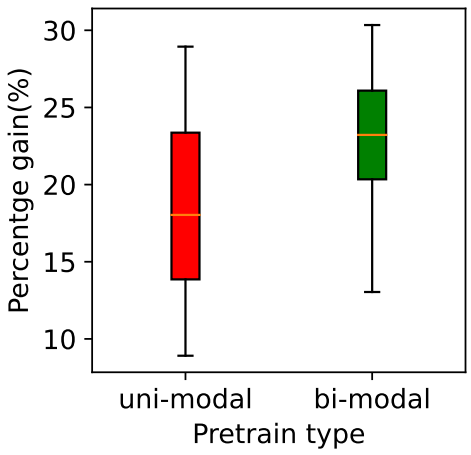
(a)



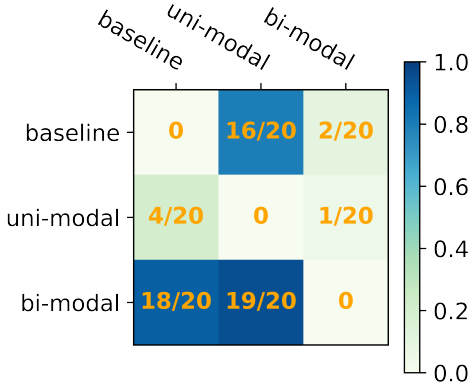
(b)



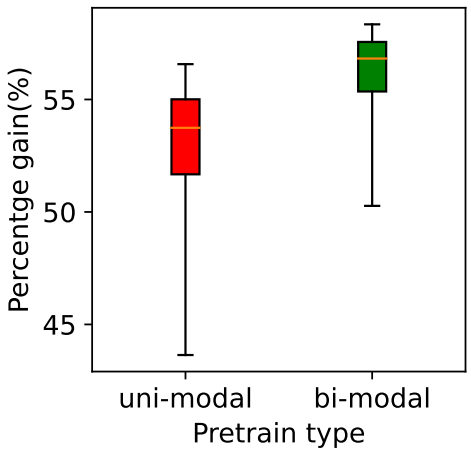
(c)



(d)



(e)



(f)

Figure 5.26: **Win-matrix and box plot for ResMLP backbone model on downstream task 3.** Plots (a), (c), and (e) show the win-matrix for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing a relative gain for both uni-modal and bi-modal self-supervised compared to the supervised experiments for LSTM, inception, and transformer, respectively.

Table 5.1: Accuracy of the supervised setup and relative gain of uni-modal and bi-modal self-supervised pre-training for the three different downstream tasks.

Downstream Task	Downstream Network	Supervised Accuracy (mean ±std)	Relative gain for uni-modal (mean)	Relative gain for bi-modal (mean)
Task1	LSTM	**66.70% ± 2.53%**	-2.26%	3.75%
	InceptionTime	**25.84% ± 4.65%**	-10.43%	8.92%
	Transformer	**71.39% ± 4.54%**	3.36%	8.78%
Task2	LSTM	**59.31% ± 5.75%**	-0.11%	5.62%
	InceptionTime	**20.43% ± 3.98%**	-5.78%	1.77%
	Transformer	**80.83% ± 2.69%**	-4.63%	-0.25%
Task3	LSTM	**53.11% ± 1.02%**	1.4%	3.17%
	InceptionTime	**21.99% ± 6.57%**	-3.20%	0.42%
	Transformer	**54.11% ± 2.09%**	-2.09%	1.56%

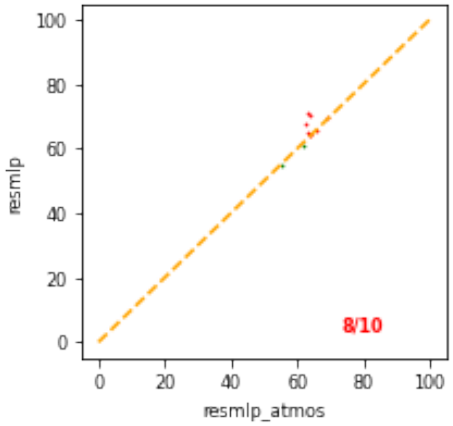
These subsequent experiments used a different set of hyperparameters for the 10 models compared to those used earlier. We employed an 8-layer ResMLP as the backbone architecture. As SCARF was not used in this uni-modal setup, we conducted 10 sets of experiments for each type of base model. Given the presence of only two competitive models, we opted for comparison plots instead of win-matrices. The relative gain box plot illustrates the accuracy improvement of the proposed bi-modal self-supervised model over the uni-modal setup employing atmospheric transformation.

Table 5.2 summarizes the comparison results for all conducted tests. Our findings indicate that the bi-modal self-supervised model outperforms the uni-modal self-supervised employing atmospheric transformation in 8 out of 9 test cases, with the exception being the LSTM model in downstream task 3. Figures 5.27, 5.28, and 5.29 display comparison plots evaluating the competitive models across downstream tasks 1, 2, and 3, respectively. In the figures, points above the diagonal break-even line, plotted in red, indicate superiority of bi-modal pre-trained model over uni-modal pre-trained model, while green points below the line indicate inferior performance. In the plot, we also reported the fractional win score, calculated as the number of times out of 10 experiments that bi-modal model surpassed the uni-modal model.

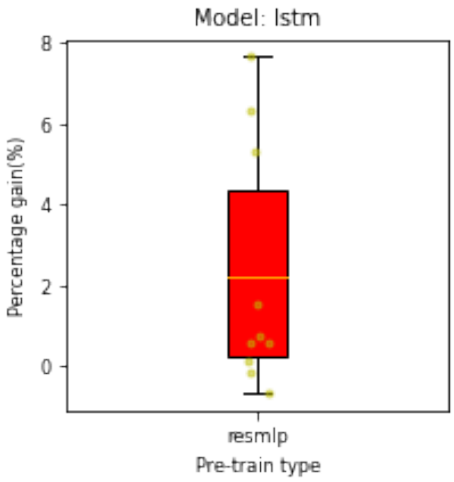
Even in the case of the LSTM model on downstream task 3, where the relative gain is slightly negative, the difference is minimal, suggesting comparable performance. This is evident in Figure 5.29, where all data points are around the break-even line.

5.7 Conclusion

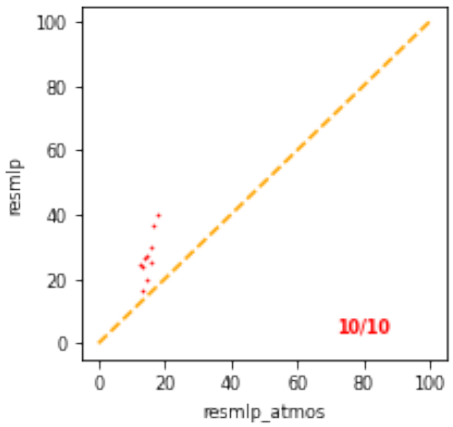
In this work, we presented a novel bi-modal, self-supervised contrastive learning method for pixel-wise crop classification from satellite images. The method uses Sentinel2 and Planetscope data together with a feature corruption technique for pre-training and employs various networks to learn the temporal patterns of the pixel spectra of different crop types. After the pre-training, the model can be applied with one data source only.



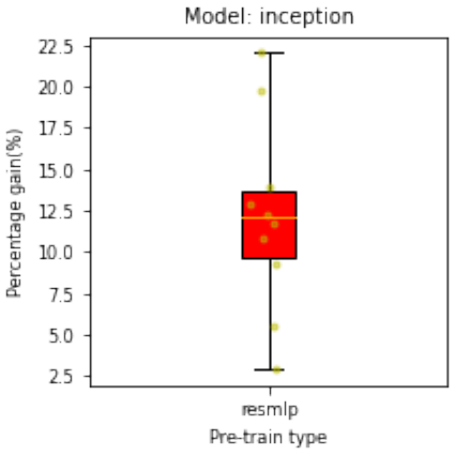
(a)



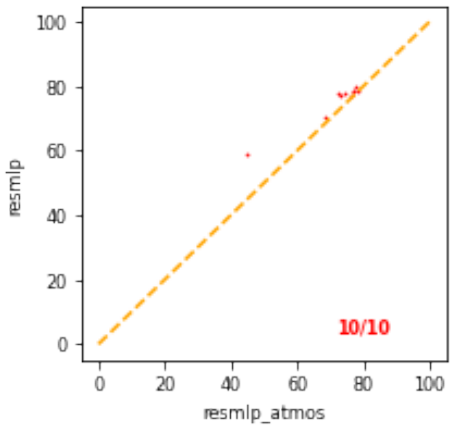
(b)



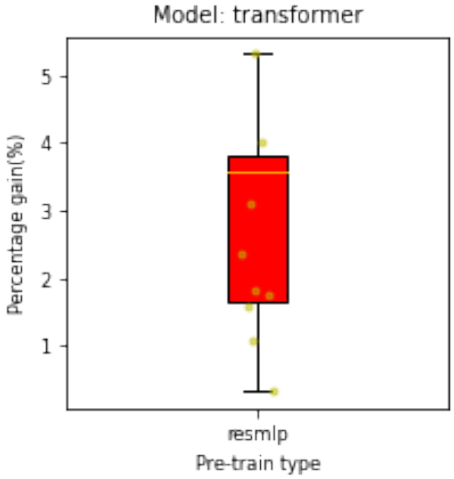
(c)



(d)



(e)



(f)

Figure 5.27: **Comparison and box plot for both pre-trained models on downstream task 1.** Plots (a), (c), and (e) show the comparison plots for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing an absolute gain of bi-modal self-supervised model over the uni-modal self-supervised model employing atmospheric transformation.

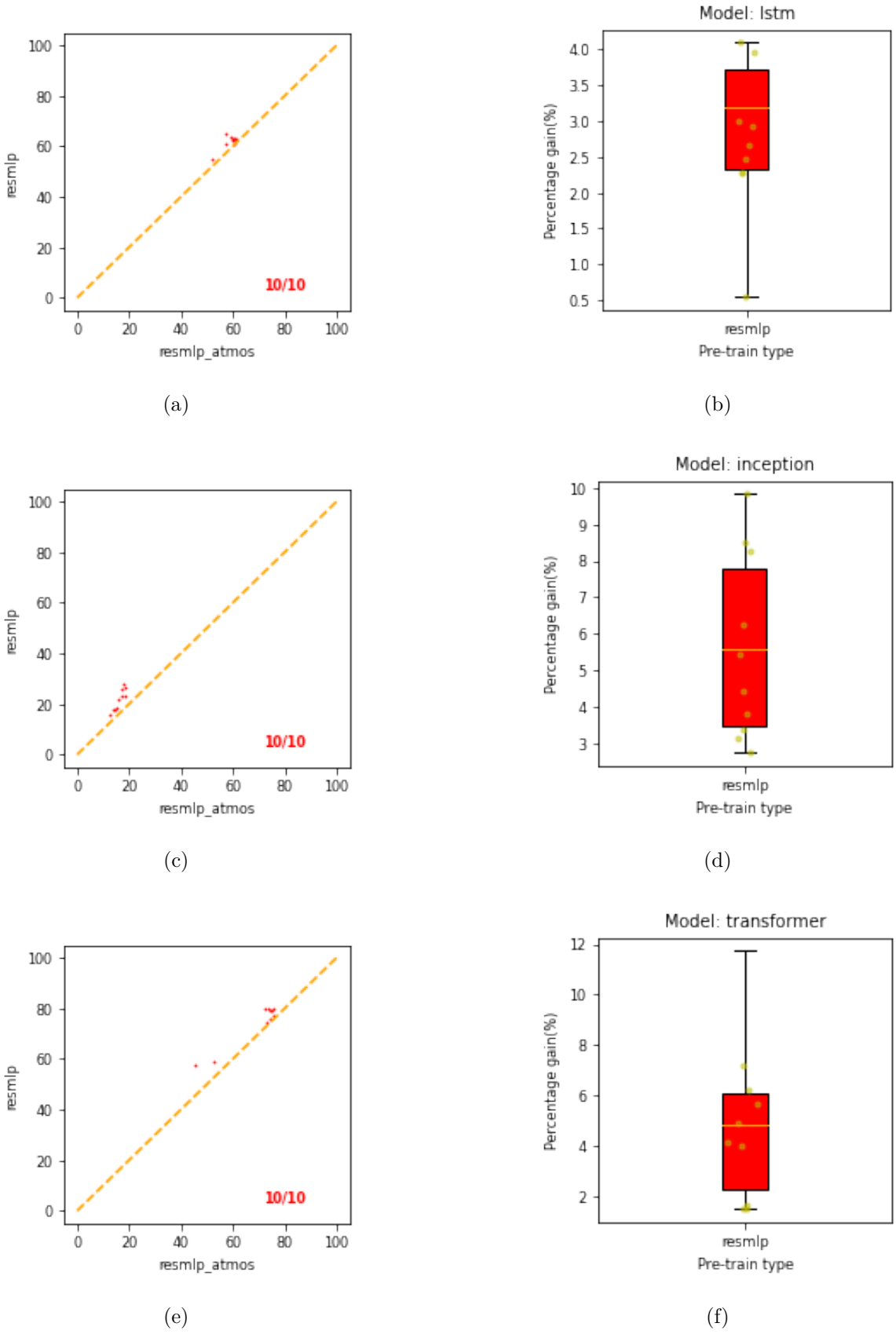
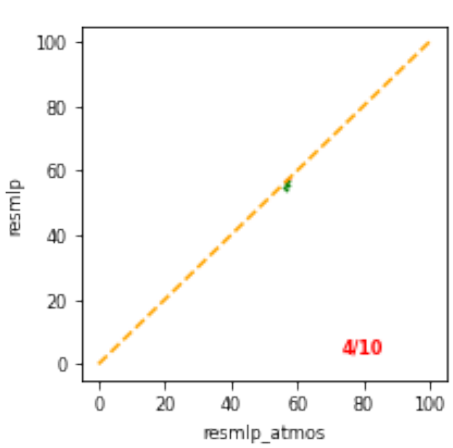
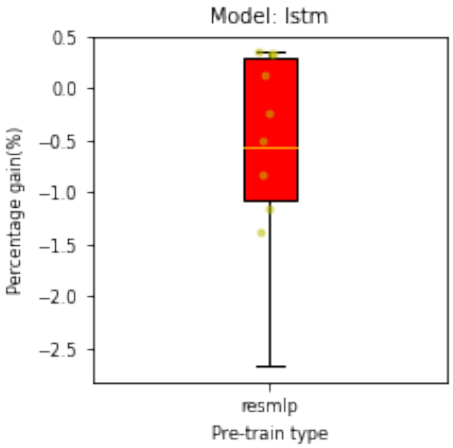


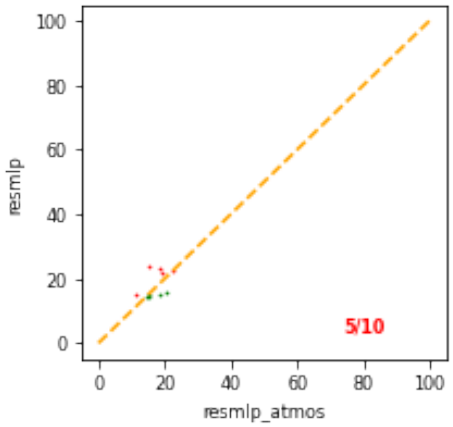
Figure 5.28: **Comparison plot and box plot for all pre-trained models on downstream task 2.** Plots (a), (c), and (e) show the comparison plots for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing an absolute gain of bi-modal self-supervised model over the uni-modal self-supervised model employing atmospheric transformation.



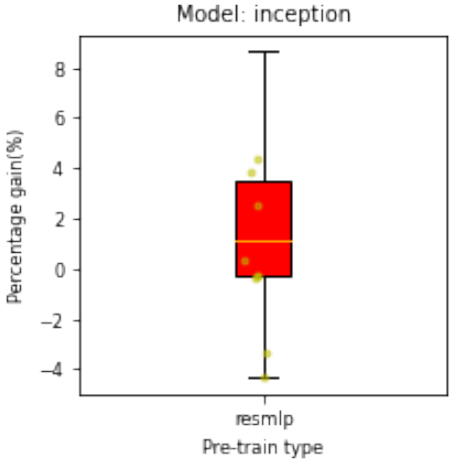
(a)



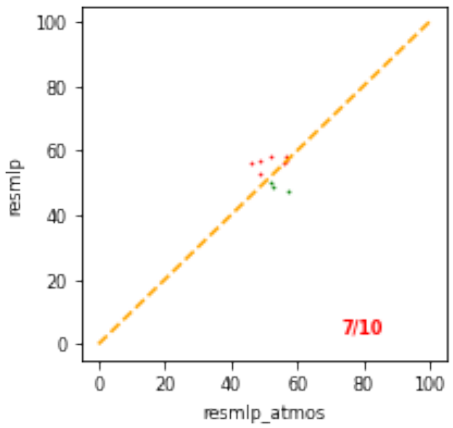
(b)



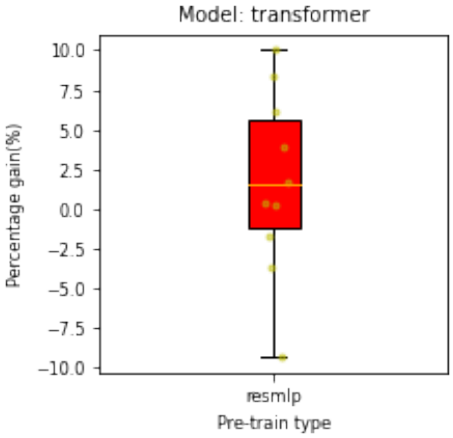
(c)



(d)



(e)



(f)

Figure 5.29: **Comparison plot and box plot for all pre-trained models on downstream task 3.** Plots (a), (c), and (e) show the win-matrix for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing an absolute gain of bi-modal self-supervised model over the uni-modal self-supervised model employing atmospheric transformation.

Table 5.2: Accuracy of the models for uni-modal self-supervised employing atmospheric transformation and bi-modal self-supervised model in three different downstream tasks.

Downstream Tasks	Downstream Network	Uni-modal self-supervised (atmospheric transformation) (mean +/- std)	Relative gain for bi-modal self-supervised (mean)
Task 1	LSTM	63.27% +/- 3.27%	2.20%
	InceptionTime	14.84% +/- 1.53%	12.13%
	Transformer	71.94% +/- 7.53%	3.57%
Task 2	LSTM	58.87% +/ 2.54%	3.18%
	InceptionTime	16.26% +/- 1.97%	5.59%
	Transformer	69.26% +/- 8.28%	4.83%
Task 3	LSTM	56.69% +/- 0.37%	-0.56%
	InceptionTime	17.03% +/- 3.20%	1.11%
	Transformer	52.37% +/- 3.78%	1.79%

We benchmarked our bi-modal contrastive learning framework against two uni-modal self-supervised approaches, both trained on Sentinel2 data. The first uni-modal method utilizes random feature corruption for data augmentation, while the second employs atmospheric transformations (Chapter 4) to generate augmented inputs. For pre-training, we adopted ResMLP as the backbone architecture and assessed the learned representations by fine-tuning them on three distinct crop classification models: a bi-directional LSTM, InceptionTime, and position-encoded transformers.

In summary, we conclude that contrastive learning using the feature corruption technique to generate positive sample pairs on Sentinel2 is unable to learn an expressive representation for crop classification. The proposed bi-modal contrastive self-supervised learning combining Sentinel2 and Planetscope, demonstrates improved performance across most test scenarios when compared to the uni-modal contrastive method using random feature corruption. For the bi-directional LSTM, we find a higher gain for all the downstream tasks. The Inception model exhibits smaller gains for downstream task 3, while the transformer network shows no improvement for downstream task 2. When comparing the bi-modal approach to the uni-modal method using atmospheric transformation (Chapter 4), we observed performance gains in all experiments except for the LSTM model in downstream task 3. Given the improvement found in most test cases, we can conclude that bi-modal contrastive learning helps in learning an expressive representation for crop classification. In the bi-modal setting, the network has learned to take into account finer-scale features from the higher-resolution Planetscope data during pre-training. As a result, the classification accuracy increases even when only Sentinel2 data are fed into the pre-trained network. All our test cases are pixel-wise crop classification. The new method can, however, be transferred to other downstream tasks like a prediction of crop yield or identification of the nutritional value of the crop at a location.

5.8 Discussion

We have shown the benefits of our bi-modal contrastive learning method over baseline methods (random feature corruption and atmospheric transformation discussed in Chap-

ter 4) but there are many points to be discussed. We have already highlighted some of the existing methods and SimCLR-like loss functions for bi-modal pre-training experiment setup. Our approach differs from existing methods in several key aspects. Unlike SeCo (Mañas et al. 2021), which used only Sentinel2 data and classical image transformations, we align two optical remote sensing sources (Sentinel2 and Planetscope) with different properties for contrastive learning. While some studies (Scheibenreif et al. 2022; C. Liu et al. 2022) have implemented multi-modal contrastive learning approaches in remote sensing by aligning optical (Sentinel2) with radar (Sentinel1) images, our method focuses on pixel-level analysis rather than whole images. This pixel-wise approach is particularly beneficial for crop classification and eliminates the need for field boundary information, making it fully self-supervised. While SCARF (Bahri et al. 2021), SAINT (Somepalli et al. 2021), and VIME (Yoon et al. 2020) utilize various augmentation techniques for tabular data, our approach employs SimCLR (T. Chen et al. 2020) contrastive loss on distinct data sources. Our methods leverage the availability of multiple sources in the field of remote sensing. Although several competitive loss functions exist, we found that SimCLR outperforms others like Barlow twins (Zbontar et al. 2021) in our bi-modal setup, aligning with findings from SCARF authors on OPENML-CC18 (Bischl et al. 2021) data. The other loss functions, MoCo (He, Fan, et al. 2019), BYOL (Grill et al. 2020), and DiNo (Caron, Touvron, et al. 2021) are not feasible for bi-modal contrastive experimental setups as the training with loss functions employs two networks with similar architecture but the weights are not shared between them. So, our proposed method offers a novel approach to crop classification using contrastive self-supervised learning, thereby advancing the fields of remote sensing and agricultural monitoring. Using a single 32GB NVIDIA Tesla V100 GPU, we can obtain the pre-trained model in just 6 hours. The model architecture consists of an 8-layer ResMLP, which is relatively small and allows for larger batch sizes. This is particularly advantageous because SimCLR is not parallelizable. However, increasing the number of layers will lead to longer pre-training times. Once pre-trained, the model can utilize either LSTM or transformer architectures as its base. While LSTMs are inefficient during training due to backpropagation through time, they run sequentially during deployment with a computational complexity of $O(1)$. In contrast, the attention mechanism in transformers has a computational complexity of $O(N^2)$. The Sentinel2 time series data for one year contains a maximum of 144 timestamps, which is significantly less than the data typically handled in generative tasks performed by GPT models. This limited number of timestamps has minimal impact on computational time when using modern GPUs. During training for 20 epochs, both models averaged less than 20 minutes for the datasets in downstream tasks 1 and 2. When scaling to millions of pixels, parallelization becomes necessary during deployment. Since this application is not real-time, we find our approach to be practical in large-scale implementations. There are some limitations of the network architecture in our approach. ResMLP is still not a state-of-the-art network. The recently proposed Spectral MAMBA network (Yao et al. 2024) has shown promising results on hyperspectral image classification. It is worth noting the feasibility of such a model as a substitute for ResMLP. The second limitation is that our work assumes that a landcover classification model is available that can detect the

croplands in arbitrary satellite scenes. Pre-training on all types of landcover might result in a representation that is less suitable for crop classification. Furthermore, the method only considers the spectral component and does not consider the potential information coming from neighboring pixels. Extending our method to include the spatial context might improve the results further.

CHAPTER 6

BERT Bi-Modal Self-Supervised Learning for Crop Classification Using Sentinel2 and Planetscope

This chapter builds upon the work presented in Chapter 5, advancing the exploration of self-supervised learning strategies to encompass both spectral and temporal aspects of remote sensing data. To enhance the multi-modal approach introduced earlier, those approaches have been expanded to models capable of processing sequential data, thereby incorporating temporal information. This strategy enables the model to implicitly learn general temporal patterns from a large volume of unannotated data samples. The newly proposed approach, when compared to the method outlined in Chapter 5, demonstrated improved performance in the majority of test cases. In the remaining instances, the results were comparable.

Individual Contribution

The following chapter is based on our paper which is currently submitted.

BERT Bi-modal Self-Supervised Learning for Crop Classification Using Sentinel2 and Planetscope
Ankit Patnala, Martin G. Schultz, and Juergen Gall
Frontiers in Remote Sensing

Ankit Patnala authored the initial draft of this manuscript. Martin G. Schultz and Juergen Gall contributed valuable technical enhancements to the experiments conducted under the proposed plan. Their scientific feedback and guidance assisted Ankit Patnala in addressing complex research challenges. The original concept originated from Ankit Patnala with the motivation to use advanced methods. Martin G. Schultz and Juergen Gall reviewed the manuscript, offering suggestions to refine certain sections of the text. Ankit Patnala took primary responsibility for implementing the project and conducting the evaluation. The successful completion of this work was made possible by the combined efforts and expertise of all team members involved.

Contents

6.1 Introduction

Crop classification is the process of identifying crops at a particular location based on the temporal pattern of their spectral signature obtained from satellite missions. The evolution of the spectral signature, which varies from crop to crop, is influenced by the crop's phenological traits such as its life cycle stages (seeding, budding, growing, and sprouting) (Meier et al. 2009). Thus, temporal information plays a crucial role in crop classification by capturing these phenological patterns over the growing season. Exploiting the temporal information will help in various applications such as optimizing farming practices and increasing crop yields.

Accurate crop classification from satellite imagery is crucial for agricultural monitoring (Luo et al. 2024), yield estimation (Dell'Acqua et al. 2018), and ensuring food security (Ray et al. 2022). Satellite missions such as Sentinel2 (ESA 2021) have provided large amounts of data, but annotating them is costly and laborious. Conventional approaches like random forest algorithms have shown limitations in their ability to generalize effectively. These models struggle to accurately predict outcomes for crop fields at different locations and even the same crop fields at different time points, as evidenced by (Račič et al. 2020; Hütt, Waldhoff, and Bareth 2020b).

Recent self-supervised approaches have shown promise in leveraging unlabeled remote sensing data by learning representations that capture meaningful patterns (Scheibenreif et al. 2022). Self-supervised learning involves training a model on a pretext task where the supervision signal is derived from the input data itself, rather than human-annotated labels (Gui et al. 2024).

Training such pre-text task is termed as pre-training. Once the model is pre-trained, it can easily be transferred to tasks where few annotated samples are available. It is found

that such models have better performance than equivalent-sized models that are trained from scratch. Self-supervised learning has shown promising results through contrastive learning approaches in computer vision (T. Chen et al. 2020) and masked language modeling techniques like BERT (Devlin et al. 2018) and GPT (Brown et al. 2020) in natural language processing. Contrastive learning aims to learn representations that bring similar data samples closer while pushing dissimilar ones apart.

Data augmentation generates meaningful similar pairs, allowing the model to learn the shared signals between them. However, designing these transformations is critical (Purushwalkam and Gupta 2020). This task becomes even challenging when working with tabular data, such as satellite reflectance values. One approach to generate such a similar pair for the tabular data is by employing SCARF (Bahri et al. 2021). SCARF employs random feature corruption, where parts of the input data are randomly corrupted to create a noisy "view" that serves as the positive pair for contrastive learning. Another approach to generate the required positive samples for remote sensing images is by using two data sources, i.e. bi-modal contrastive learning (refer to Chapter 5). Specifically in the context of crop classification, this approach leverages the complementary benefits from the higher spectral information of Sentinel2 data (ESA 2021) and the higher spatial resolution of Planetscope data [1]. This bi-modal approach outperformed uni-modal self-supervised baselines for downstream crop classification. The two sources are only used during pre-training. This means that inference of crop types can later be done with open access Sentinel2 data alone, while the fine spatial resolution information implicitly learned from Planetscope is still implicitly available from the model.

In this work, we propose to supplement the idea of exploiting varying spectral and spatial resolutions (refer to Chapter 5) by also leveraging the varying temporal resolutions of Sentinel2 and Planetscope. We identified the challenges associated with extending spectral bi-modal contrastive learning to a spectro-temporal bi-modal contrastive framework (as detailed in Subsection 6.5.4). To address these challenges, we proposed utilizing BERT (Devlin et al. 2018), a bidirectional transformer model, as an alternative approach to contrastive self-supervised learning while employing on a spectro-temporal domain. BERT is widely adopted in natural language processing but has also been applied, for example, for pre-training of a generalized weather model (Lessig et al. 2023). Its bi-directional nature allows capturing context from both preceding and succeeding time steps, providing a more comprehensive sequential representation. A key advantage of our bi-modal BERT approach over contrastive learning setups (T. Chen et al. 2020) is that it requires only a single transformer model. Contrastive methods typically involve separate encoders for different modalities. Since contrastive approaches require very large batch sizes for training, requiring two transformer based encoders is a strong limitation.

Recent works have shown that adding auxiliary tasks in parallel to the main pretext objective can further boost the performance of the pre-trained model on downstream applications (Ayush et al. 2020). Here in the context of our tasks, we propose two novel auxiliary losses alongside our bi-modal BERT approach: seasonal classifier loss and cloud prediction loss. The seasonal classification loss enables the model to learn phenological

[1] https://api.planet.com

nuances across different crop growing seasons, and the cloud prediction loss makes the model aware of atmospheric distortions of the measured satellite reflectance.

The structure of this chapter is organized as follows: Section 6.2 outlines the dataset employed in our experimental framework and emphasizes the subtle differences in data utilization between our competitive experimental setup and our standard approach. Section 6.3 provides a comprehensive explanation of the BERT methodology and its application in this study, along with a detailed description of the auxiliary losses incorporated in our research. Section 6.4 delves into the implementation specifics of our experimental setup, while the ablation study discussed in Section 6.5 examines the sensitivity and impact of various parameters. Section 6.6 presents a comparative analysis of our experimental setup against the baseline configuration. Finally, Section 6.7 summarizes our findings, and Section 6.8 explores the implications and potential avenues for future research.

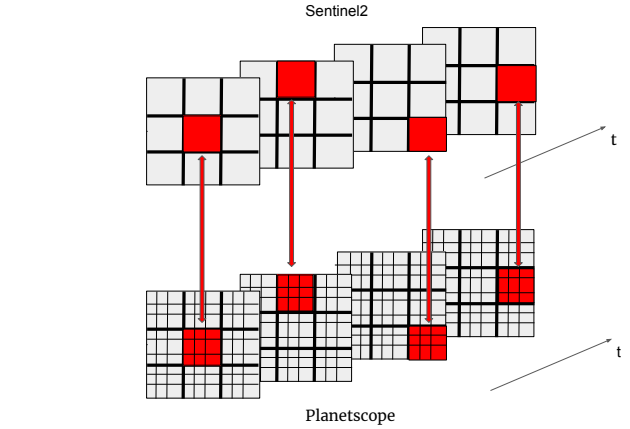
6.2 Datasets

Before we describe the self-supervised learning method in Section 6.3, we briefly describe the datasets that are used for self-supervised learning and the evaluation on three different downstream tasks. We use the data from the DENETHOR (Kondmann et al. 2021) dataset that is used in Chapter 5. The dataset includes data from Sentinel2 and Planetscope. As illustrated in Figure 6.1, Planetscope has a finer spatial resolution than Sentinel2. While the previous work employs ResMLP for their contrastive learning approach that randomly sampled pixels at each time step of Sentinel2 and enforced similarity to the corresponding pixels in the Planetscope data, we propose a spectro-temporal self-supervised method that leverages the temporal dimension by fixing a set of pixel locations and collecting the associated time series of Sentinel2 and Planetscope reflectance values, as illustrated in Figure 6.1. For self-supervised training, we sample $150,000$ time series where each time series consists of 144 points, spanning the whole year.

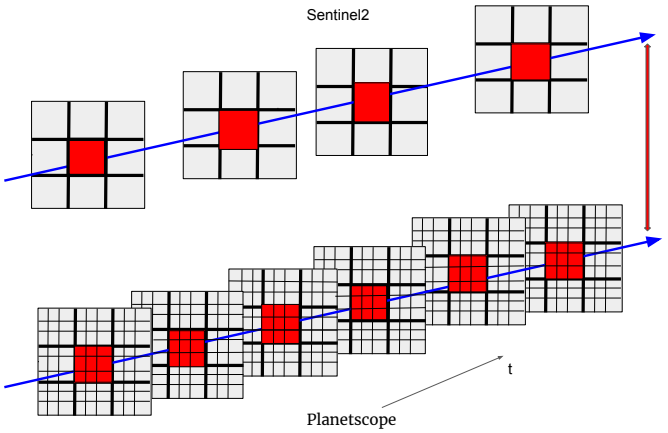
The downstream evaluation tasks for crop classification are the same as those used in Chapter 5. There are three downstream tasks in total. The downstream task 1 is from the same spatial region (Brandenburg, Germany) as the pre-training dataset from the year 2018. The downstream task 2 is from a different spatial region, albeit still in Brandenburg, and the measurements are from the year 2019. Both downstream task 1 and 2 consists of total of 45000 training data samples and 9000 validation data samples each with equal distribution across 9 classes of crops. In downstream task 3, the measurements are taken from a different region (Brittany, France). It includes 54000 training data samples and 6000 validation data with samples uniformly distributed across 6 classes.

6.3 Methods

We propose a novel bi-modal BERT-inspired pre-training strategy inspired by the original BERT model (Devlin et al. 2018) for encoding contextual representations of sequential data like text. In the original BERT, a tokenized text from a sentence is passed through



(a)



(b)

Figure 6.1: **Illustration of the sampling of the pre-training dataset in the baseline model (a) and in the spectro-temporal self-supervised method (b).** In the baseline spectral contrastive method, the spectrally resolved measurements from the same spatial location and the same timestamp are chosen as complementing pairs as indicated by the red lines connecting the pixels. In our new spectro-temporal self-supervised method, the time series of each pixel or group of pixels (shown with blue color) are used as corresponding pairs. In both panels, the top part represents Sentinel2 and the bottom part represents Planetscope.

the encoder part of a transformer. Before passing it to the encoder, some tokens are masked. Among these masked tokens, some are simply replaced by random values, some are replaced by other values from the distribution, and a few tokens are re-inserted with their original values. Similar to BERT's masked language modeling approach, we randomly mask timestamps in the Sentinel2 input sequence before passing it through the transformer encoder.

Our pre-training objective is to predict the corresponding high-resolution Planetscope reflectance values at the masked time steps and spatial locations. For this, the model has to use the contextual information from the unmasked time steps of the Sentinel2 data. To exploit the finer temporal resolution of Planetscope data, we extend this approach to predict not just the original masked time step, but also the reflectance values for two preceding time steps. This multiple timestamp prediction strategy encourages the model to capture the finer temporal resolution of Planetscope. Since we are dealing with a regression task, the loss function used is the mean squared error (MSE) between the predicted and actual Planetscope reflectance values averaged over the three time steps. Figure 6.4 illustrates this setup.

We now describe the loss functions that are used for self-supervised learning more in detail. A Sentinel2 time series with 12 spectral channels is denoted as $x_s = (x_{s1}, x_{s2}, x_{s3} \ldots, x_{st})$ where $t = 144$ and $x_{si} \in \mathbb{R}^{12}$. A Planetscope time series is denoted as $x_p = (x_{p1}, x_{p2}, x_{p3} \ldots, x_{pt})$ where $t = 365$ and each time step $x_{pi} \in \mathbb{R}^{36}$, where x_{pi} is the concatenation of 4 spectral channels over 9 pixels. Note that Planetscope has a higher spatial and temporal resolution than Sentinel2 as illustrated in Figure 6.1. While it is already demonstrated the benefit of using additional data from Planetscope for self-supervised learning although the downstream tasks are only for Sentinel2 data, we show that our proposed spectro-temporal model, which considers the temporal information over an entire year and includes two novel loss functions that consider seasonal and cloud effects, outperforms the bi-modal contrastive learning method.

Our BERT model is denoted by F and it learns a representation $z = F(x_s)$ for a Sentinel2 time series with 144 timesteps. Each timestep $z_t \in \mathbb{R}^{256}$ is represented by a 256 dimensional vector. Out of the 144 timesteps, we randomly select 90% of the timesteps denoted by $\tilde{T}$. The selected timestamps are then passed to a linear layer $g(z_{si})$ to obtain $y_{si} \in \mathbb{R}^{36}$. The corresponding timestamp for Planetscope is denoted as $x_{pi} \in \mathbb{R}^{36}$. We then compute the MSE:

$$\mathbb{L}_{bert} = \frac{1}{2|\tilde{T}|} \sum_{i \in \tilde{T}} \|y_{si} - x_{pi}\|^2, \tag{6.1}$$

i.e., we aim reconstruct the values of the spectral channels of the Planetscope data for the corresponding timestep. Since Planetscope has a higher temporal resolution than Sentinel2, we reconstruct not only 1 timestep but 3. In this case, $y_{si} \in \mathbb{R}^{3x36}$ and $x_{pi} \in \mathbb{R}^{3x36}$.

To further improve the representation that is learned in our bi-modal BERT model, we incorporate two auxiliary losses in parallel, namely seasonal loss and cloud loss. A seasonal classification loss is used to capture phenological nuances across different crop growing seasons. The transformer outputs z are aggregated by month, i.e., z_m for $m \in \{1, \ldots, 12\}$,

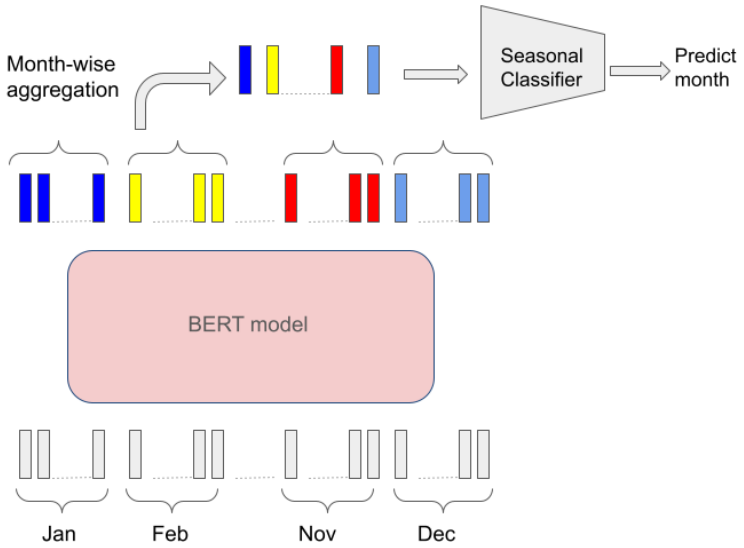


Figure 6.2: Auxiliary seasonal loss

and passed through a linear classifier h_m to predict c_m, the class probabilities of each month label $\{1,\ldots,12\}$. We then use the cross-entropy as seasonal loss:

$$\mathbb{L}_{seasonal} = -\frac{1}{12}\sum_{m=1}^{12}\log(c_{m,m}), \tag{6.2}$$

where $c_{m,m}$ is the predicted probability of the month m for z_m. This encourages the model to implicitly learn seasonal patterns that influence crop traits. Figure 6.3 illustrates the seasonal loss.

An additional cloud prediction loss is used to make the model aware of atmospheric distortions and implicitly learn the effect of clouds on the measured reflectance. Cloud measurements are readily available in the Sentinel2 dataset and discretized into 32 cloud levels. We select a subset of timesteps $\tilde{T}_{cloud}$ to directly predict the cloud levels for each timestep $i \in \tilde{T}_{cloud}$ using a linear layer with softmax $l_i = h(z_{si})$, where we denote the probability of a cloud level j at timestep i by $l_{i,j}$ and the ground-truth cloud level by cl_i. As cloud loss, we then use the cross entropy loss:

$$\mathbb{L}_{cloud} = -\frac{1}{|\tilde{T}_{cloud}|}\sum_{i\in\tilde{T}_{cloud}}\sum_{j=1}^{32}\mathbb{I}_{j=cl_i}\log(l_{i,j}), \tag{6.3}$$

where $\mathbb{I}_x$ is the indicator function, which is 1 if x is true and otherwise 0. The loss is illustrated in Figure 6.3.

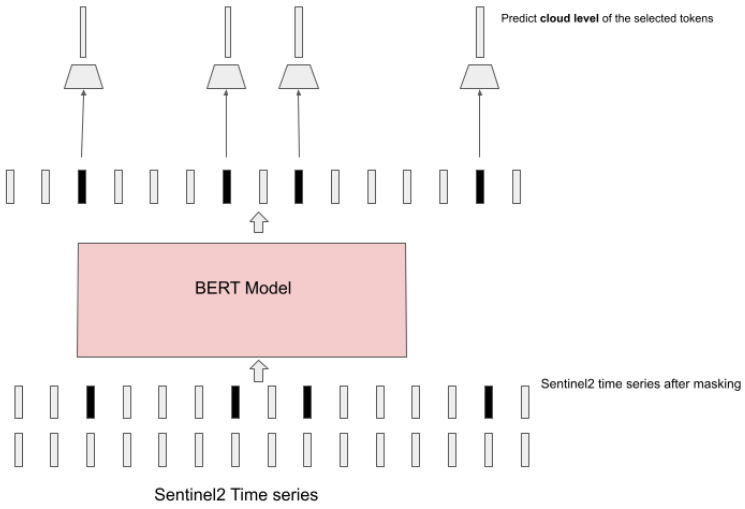


Figure 6.3: Auxiliary cloud prediction loss

6.4 Experiments

6.4.1 Implementation Details

As a backbone, we used a transformer model (Dosovitskiy et al. 2020) with 32 layers. Instead of using 2D convolutions in the initial layer, we used 1D convolutions to process the time series data of Sentinel2. The pre-training objective was to minimize the mean squared error (MSE) between the model's predicted reflectance values and the actual measurements from the Planetscope sensors. For training, we use the following masking. 10% of the tokens are masked to predict the cloud level for the cloud prediction auxiliary task. Another 70% are masked and replaced with random values, 10% are replaced with values sampled from the data distribution, and the remaining 10% are left unchanged. The network was trained for 100 epochs using a batch size of 64. The initial learning rate was set to 10^{-3}, with a warmup period of 5 epochs. A cosine annealing scheduler was employed to regulate the learning rate during training. The self-supervised trained transformer model provides a contextual time series representation.

For downstream tasks, this contextual time series representation serves as input to various base models (bi-directional LSTM, inceptiontime, and transformer) as depicted in Figure 6.5. To evaluate the effectiveness of our pre-trained model across different model configurations, we randomly generated 10 network instances for each base model type (LSTM, inceptiontime, and transformer) using Optuna (Akiba et al. 2019). For bidirectional LSTM, the hyperparameter space is defined as follows: dimensions of the hidden layer as one of $\{32, 64, 128, 256\}$, number of layers between 2 and 6, and learning rate in the range from 10^{-5} to 10^{-3}. For inceptiontime, the hyperparameter space is specified as follows: number of layers as either 2, 4, or 8, dimension of hidden layer as one of $\{128, 256, 512, 1024\}$, kernel size as one of $\{40, 80, 120, 136\}$, and learning rate between 10^{-5} and 10^{-3}. The hyperparameter space for position encoded transformers is defined

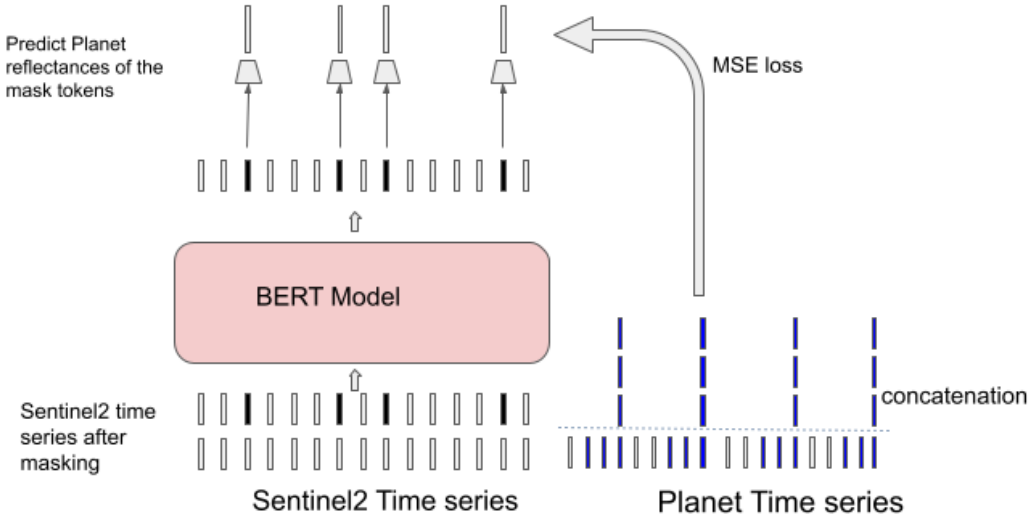


Figure 6.4: Schematic setup of our bimodal BERT model. Our model is trained on time series data from Sentinel2 and Planetscope. The time series have different spatial and temporal resolutions. In this case, we only illustrate the temporal resolution. A subset of the Sentinel2 time series is masked and the model is trained to predict the corresponding values of the Planetscope time series. Since Planetscope has a higher temporal resolution, the values of 3 timestamps instead of 1 timestamp are predicted.

as follows: the dimension of the model is either of $\{32, 64, 128\}$, the number of attention heads as one of $\{2, 4, 8\}$, the number of layers between 2 and 6, and the learning rate ranges between 10^{-5} and 10^{-3}.

6.5 Ablation Studies

This section presents an ablation analysis investigating how various parameters affect the performance of both the ResMLP and BERT pre-trained models. We explore the following aspects: Impact of number of layers in the ResMLP model (Subsection 6.5.1), effect of number of layers in the BERT model (Subsection 6.5.2), influence of masking rate on BERT (Subsection 6.5.3), and comparison between BERT and the spectro-temporal contrastive approach (Subsection 6.5.4). Analysis of auxiliary losses' contribution to the BERT model, specifically the seasonal classification and cloud prediction losses. Further, in the same subsection, we analyzed the effect of predicting multiple timestamps (Subsection 6.5.5). Finally, we examined the effects of different loss functions for auxiliary tasks (Subsection 6.5.6)

For all case studies, we report the mean classification accuracy and standard deviation across 10 models with varying hyperparameters (see Section 6.4.1). Our results focus on downstream task 2, as described in Section 6.2. We limit our ablation studies to LSTM and transformer architectures due to the inferior performance observed in inception models.

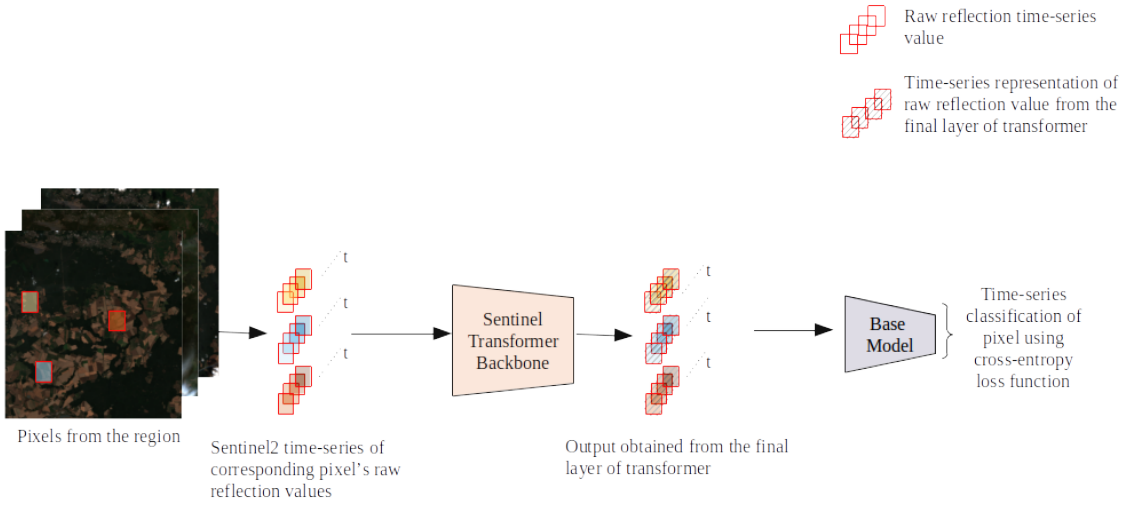


Figure 6.5: Experiment setup of the crop classification downstream tasks using the pre-trained transformer network. Each pixel is selected and the corresponding time series is then passed to the pre-trained Sentinel transformer. The output of the transformer model is an encoded representation of the time series, which is then passed to the base model for multi-class time series classification.

6.5.1 Number of layers on ResMLP model

Figure 6.6 illustrates the effect of increasing the number of layers on the ResMLP model's performance for both LSTM and Transformer architectures. The mean accuracy increases by increasing the number of layers. Notably, for LSTM models, there is a marginal increment observed when transitioning from 32 to 64 layers.

6.5.2 Effect of Number of Layers on BERT Model

Figure 6.7 illustrates the effect of increasing the number of layers for the BERT model. We observe improvement for both LSTM and transformer architectures upon increasing the number of layers. To accommodate the larger models on a single GPU, a batch size of 128 was used for the transformer with 16 layers, while a batch size of 64 was used for 32 layers.

It is important to note that for the rest of the BERT ablation experiments, a 16-layer transformer architecture was used as the benchmark model for comparing the effects of other parameters and design choices.

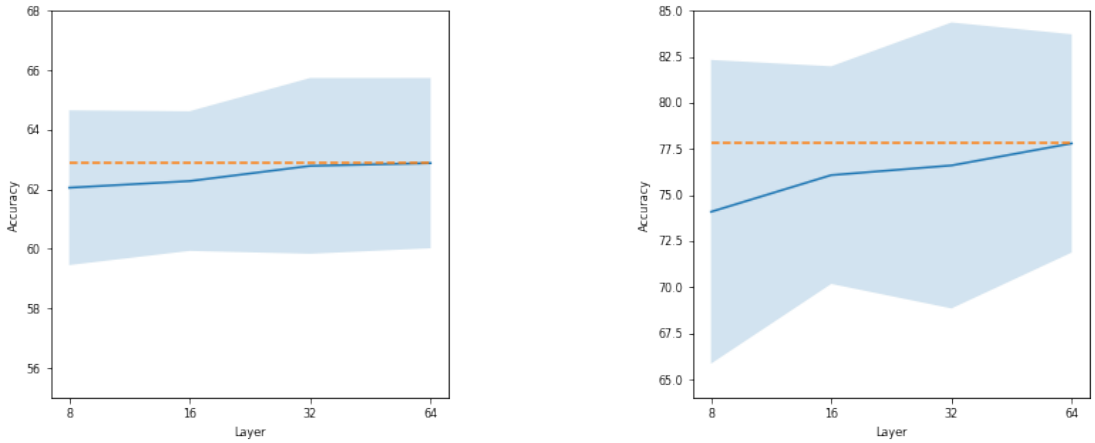


Figure 6.6: Effect of number of layers for ResMLP model. Plot a) corresponds to LSTM and plot b) corresponds to the transformer.

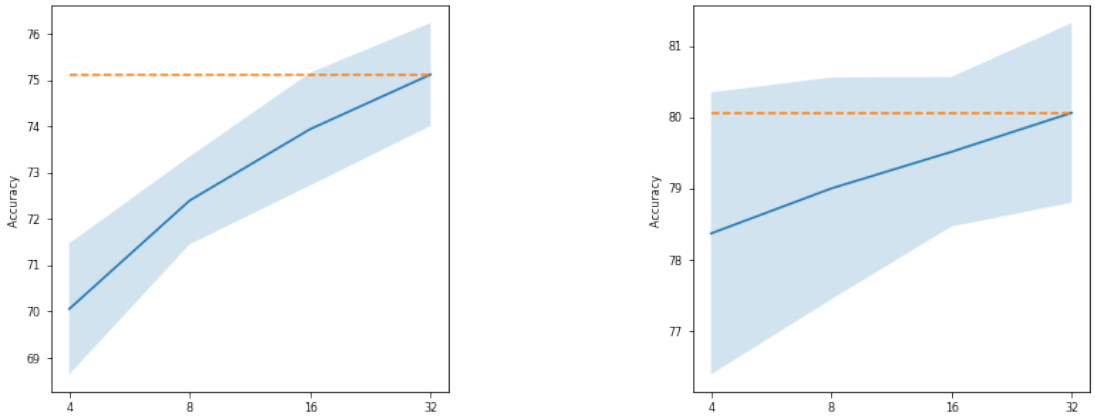


Figure 6.7: Effect of the number of layers for BERT model. Plot a) corresponds to LSTM and plot b) corresponds to transformer.

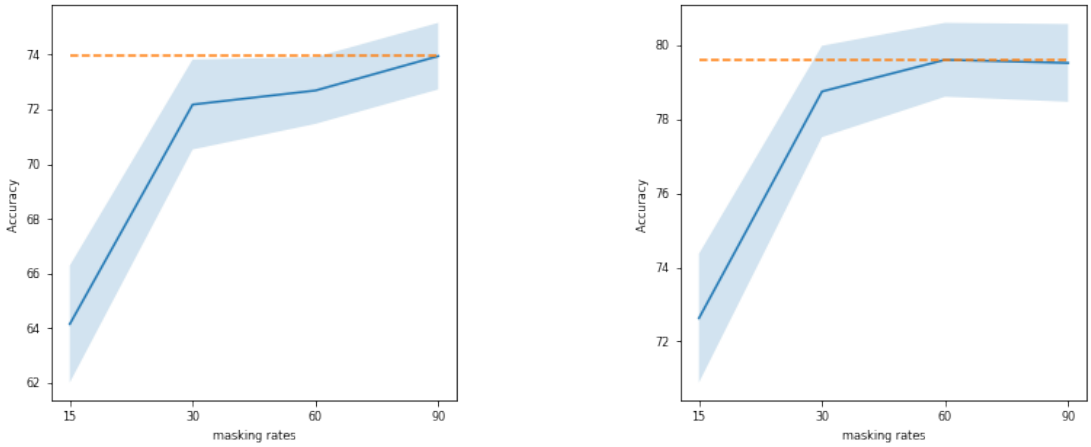


Figure 6.8: Effect of masking rate for BERT model. Plot a) corresponds to LSTM and plot b) corresponds to transformer.

6.5.3 Effect of masking rate on BERT model

Figure 6.8 illustrates a relationship between the masking rate and its effect on accuracy. For LSTM architectures, we observe an increasing trend in performance as the masking rate is increased from 15% to 90%. In the case of Transformer models, the accuracy improves from a 15% masking rate up to 60%, after which it plateaus with no significant gains observed at the 90% masking level. The mean difference in accuracy between the 15% and 90% masking rates is approximately 11.01% for LSTM models and 6.84% for transformer models.

6.5.4 Comparison between BERT and Spectro-Temporal Contrastive

Spectro-temporal contrastive should have been a natural extension of the spectral contrastive method. Instead of utilizing spectral pairs from the same timestamp as complementary views, the spectro-temporal contrastive strategy uses the time series of spectral measurements for each pixel location as the corresponding complementary pair. Figure 6.9 provides a visual description of our spectro-temporal contrastive method.

The spectro-temporal contrastive approach has certain drawbacks. This method uses two separate transformer models, which poses a constraint on the maximum batch size. Consequently, we conducted a comparative evaluation between our proposed BERT method and the spectro-temporal contrastive strategy using a smaller transformer architecture, specifically one with 4 layers and an embedding dimension of 64.

Table 6.1 demonstrates the superiority of BERT over the spectro-temporal contrastive method for both LSTM and transformer. For LSTM, there is a mean overall difference of 15.05% and for transformer, it is around 10.55%.

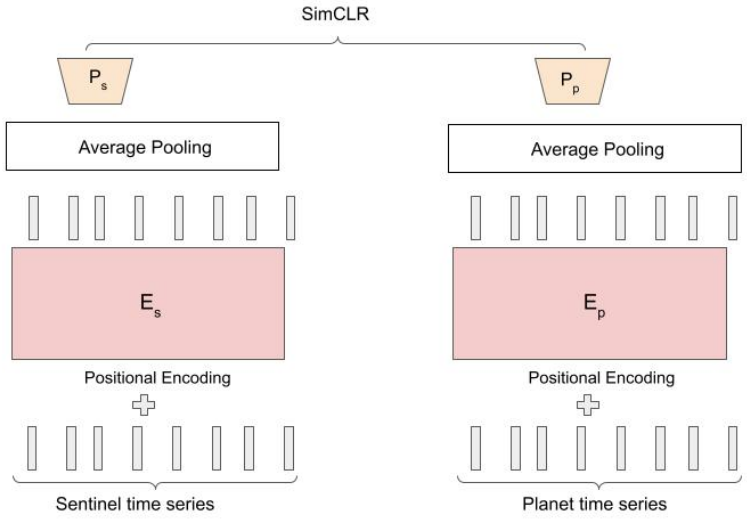


Figure 6.9: Spectro-temporal contrastive method.

Table 6.1: Comparison of BERT and spectro-temporal contrastive method when evaluated on LSTM and transformer base model

	Accuracy(mean ±std)	
Pre-trained model	LSTM	Transformer
BERT	73.94 ±1.21	79.51 ±1.05
Spectro-temporal contrastive	58.89 ±1.53	68.96 ±2.21

6.5.5 Contribution of Auxiliary Losses and Multiple Timestamps

The inclusion of the cloud prediction auxiliary loss results in an overall increase in mean accuracy of 3.88% for LSTM and 2.14% for Transformer models, highlighting the effectiveness of this auxiliary task. Similarly, the addition of the seasonal classifier as an auxiliary loss during pre-training leads to an overall increase in mean accuracy of 6.99% for LSTM and 4.65% for Transformer models, showing the benefits of including seasonal information.

Furthermore, Table 6.3 highlights the advantages of predicting three future time steps during pre-training over predicting a single time step. This results in an overall increase in mean accuracy of 6.27% for LSTM and 3.74% for Transformer models.

6.5.6 Comparison between Different Loss Function for Auxiliary Tasks

We compared our proposed cross-entropy loss for the auxiliary task against the mean square loss.

Table 6.2: Impact of seasonal loss and cloud loss

	Accuracy (mean ± std)	
Loss	LSTM	Transformer
BERT (proposed)	73.94 ± 1.21	79.51 ± 1.05
BERT without cloud prediction loss	70.06 ± 1.41	77.37 ± 1.98
BERT without seasonal classifier loss	66.95 ± 1.20	74.86 ± 3.90

Table 6.3: Impact of the number of predicted timestamps

	Accuracy (mean ± std)	
Predicted timestamps	LSTM	Transformer
3 (proposed)	73.94 ± 1.21	79.51 ± 1.05
1	67.67 ± 1.96	75.77 ± 2.98

Table 6.4 demonstrates the superiority of cross-entropy loss over the use of MSE loss for auxiliary tasks. For LSTM, there is a mean overall difference of 3.62% and for transformer, it is around 3.55%.

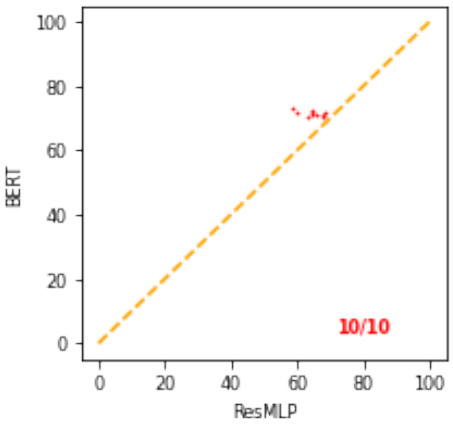
6.6 Results

We compare our proposed spectro-temporal BERT model with a contrastive learning baseline that uses datasets with different spatial resolutions, but does not rely on temporal information for self-supervised learning, as illustrated in Figure 6.1. The baseline ResMLP consists of 64 layers and was trained with a batch size of 2048. For comparison, we use comparison plots and absolute gain in performance. In the comparison plot, the x-axis presents the accuracy of the ResMLP model and the y-axis represents the accuracy of the BERT model for self-supervised learning. The dotted line across the plot represents a break-even line. Points above the diagonal break-even line, plotted in red, indicate BERT's superiority over ResMLP, while green points below the line indicate inferior performance. In the plot, we also reported the fractional win score, calculated as the number of times out of 10 experiments that BERT surpassed the ResMLP model, i.e.,

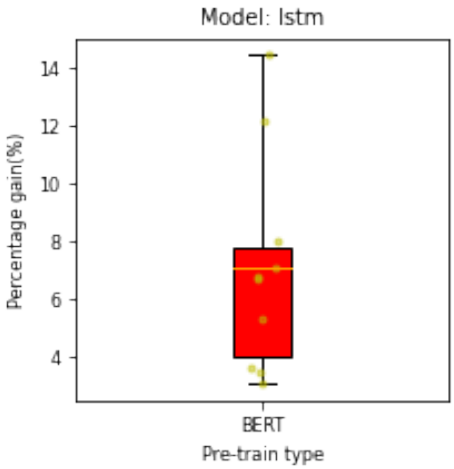
$$Win\_score = \frac{\sum_{i=1}^{N} \mathbb{I}(acc_{BERT} > acc_{ResMLP})}{N} \tag{6.4}$$

Figures 6.10, 6.11, and 6.12 present the comparison plots evaluating the competitive pre-trained models across downstream tasks 1, 2, and 3, respectively. A separate evaluation is done for each base model, i.e., LSTM, inception, and transformer. Since 10 different hyperparameter configurations are employed for each base model, the number of experiments is 10.

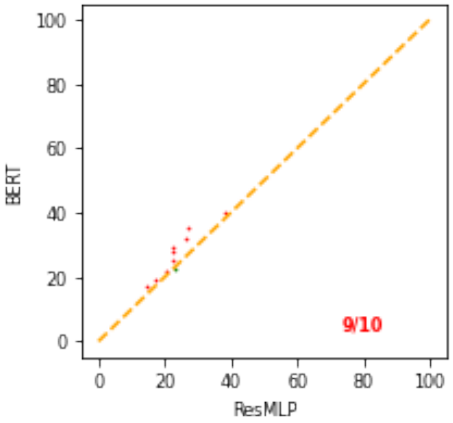
Figure 6.10 presents the comparison and box plots comparing the performance of our proposed BERT model against the ResMLP baseline for downstream task 1. The win-ratios, indicating the number of times (out of 10 experiments) BERT surpassed ResMLP,



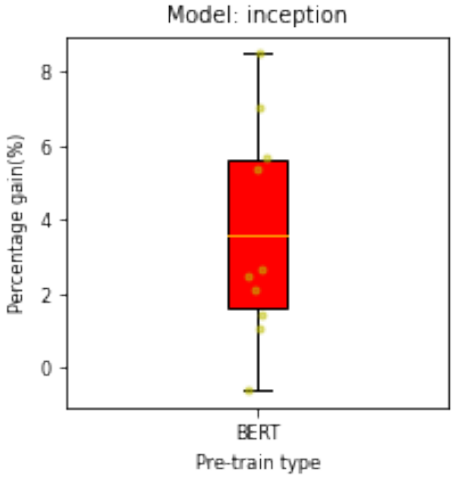
(a)



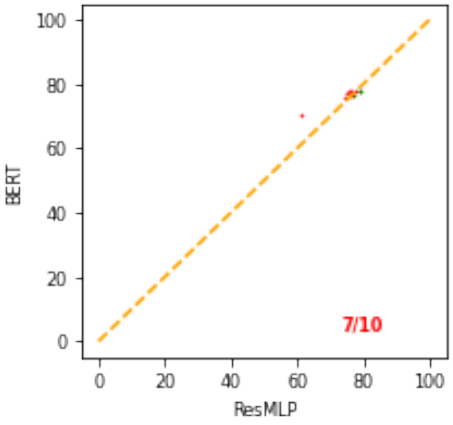
(b)



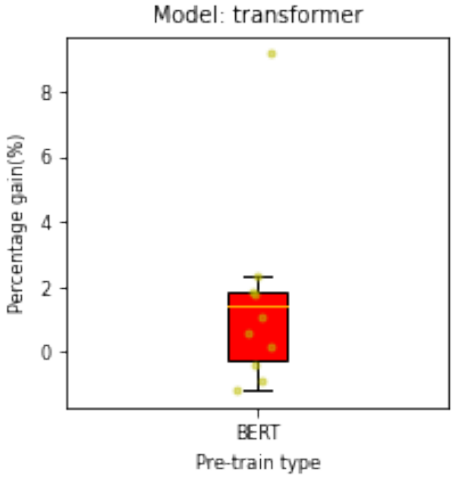
(c)



(d)



(e)



(f)

Figure 6.10: **Comparison and box plot for all pre-trained models on downstream task 1.** Plots (a), (c), and (e) show the comparison plots for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing an absolute gain for our proposed BERT model over the ResMLP pre-trained model.

Table 6.4: Comparison between different losses for the auxiliary task on BERT model when evaluated on LSTM and transformer base model

	Accuracy(mean ±std)	
Pre-trained model	LSTM	Transformer
BERT (Cross-entropy)	73.94 ±1.21	79.51 ±1.05
BERT (MSE)	70.32 ±1.99	75.96 ±1.97

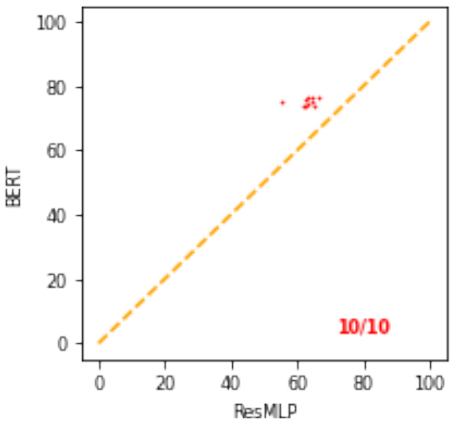
are 10/10 for LSTM, 9/10 for inception, and 7/10 for transformer models. The mean classification accuracies achieved by ResMLP models are $64.43\% \pm 3.06\%$ for LSTM, $23.41\% \pm 6.18\%$ for inception, and $75.20\% \pm 4.76\%$ for transformer models. For LSTM, the mean gain of BERT over the ResMLP is 7.04% (min: 3.02%, max: 14.46%). In the case of inception, the mean gain is 3.57% (-0.62%, 8.49%). For transformers, the mean gain is 1.44% (-1.21%, 9.18%).

Figure 6.11 presents the comparing results for downstream task 2. The win-ratios are 10/10, 10/10, and 6/10 for LSTM, inception, and transformer, respectively. The mean classification accuracies achieved by ResMLP models are $62.88\% \pm 2.86\%$ for LSTM, $22.01\% \pm 4.28\%$ for inception, and $77.80\% \pm 5.92\%$ for transformer. For LSTM, the mean gain of BERT over the ResMLP is 12.24% (8.68%, 19.75%). In the case of inception, the mean gain is 12.28% (9.05%, 15.98%). For transformers, the mean gain is 2.26% (-3.27%, 13.54%).

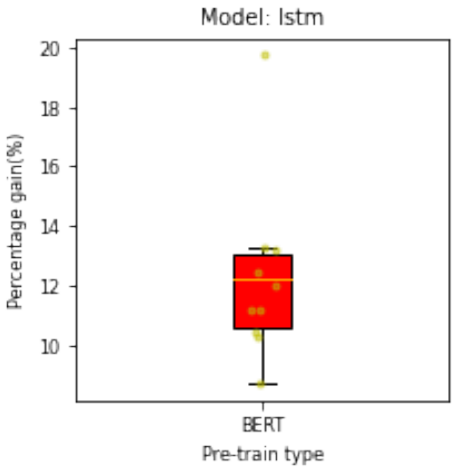
Figure 6.12 presents the results for downstream task 3. The win-ratios are 3/10 for LSTM, 9/10 for inception, and 6/10 for transformer. The mean classification accuracy achieved by the ResMLP models is $56.32\% \pm 0.47\%$ for LSTM, $16.37\% \pm 5.97\%$ for inception, and $53.87\% \pm 2.82\%$ for transformer. For LSTM, the mean gain of BERT over the ResMLP is -0.11% (-0.70%, 0.56%). It is the only setting where BERT performs on average worse than ResMLP. In the case of inception, the mean gain is 5.91% (-3.47%, 18.59%). For transformers, the mean gain is 1.15% (-2.05%, 6.93%) for BERT.

Table 6.5 presents the classification accuracy achieved by models trained using the representations learned from the ResMLP self-supervised approach and our proposed BERT bimodal method across all three downstream crop classification tasks.

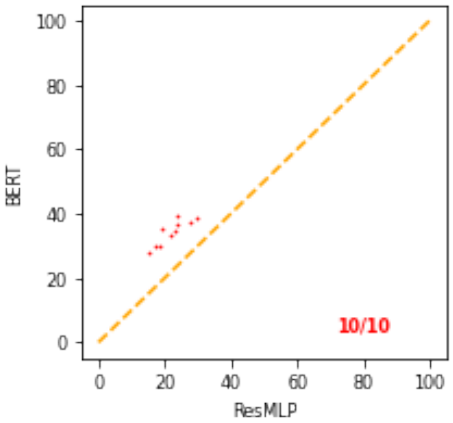
Based on the experimental results presented, the proposed bi-modal BERT approach demonstrates consistent performance gains over the baseline ResMLP method across the three downstream crop classification tasks and different base model architectures (LSTM, InceptionTime, Transformer). For LSTM architectures, the BERT approach is clearly superior on downstream tasks 1 and 2. On task 3, there is no improvement and the performance is comparable to ResMLP, with most model configurations lying close to the break-even line as shown in Figure 6.12. In the case of transformer models, the BERT method consistently outperforms ResMLP across all tasks. Figures 6.13 and 6.14 present visualization maps that compare the performance of the BERT model against the ResMLP baseline model. These maps specifically illustrate the results for one configuration on downstream task 2. The plots from different configurations also showed similar results. We are showing results from downstream task 2 because its data is relatively similar to



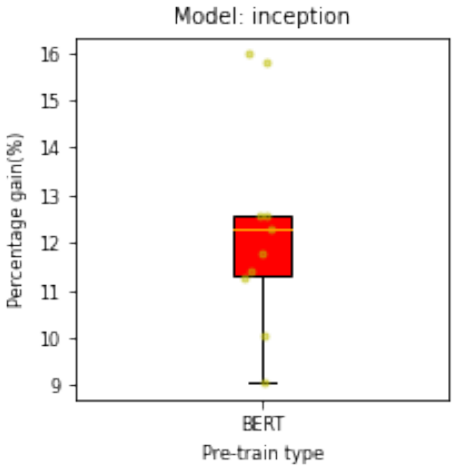
(a)



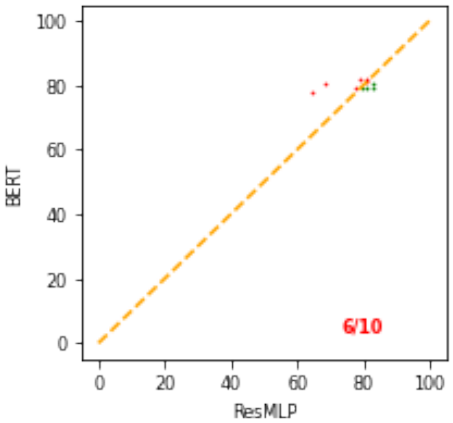
(b)



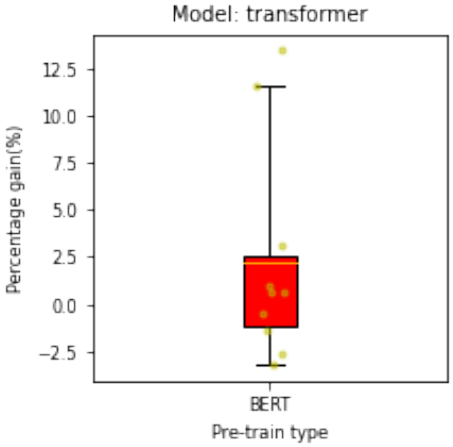
(c)



(d)

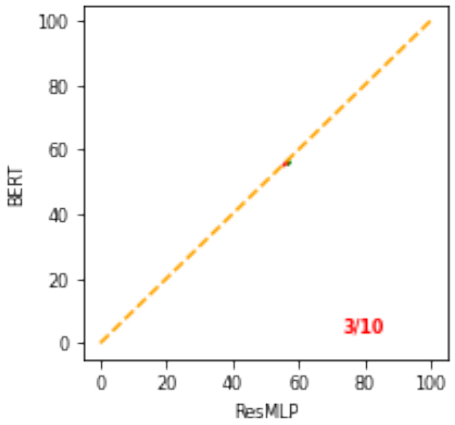


(e)

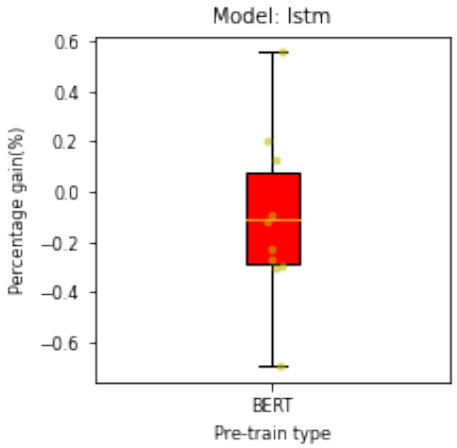


(f)

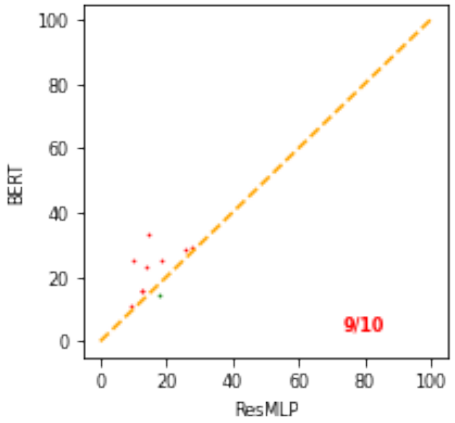
Figure 6.11: **Comparison plot and box plot for all pre-trained models on downstream task 2.** Plots (a), (c), and (e) show the comparison plots for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing absolute gain for our proposed BERT model over the ResMLP pre-trained model.



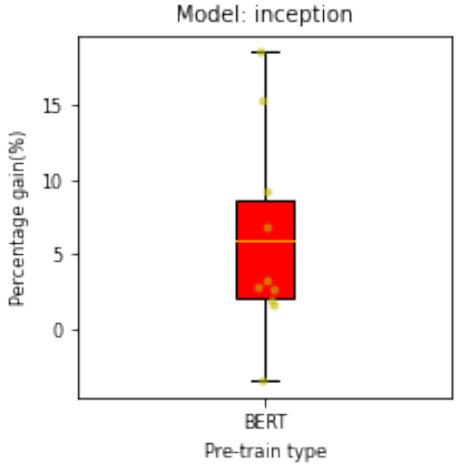
(a)



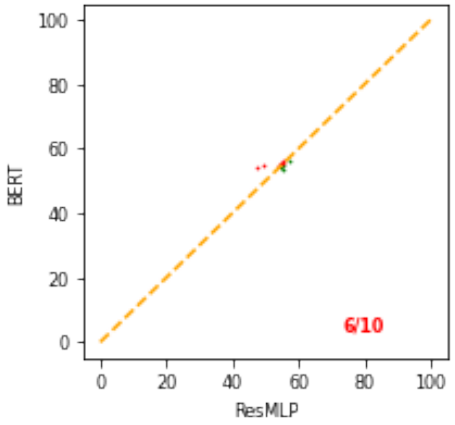
(b)



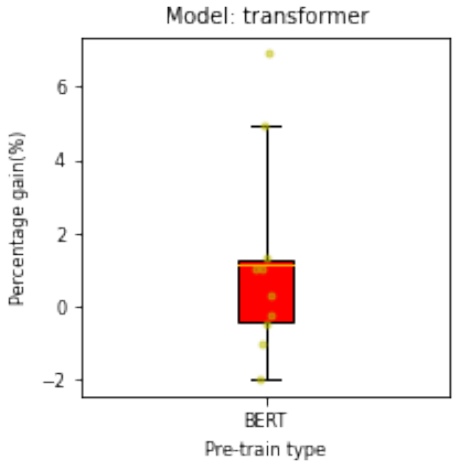
(c)



(d)



(e)



(f)

Figure 6.12: **Win-matrix and box plot for all pre-trained models on downstream task 3.** Plots (a), (c), and (e) show the win-matrix for LSTM, inception, and transformer, respectively. Panels (b), (d), and (f) correspond to box plots showing an absolute gain for our proposed BERT model over the ResMLP pre-trained model.

Table 6.5: **Accuracy of the models for both pre-trained ResMLP self-supervised model and our proposed BERT model in three different downstream tasks.**

Downstream Tasks	Downstream Network	ResMLP (mean +/- std)	Proposed BERT model (mean +/- std)
Task 1	LSTM	64.43% +/- 3.06%	71.47% +/- 0.77%
	InceptionTime	23.41% +/- 6.18%	26.97% +/- 6.95%
	Transformer	75.20% +/- 4.76%	76.64% +/- 2.09%
Task 2	LSTM	62.88% +/ 2.86%	75.12% +/- 1.11%
	InceptionTime	22.01% +/- 4.28%	34.29% +/- 3.68%
	Transformer	77.78% +/- 5.90%	80.04% +/- 1.26%
Task 3	LSTM	56.32% +/- 0.47%	56.21% +/- 0.24%
	InceptionTime	16.37% +/- 5.97%	22.28% +/- 7.13%
	Transformer	53.87% +/- 2.82%	55.02% +/- 0.81%

the data used for pre-training, unlike the data in downstream task 3. Also, the data for downstream task 1 is from the same region and time, so it doesn't provide a nice variability.

In summary, the proposed spectro-temporal BERT method outperforms the ResMLP baselines in 8 out of 9 cases and it is on par in one case. By using very different architectures for the time series classification with 10 different hyper-parameters each, we showed that the results generalize across different architectures. The results show the advantage of a temporal model compared to a contrastive learning approach. Note that both the proposed spectro-temporal BERT model and ResMLP utilize data from different sources with different spatial and temporal resolutions, namely Sentinel2 and Planetscope, but only our approach leverages the higher temporal resolution of Planetscope.

6.7 Conclusion

In this study, we extended the work mentioned in Chapter 5 by introducing an innovative bi-modal approach that combines spectral and temporal data from two satellites for self-supervised pre-training and by adopting a BERT-style training strategy. We tested this method on three distinct downstream tasks related to crop classification. Our methodology draws inspiration from the BERT model but is adapted to predict Planetscope reflectance values from Sentinel2 data. Our model utilizes spectral values for a spatial and temporal range of Sentinel2's coarser resolution temporal signature to predict temporal signature with the properties of Planetscope's higher spatial and temporal resolution. We also developed two new loss functions for self-supervised learning. The seasonal classifier loss enhances the model's ability to differentiate between seasons, while the cloud prediction task utilizes metadata to inform the model about cloud coverage in pixels at specific times, allowing it to implicitly understand cloud-related distortions in reflectance readings. As evidenced by the results in Section 6.6, our BERT bi-modal model consistently outperformed the ResMLP model across most test scenarios, and in cases with

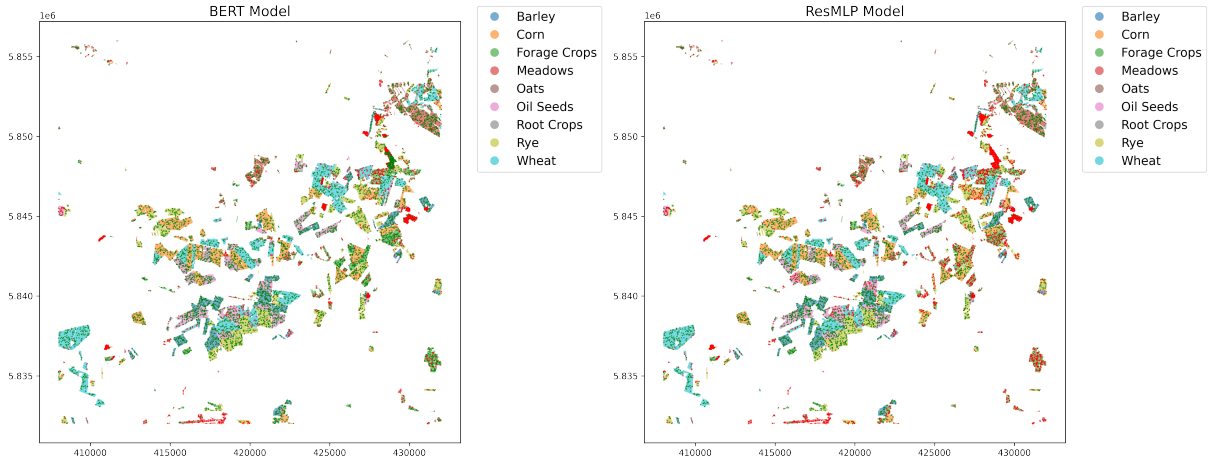


Figure 6.13: **Visualization of the crop classification map for both spectral-temporal BERT and ResMLP pre-trained model.** Green points indicate where the predictions are correct, and red points show where predictions are wrong. The results are from the LSTM model evaluated on the validation set of downstream task 2. Out of 10 available hyperparameter configurations, these results were obtained using hyperparameters corresponding to model number 5.

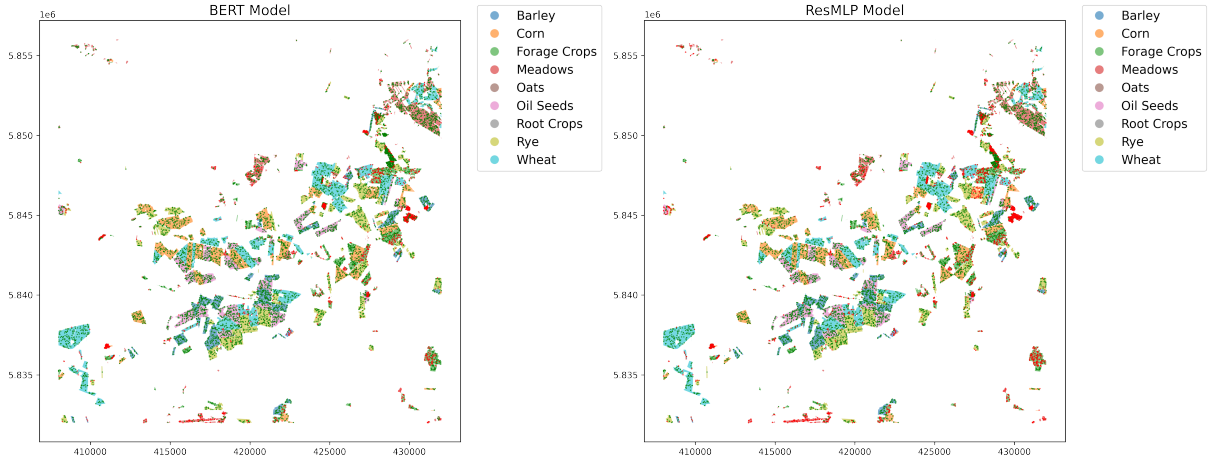


Figure 6.14: **Visualization of the crop classification map for both spectral-temporal BERT and ResMLP pre-trained model.** Green points indicate where the predictions are correct, and red points show where predictions are wrong. The results are from the transformer model evaluated on the validation set of downstream task 2. Out of 10 available hyperparameter configurations, these results were obtained using hyperparameters corresponding to model number 5.

marginal gains, it performed comparably. This suggests that our BERT approach, pre-trained on combined spectro-temporal information and enhanced with auxiliary seasonal and cloud prediction losses, learns representations that are highly effective for crop classification. We anticipate that this approach is scalable and that the model's performance could further improve with larger training datasets.

6.8 Discussion

We have demonstrated the advantages of employing spectro-temporal self-supervised methods over those that solely utilize spectral components. The BERT approach offers a key benefit over its spectro-temporal contrastive counterpart: it requires only one transformer model during training and can be easily scaled across multiple devices using data-distributed parallelism techniques (S. Li et al. 2020b). While alternative loss functions to contrastive loss, such as Barlow Twins, have been explored, studies like SCARF have demonstrated that these methods fail to yield superior representations for tabular data. Similarly, Chapter 5 highlights these findings, showing comparable results for time series crop data. Other loss functions, including MoCo (He, Fan, et al. 2019), DiNo (Caron, Touvron, et al. 2021), and BYOL (Grill et al. 2020), employ momentum-based distillation techniques, thus making them less adaptable to multi-modal setups. Our transformer model takes roughly 20 hours to train for 100 epochs on V100 32GB GPUs, mainly due to our selection of a higher number of layers and a large dimension for the intermediate MLP layer. However, the ResMLP model, with an equivalent number of layers and hidden dimensions, exhibits similar training times. It is worth noting that the transformer has a computational complexity of $O(N^2)$. In our experimental setup, we use 144 time stamps as tokens, which mitigates the impact of this complexity. Emerging state space models like MAMBA (Gu and Dao 2024a) show promise in competing with transformers, offering both convolutional and recurrent setups for faster training and inference, respectively. Such models are beginning to gain traction in the remote sensing community (Yao et al. 2024). The limitations of our approach are as follows: Our pre-training process uses a regional dataset that focuses specifically on croplands in a small area of Brandenburg. This limited geographical scope restricts the model's ability to generalize to other regions, particularly in developing areas where labeled data is scarce. The model's performance could potentially be improved by expanding the pre-training dataset to include samples from a wider range of geographic locations. This expansion would allow the model to learn more diverse crop patterns and potentially make it more suitable for use in developing countries where fewer labeled examples are available. In this study, we utilized data from DENETHOR and Breizhcrop, where the crop data covers a complete annual production cycle. However, if crops with varying annual cycles are involved, systematic batching or modern GPUs equipped with flash attention would be required. Flash attention enables efficient processing of variable-length data within a single batch, making it suitable for such scenarios. Additionally, our current approach uses a 3×3 set of pixels for Planetscope without considering larger spatial contexts. Recent works like UBARN (Dumeur, Valero, and Inglada 2024b) and ALISE (Dumeur, Valero,

and Inglada 2024a) have explored BERT-style training in the spatio-spectro-temporal domain, albeit using a single source. A promising future direction would be to investigate how our multi-modal strategy could be applied to such models.

CHAPTER 7

Conclusion and New Research Directions

Contents

7.1 Overview

This thesis explores the significance of self-supervised learning in remote sensing and investigates strategies to maximize the utilization of remote sensing data without annotation. While most self-supervised learning algorithms have been developed for common data types like natural images and text, our research focuses on adapting these techniques to the unique characteristics of remote sensing data. Remote sensing has a wide range of applications but in this thesis, our focus is on two applications: static landcover classification, and time series crop classification. The unique characteristics of remote sensing data pose significant challenges when applying contrastive self-supervised learning methods. Contrastive learning fundamentally depends on meaningful transformations. However, remote sensing images differ substantially from natural images, particularly due to the presence of multiple spectral bands beyond the visible spectrum. This distinction complicates the application of standard transformations, which are primarily designed for natural RGB images. Consequently, adapting contrastive learning techniques to remote sensing data requires careful consideration of these spectral complexities.

In Chapter 4, the first challenge is addressed by developing atmospheric transformations tailored for contrastive self-supervised learning on remote sensing images. Conventional transformations designed for natural images are incompatible with remote sensing data.

The proposed approach leverages atmospheric correction algorithms to create transformations that are applicable across all spectral bands. This method enables the use of contrastive self-supervised learning techniques on multi-spectral remote sensing data. The second challenge involves extending the benefits of self-supervised contrastive learning to time series crop classification tasks.

In Chapter 5, a bi-modal contrastive learning approach is proposed that uses spectral information from two different satellite missions. This method addresses the difficulty of creating meaningful transformations for time series data of spectral reflections.

In Chapter 6, the bi-modal self-supervised technique is further explored such that it leverages both the spectral and temporal aspects of remote sensing data. BERT, a widely used method in natural language processing is adapted to address the unique challenges of time series crop classification. This adaptation showcases the potential of BERT in handling bi-modal self-supervised learning tasks within the remote sensing domain. The proposed approach demonstrates how models originally designed for text analysis can be effectively repurposed to extract meaningful patterns from the complex, multiple spectral bands of satellite imagery time series for remote sensing applications such as crop classification.

7.2 Summary of Contribution

The primary objective of this thesis was to develop and apply self-supervised learning techniques specifically tailored for remote sensing data and its associated applications. At the start of this research, self-supervised learning methods for remote sensing data were in their early stages of development. During the course of this work, several research groups rapidly explored and advanced these techniques. Nevertheless, this work maintains its novelty through innovative approaches and provides a rigorous evaluation of results. The following sections present a comprehensive overview and summarize the key contributions made by this research.

7.2.1 Atmospheric Transformation as an Alternative to Color Jittering

Chapter 2 highlighted the unique characteristics of remote sensing imagery, prompting research question 1: How can significant features be extracted from a large set of unlabeled remote sensing data? To address this, contrastive learning, a cutting-edge self-supervised learning technique, is proposed for remote sensing applications. Traditionally, contrastive learning has been applied to natural images, which typically comprise three bands in the visible spectrum. Key to this approach is meaningful transformations, particularly color jittering and grayscaling, which alter images while preserving object textures. However, naively extending these techniques to remote sensing imagery, with its multiple spectral bands, alters the physical information inherent in each channel. In response, Chapter 4 introduces “atmospheric transformation” as a novel alternative to color jittering and grayscaling. This method leverages atmospheric correction algorithms across all spectral bands. By interpolating between corrected and uncorrected images, multiple views of

the same scene are generated without distorting the underlying physical properties of the land cover. We integrated this transformation into three distinct contrastive learning algorithms and evaluated their performance on various downstream tasks. This research led to two significant findings, the importance of non-visible spectrum channels in enhancing land cover classification accuracy and the efficacy of atmospheric transformation in contrastive self-supervised learning for extracting crucial features from remote sensing imagery.

7.2.2 Bi-Modal Contrastive Learning for Pixelwise Time Series Crop Classification

As discussed in Chapter 5, crop classification presents unique challenges compared to static landcover classification. Typically applied at either field parcel or pixel level, crop classification becomes complex in an unsupervised context due to the absence of field boundary information when analyzing random image patches containing multiple crop fields. Our research question 2 addresses this challenge: How can a meaningful transformation for tabular data, such as pixel-level spectral reflectance measurements, be developed to enable self-supervised learning in this domain? The bi-modal contrastive self-supervised learning approach is proposed that circumvents the need for explicit transformations by leveraging multiple data sources (modes). In this study, the spectral information from two satellite sources is used. The two sources are Sentinel2 and Planetscope. An experimental setup is designed for downstream tasks that incorporate input from the self-supervised pre-trained model into various time series classifier models, including convolutional (inceptiontime), recurrent (bi-directional lstm), and transformer (vanilla position-encoded transformers) architectures. Chapter 5 demonstrates the superiority and effectiveness of our bi-modal self-supervised learning approach compared to SCARF, a popular uni-modal contrastive learning method for tabular data. A key advantage of the proposed bi-modal approach is its flexibility in deployment: the pre-trained model can be applied using only publicly available Sentinel2 data for end-user downstream applications, eliminating the need for commercial PlanetScope data while applying the pre-trained model.

7.2.3 Bi-Modal BERT for Spatio-Temporal Data

Crops exhibit temporal signatures and the crop classification models can be significantly enhanced by incorporating temporal signatures, which provide richer and more dense information. Addressing research question 3 led to the exploration and development of a self-supervised method that integrates both spectral and temporal components of crop data. Chapter 6 demonstrated that simply extending contrastive learning from spectral to spectro-temporal domains poses technical challenges, particularly in terms of batch size limitations. Contrastive learning relies on large batch sizes for better generalization, adhering to principles of alignment and uniformity. However, the use of two transformer networks, coupled with GPU memory constraints, restricts this approach. By adapting the BERT model, originally designed for encoding sequential text data, a better alternative

for our spectro-temporal crop classification task is developed. This adaptation required careful consideration of two key aspects, the first concern was designing an experimental setup that effectively uses the BERT model for both pre-training and downstream tasks and the other important concern was how to leverage the benefits of multiple modalities within the BERT framework. The innovative setup involves training the model on Sentinel2 data to predict Planetscope reflectance measurements. This approach maintains the advantages of bi-modal contrastive learning and also provides the similar advantage of deploying the model using only a single data source for end-user applications. The downstream tasks are designed to be compatible with three different time series classifier models, positioning the proposed BERT-based approach as a viable alternative to spectral contrastive methods. Furthermore, the pre-training process is enhanced by incorporating auxiliary losses: seasonal classifier and cloud prediction loss. A seasonal classifier loss implicitly enables the model to generate distinct representations for different timestamps. A cloud prediction loss uses metadata to help the model account for cloud-induced distortions in surface reflectance measurements. The comprehensive ablation studies quantified the significant performance improvements achieved by integrating these auxiliary losses into our pre-trained model. This approach not only advances the field of crop classification but also demonstrates the potential for adapting and enhancing NLP techniques for remote sensing applications.

7.3 Outlook

Chapter 1 highlighted the significance of remote sensing across various domains, including environmental research, agriculture, and urban planning. Further, it emphasizes how a deeper understanding of land cover can enhance ecological, social, and economic aspects of our lives. The chapter also addressed the vast amount of data available from satellite missions, while noting the challenges associated with data annotation, particularly its cost and complexity.

This research proposes the development of self-supervised methods as a solution to harness the potential of this extensive unlabeled data, thereby mitigating the annotation costs. The continued advancement of self-supervised techniques will be crucial for future research in this field.

In the following section, the limitations of the approach in this work are critically examined and potential avenues for future research are explored. This discussion aims to provide a balanced view of our contributions while identifying areas for improvement and further investigation in the realm of self-supervised learning for remote sensing applications.

7.3.1 Limitations

Most of this work is built on top of contrastive learning. They have gained popularity in several fields, including remote sensing, due to their effectiveness in learning meaningful

representation without labeled data. However, several limitations will be discussed in this section with a special focus on remote sensing.

Contrastive learning constraints The limited availability of similar pairs is a big concern not only for contrastive learning but also for other instance-based discrimination loss functions. Though the atmospheric transformation has proved to be effective, there still remains a need for transformation that generates a higher degree of varying augmented views such that it can take into account the complex nature of remote sensing data i.e. the variability in temporal, spectral as well as spatial properties.

Scalability and performance issues Contrastive learning faces significant scalability obstacles, primarily due to its reliance on large batch sizes. The inherent characteristics of remote sensing data, including multiple spectral bands and high resolution, substantially increase the computational demands for data handling and processing. Furthermore, the incompatibility of standard available transformations with remote sensing data exacerbates the inefficiency. Thus, applying contrastive learning to remote sensing is not only computationally intensive and time-consuming but also suffers from performance inefficiencies.

Data variability and noise issues Satellite imagery is acquired under diverse environmental conditions, including different times of day, seasons, weather patterns, and cloud cover. These harsh and variable conditions often introduce noise and inconsistencies in the measurements, altering the data captured by sensors. Such variability poses challenges for contrastive learning algorithms in producing generalizable representations. These algorithms may be misled by persistent noise patterns as a supervisory signal while attempting to maximize mutual information between different views of the same scene. Consequently, this hinders the ability of contrastive learning methods to extract right features from remote sensing data.

Semantic understanding limitations Unlike natural images that typically feature one or a few distinct objects, remote sensing images are characterized by a complex distribution of numerous objects with similar shapes or spectral properties across the entire scene. This fundamental difference makes the extraction of semantic information particularly important for remote sensing applications. While most contrastive self-supervised learning algorithms excel at instance-level discrimination, which is suitable for classification tasks, they may struggle when applied to remote sensing imagery. This is because remote sensing often demands pixel-level semantic understanding.

In the following part, the limitations in the context of self-supervised learning for crop classification are discussed.

Crop classification challenges Crop classification is not as well investigated compared to the landcover classification. The crop classification in the scientific community is not standardized. Sometimes, a crop is represented as a pixel selected from its corresponding crop's field parcels, sometimes it is represented as an image of the field parcel. Sometimes the field parcel is represented as a mean or the median of the surface reflectance of all pixels in the field parcel. This diversity in representation methods creates significant obstacles in establishing a standardized process for crop classification.

The field of crop classification exhibits two distinct approaches: method-centric and data-centric. Method-centric works, such as UBARN (Dumeur, Valero, and Inglada 2024b), incorporate image analysis, while data-centric approaches like PRESTO (Tseng et al. 2024) focus on pixel-level data, integrating various sources including weather data, static variables (e.g., digital elevation models), and optical data. To advance the field, there is a clear need for standardized benchmarks that address both methodological and data aspects. For method-centric research, benchmarks like PASTIS (Garnot and Landrieu 2021) could serve as a foundation for developing state-of-the-art algorithms. In the data-centric domain, using a fixed model like PRESTO and supplementing it with additional variables could help analyze which factors contribute most significantly to crop classification accuracy. Distinguishing between these two approaches and developing them separately is crucial for promoting comprehensive research in the field. This delineation will allow focused advancements in both methodological advancement and data utilization strategies, potentially leading to more robust and effective crop classification systems.

Scarcity of Comprehensive Labeled Datasets A significant challenge in crop classification is the dearth of readily available labeled data. While landcover classification benefits from extensive programs like CLC (CORINE Land Cover) and IGBP (International Geosphere-Biosphere Programme), which have facilitated the creation of large-scale datasets such as Bigearthnet (Sumbul et al. 2019) and Sen12MS (Schmitt et al. 2019), crop classification lacks comparable initiatives. Though there exists datasets for crops such as PASTIS (Garnot and Landrieu 2021), Eurocrops (Schneider et al. 2023), and TimeSen2Crop (Weikmann, Paris, and Bruzzone 2021) but these are comparatively small. The absence of such large datasets has created a critical gap in this research field. Currently, crop data is primarily sourced through commercial channels or in limited scope field campaigns conducted for research purposes. These data collection efforts are often confined to developed countries, creating a geographical bias in available datasets. This limitation severely hampers the development and evaluation of self-supervised methods for crop classification in diverse ecoregions and geographical regions. The scarcity of diverse, comprehensive, and freely accessible crop datasets poses a significant obstacle to advancing research and innovation in this field. It restricts the ability to develop robust, generalizable models and impedes the progress of self-supervised learning techniques specifically tailored for crop classification tasks.

Subtle crop distinctions Certain crops exhibit similar growth patterns, making it challenging to distinguish subtle variations between them. Most publicly available remote sensing data have a temporal resolution on the order of days. In many regions, particularly in Europe, Sentinel2 measurements are frequently distorted by cloud cover. This combination of cloud-induced distortions and similar crop growth patterns creates significant difficulties for models to learn distinctive features.

Pretext task design It is difficult to craft domain specific pretext tasks for crops. Our multi-modal approach, while innovative, faced limitations in data collection due to the commercial nature of Planetscope imagery. Consequently, our pre-training dataset was restricted to a small region and a narrow time frame. This contrasts with the ideal self-supervised learning scenario, which typically leverages vast and diverse datasets. For

instance, a more comprehensive approach in our case could have used datasets spanning the entirety of Europe. Furthermore, the creation of truly domain-specific pretext tasks incorporating crucial agricultural factors of the multi-annual data such as phenology, weather patterns, and soil types remains a complex undertaking. These specialized tasks, if successfully developed, could substantially enhance the representational power of models for agriculture-related applications. The current limitations in pretext task design underscore the need for continued research and innovation in this area to fully harness the potential of self-supervised learning in crop classification and broader agricultural remote sensing applications.

7.3.2 Potential Research Direction

Alternative instance based discriminative self-supervised methods The use of contrastive learning has shown quite a lot of advances in the field. Different instance-based discrimination self-supervised methods discussed in Chapter 3 have shown equally good results when compared to contrastive losses. Some of them do not even require a higher batch size. It would be worth investigating how other instance-based discrimination losses perform.

SigLIP an alternative to CLIP Recently, SigLIP (sigmoid loss for language image pre-training) (Zhai et al. 2023) has been developed as an alternative to CLIP (contrastive language image pre-training) (Radford et al. 2021). Though it has been used in the context of language and image, it can be easily applied to both the uni-modal and multi-modal contrastive setup used in this work. In the following loss function, the softmax-based contrastive loss is replaced by sigmoid-based loss which processes the pair independently. The loss is given as per Equation (7.1)

$$L_{ij} = -\frac{1}{|\beta|}\sum_{i=1}^{|\beta|}\sum_{j=1}^{|\beta|} log\frac{1}{1+e^{z_{ij}(-tx_iy_j+b)}} \tag{7.1}$$

In Equation (7.1), x_i and y_j denote embeddings from two distinct modalities. The binary variable z_{ij} is 1 if indices i and j form a positive pair (matched samples), and -1 otherwise. Further, b and t are learnable parameters. t is called globally free learnable parameters which is similar to the temperature parameter in contrastive loss. b is a learnable bias term used to alleviate the heavy imbalance loss coming from the more negative examples. In the equation, β represents the batch. The SigLIP loss function is quite easier to implement than the contrastive loss and also has the ability to process the pairs independently enabling scaling the batch size to a limit which is much higher compared to softmax-based contrastive loss. Independent processing also makes it friendlier in the context of distributed machine learning. While SigLIP offers several advantages over CLIP, the importance of data curation remains critical for large-scale multi-modal pretraining. A significant advancement in this area is the Joint Example Selection (JEST) algorithm (Evans et al. 2024), which emphasizes the importance of selecting effective data batches rather than individual examples. JEST can be seamlessly integrated with both CLIP and SigLIP algorithms, offering a simple yet powerful approach that not only ac-

celerates training iterations and reduces computational requirements but also surpasses the performance of reference models on various downstream tasks. This innovative approach to data selection has potential applications in remote sensing. By adapting JEST or similar methodologies, researchers could optimize the selection of crucial samples for pre-training models using contrastive or sigmoid-based loss functions. This could lead to more efficient and effective training processes in remote sensing applications, potentially improving model performance while reducing computational demands.

Masked autoencoders and BERT for remote sensing appliations While our BERT implementation used sequential spectral time series data, recent developments in Vision Transformer (ViT) models have led to a surge in the use of masked autoencoders for self-supervised learning across various domains. This trend has resulted in the success of BERT-like models applied to image data. (He, X. Chen, et al. 2021) shows masked autoencoders are scalable learners. They further showed the generalization ability to several downstream tasks with similar data types. In their approach, they discard the masked token rather than replacing it with other values (use position encoding as supervisory information about the location of patches/tokens) which indeed makes it efficient and fast. Using this approach, that is by discarding the masked tokens, the proposed BERT can be made efficient, and further, the proposed BERT can be extended to images by taking spatial neighborhood information into account.

Crops specific pre-text task The development of effective pretext tasks for self-supervised learning in crop classification remains a crucial area for advancement in remote sensing and agricultural monitoring. As the field progresses, we recognize the need for more generalized and robust pretext tasks that can capture the unique characteristics of crop data. There is a possibility for two tasks that could be very promising for the field: field parcel boundary detection, and unsupervised object (crop field) detection. Field boundary detection involves the unsupervised detection of field boundaries from satellite imagery. This task is particularly relevant in agricultural remote sensing, as field boundaries are crucial for subsequent crop classification and monitoring. There exist supervised learning algorithms for such task (Waldner and Diakogiannis 2019). By developing unsupervised algorithms that can accurately delineate field parcels without supervision, a foundation can be created for more sophisticated two-stage methods. These methods could then employ CNNs or ViTs to analyze both spatial and temporal patterns within the identified fields. The fully unsupervised object detection pretext task aligns with trends in the machine learning community, focusing on unsupervised object detection that incorporates spatial neighborhood information (Ciocarlan et al. 2024). In the context of crop classification, this could involve identifying distinct crop patches or agricultural features without labeled data, while considering their spatial relationships and contextual information. PrithviEO (Szwarcman et al. 2025) is one of the latest model that showcases the versatility of multi-temporal foundation model.

Alternative network architectures Recent developments in neural network architectures have introduced alternatives to transformer networks, which were originally designed for discrete sequential data like text. However, many sequential data types, such as spectral signals, are continuous in nature. While transformers are efficient during training,

they face challenges during inference due to their need to attend to each token and their lack of hidden states. State Space Models (SSMs) (Gu, Goel, and Ré 2021) represent a novel architecture that applies concepts from control theory to model dynamic systems using state variables. These models address the efficiency issues of transformers during inference, which have a computational complexity of $\mathbb{O}(N^2)$. SSMs offer two interchangeable representational states of a continuous state: a convolution state and a recurrent state. This dual representation allows for efficient training using the convolution state and improved inference performance using the recurrent state. However, SSMs initially faced challenges in differentiating the importance of tokens. A variant, selective SSMs such as MAMBA (Gu and Dao 2024b), addresses this by allowing the model to focus on important tokens and disregard less significant ones. MAMBA incorporates multiple stacked blocks and employs various algorithms, better initialization to enhance long-range dependency capture, information compression for token selection, and hardware aware algorithms for computational efficiency. These models have shown promising results in pre-training tasks, outperforming similarly-sized transformer models in language modeling. In a parallel development, the xLSTM model (Beck et al. 2024) has been introduced to address two key challenges: modifying the original LSTM to better capture long-term dependencies and matching the scaling capability of transformers. xLSTM employs two distinct blocks, sLSTM and mLSTM, and replaces the sigmoid activation function for input and forget gates with an exponential function. To mitigate potential overflow issues caused by large exponential values, a normalizer state is implemented. While xLSTM offers advantages such as linear storage complexity, it is still in its early stages and faces some limitations. These include limited parallelizability due to memory mixing in sLSTM and quadratic computational complexity in mLSTM. Despite these challenges, xLSTM has shown promising results, outperforming both transformers and SSMs in certain test cases, indicating its potential as a valuable addition to the field of sequential data processing.

Bibliography

Achard, Frédéric and Estreguil, Christine (1995). "Forest classification of Southeast Asia using NOAA AVHRR data". In: *Remote Sensing of Environment* 54.3, pp. 198–208. ISSN: 0034-4257. DOI: https://doi.org/10.1016/0034-4257(95)00153-0. URL: https://www.sciencedirect.com/science/article/pii/0034425795001530 (cit. on p. 2).

Akiba, Takuya et al. (2019). "Optuna: A Next-Generation Hyperparameter Optimization Framework". In: KDD '19. Anchorage, AK, USA: Association for Computing Machinery, pp. 2623–2631. ISBN: 9781450362016. DOI: 10.1145/3292500.3330701. URL: https://doi.org/10.1145/3292500.3330701 (cit. on pp. 90, 112).

Audebert, Nicolas, Saux, Bertrand Le, and Lefèvre, Sébastien (2017). "Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks". In: *CoRR* abs/1711.08681. arXiv: 1711.08681. URL: http://arxiv.org/abs/1711.08681 (cit. on p. 32).

Ayush, Kumar et al. (2020). "Geography-Aware Self-Supervised Learning". In: *CoRR* abs/2011.09980. arXiv: 2011.09980. URL: https://arxiv.org/abs/2011.09980 (cit. on pp. 52, 107).

Badrinarayanan, Vijay, Kendall, Alex, and Cipolla, Roberto (2015). "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *CoRR* abs/1511.00561. arXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561 (cit. on p. 32).

Baetens, Louis, Desjardins, Camille, and Hagolle, Olivier (2019). "Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask Processors Using Reference Cloud Masks Generated with a Supervised Active Learning Procedure". In: *Remote Sensing* 11.4. ISSN: 2072-4292. DOI: 10.3390/rs11040433. URL: https://www.mdpi.com/2072-4292/11/4/433 (cit. on p. 59).

Bahri, Dara et al. (2021). "SCARF: Self-Supervised Contrastive Learning using Random Feature Corruption". In: *CoRR* abs/2106.15147. arXiv: 2106.15147. URL: https://arxiv.org/abs/2106.15147 (cit. on pp. 86, 91, 92, 103, 107).

Bardes, Adrien, Ponce, Jean, and LeCun, Yann (2021). "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning". In: *CoRR* abs/2105.04906. arXiv: 2105.04906. URL: https://arxiv.org/abs/2105.04906 (cit. on p. 46).

Barlow, H B and Rosenblith, W A (1961). "Possible principles underlying the transformations of sensory messages". In: MIT Press, pp. 217–234 (cit. on p. 46).

Basu, Saikat et al. (2015). "DeepSat - A Learning framework for Satellite Imagery". In: *CoRR* abs/1509.03602. arXiv: 1509.03602. URL: http://arxiv.org/abs/1509.03602 (cit. on pp. 3, 58).

Beck, Maximilian et al. (2024). *xLSTM: Extended Long Short-Term Memory*. arXiv: 2405.04517 [cs.LG]. URL: https://arxiv.org/abs/2405.04517 (cit. on p. 135).

Bischl, Bernd et al. (2021). "OpenML: A benchmarking layer on top of OpenML to quickly create, download, and share systematic benchmarks". In: *NeurIPS*. URL: https://openreview.net/forum?id=OCrD8ycKjG (cit. on pp. 93, 103).

Brown, Tom B. et al. (2020). "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165. arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165 (cit. on p. 107).

Bruzzone, Lorenzo and Demir, Begüm (2014). "A Review of Modern Approaches to Classification of Remote Sensing Data". In: *Land Use and Land Cover Mapping in Europe: Practices & Trends*. Ed. by Ioannis Manakos and Matthias Braun. Dordrecht: Springer Netherlands, pp. 127–143. ISBN: 978-94-007-7969-3. DOI: 10.1007/978-94-007-7969-3_9. URL: https://doi.org/10.1007/978-94-007-7969-3_9 (cit. on p. 32).

Cai, Yaping et al. (2018). "A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach". In: *Remote Sensing of Environment* 210, pp. 35–47. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2018.02.045. URL: https://www.sciencedirect.com/science/article/pii/S0034425718300610 (cit. on pp. 4, 33).

Caron, Mathilde, Bojanowski, Piotr, et al. (2018). "Deep Clustering for Unsupervised Learning of Visual Features". In: *CoRR* abs/1807.05520. arXiv: 1807.05520. URL: http://arxiv.org/abs/1807.05520 (cit. on p. 44).

Caron, Mathilde, Misra, Ishan, et al. (2020). "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *CoRR* abs/2006.09882. arXiv: 2006.09882. URL: https://arxiv.org/abs/2006.09882 (cit. on p. 44).

Caron, Mathilde, Touvron, Hugo, et al. (2021). "Emerging Properties in Self-Supervised Vision Transformers". In: *CoRR* abs/2104.14294. arXiv: 2104.14294. URL: https://arxiv.org/abs/2104.14294 (cit. on pp. 45, 103, 125).

Chen, Ting et al. (2020). "A Simple Framework for Contrastive Learning of Visual Representations". In: *CoRR* abs/2002.05709. arXiv: 2002.05709. URL: https://arxiv.org/abs/2002.05709 (cit. on pp. 40, 41, 60, 70, 84, 103, 107).

Chen, Xinlei and He, Kaiming (2020). "Exploring Simple Siamese Representation Learning". In: *CoRR* abs/2011.10566. arXiv: 2011.10566. URL: https://arxiv.org/abs/2011.10566 (cit. on p. 45).

Chen, Yushi et al. (2014). "Deep Learning-Based Classification of Hyperspectral Data". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.6, pp. 2094–2107. DOI: 10.1109/JSTARS.2014.2329330 (cit. on p. 32).

Cheng, Gong and Han, Junwei (2016). "A Survey on Object Detection in Optical Remote Sensing Images". In: *CoRR* abs/1603.06201. arXiv: 1603.06201. URL: http://arxiv.org/abs/1603.06201 (cit. on p. 32).

Ciocarlan, Alina et al. (2024). *Self-Supervised Learning for Real-World Object Detection: a Survey*. arXiv: 2410.07442 [cs.CV]. URL: https://arxiv.org/abs/2410.07442 (cit. on p. 134).

Cornegruta, Savelie et al. (2016). "Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks". In: *CoRR* abs/1609.08409. arXiv: 1609.08409. URL: http://arxiv.org/abs/1609.08409 (cit. on pp. 28, 88).

Cover, T. and Hart, P. (1967). "Nearest neighbor pattern classification". In: *IEEE Transactions on Information Theory* 13.1, pp. 21–27. DOI: 10.1109/TIT.1967.1053964 (cit. on p. 23).

Cuturi, Marco (2013). "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf (cit. on p. 44).

Dell'Acqua, Fabio et al. (2018). "A Novel Strategy for Very-Large-Scale Cash-Crop Mapping in the Context of Weather-Related Risk Assessment, Combining Global Satellite Multispectral Datasets, Environmental Constraints, and In Situ Acquisition of Geospatial Data". In: *Sensors* 18.2. ISSN: 1424-8220. DOI: 10.3390/s18020591. URL: https://www.mdpi.com/1424-8220/18/2/591 (cit. on p. 106).

Demir, Ilke et al. (2018). "DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images". In: *CoRR* abs/1805.06561. arXiv: 1805.06561. URL: http://arxiv.org/abs/1805.06561 (cit. on p. 4).

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38. ISSN: 00359246. URL: http://www.jstor.org/stable/2984875 (visited on 10/19/2023) (cit. on p. 23).

Devlin, Jacob et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805 (cit. on pp. 52, 107, 108).

Diakogiannis, Foivos I. et al. (2019). "ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data". In: *CoRR* abs/1904.00592. arXiv: 1904.00592. URL: http://arxiv.org/abs/1904.00592 (cit. on p. 32).

Dinan, Emily et al. (2018). "Wizard of Wikipedia: Knowledge-Powered Conversational agents". In: *CoRR* abs/1811.01241. arXiv: 1811.01241. URL: http://arxiv.org/abs/1811.01241 (cit. on p. 4).

Doersch, Carl, Gupta, Abhinav, and Efros, Alexei A. (2016). *Unsupervised Visual Representation Learning by Context Prediction*. arXiv: 1505.05192 [cs.CV] (cit. on p. 38).

Dosovitskiy, Alexey et al. (2020). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929. arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929 (cit. on p. 112).

Drusch, Manfred et al. (2012). "Sentinel-2: ESA's optical high-resolution mission for GMES operational services". In: *Remote sensing of Environment* 120, pp. 25–36 (cit. on p. 2).

Dumeur, Iris, Valero, Silvia, and Inglada, Jordi (Sept. 2024a). "Paving the way toward foundation models for irregular and unaligned Satellite Image Time Series". working paper or preprint. URL: https://hal.science/hal-04639033 (cit. on p. 125).

— (2024b). "Self-Supervised Spatio-Temporal Representation Learning of Satellite Image Time Series". In: *IEEE Journal of Selected Topics in Applied Earth Observations and*

Remote Sensing 17, pp. 4350–4367. DOI: 10.1109/JSTARS.2024.3358066 (cit. on pp. 125, 132).

Dwibedi, Debidatta et al. (2021). "With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations". In: *CoRR* abs/2104.14548. arXiv: 2104.14548. URL: https://arxiv.org/abs/2104.14548 (cit. on p. 51).

Ermolov, Aleksandr et al. (2020). "Whitening for Self-Supervised Representation Learning". In: *CoRR* abs/2007.06346. arXiv: 2007.06346. URL: https://arxiv.org/abs/2007.06346 (cit. on p. 47).

ESA (2021). *MSI Level-2A BOA Reflectance Product*. DOI: https://doi.org/10.5270/S2_-6eb6imz. URL: https://sentinels.copernicus.eu/web/sentinel/sentinel-data-access/sentinel-products/sentinel-2-data-products/collection-0-level-2a (cit. on pp. 4, 14, 59, 74, 106, 107).

Ester, Martin et al. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, pp. 226–231 (cit. on p. 23).

Evans, Talfan et al. (2024). *Data curation via joint example selection further accelerates multimodal learning*. arXiv: 2406.17711 [cs.LG]. URL: https://arxiv.org/abs/2406.17711 (cit. on p. 133).

F.R.S., Karl Pearson (1901). "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572. DOI: 10.1080/14786440109462720 (cit. on p. 37).

Fawaz, Hassan Ismail et al. (2019). "InceptionTime: Finding AlexNet for Time Series Classification". In: *CoRR* abs/1909.04939. arXiv: 1909.04939. URL: http://arxiv.org/abs/1909.04939 (cit. on pp. 29, 30, 88).

Gansbeke, Wouter Van et al. (2021). "Revisiting Contrastive Methods for Unsupervised Learning of Visual Representations". In: *CoRR* abs/2106.05967. arXiv: 2106.05967. URL: https://arxiv.org/abs/2106.05967 (cit. on p. 49).

Garnot, Vivien Sainte Fare and Landrieu, Loic (Oct. 2021). "Panoptic Segmentation of Satellite Image Time Series With Convolutional Temporal Attention Networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4872–4881 (cit. on p. 132).

Gidaris, Spyros, Singh, Praveer, and Komodakis, Nikos (2018). *Unsupervised Representation Learning by Predicting Image Rotations*. DOI: 10.48550/ARXIV.1803.07728. URL: https://arxiv.org/abs/1803.07728 (cit. on pp. 35, 38).

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press (cit. on pp. 36, 37).

Gorelick, Noel et al. (2017a). "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment*. DOI: 10.1016/j.rse.2017.06.031. URL: https://doi.org/10.1016/j.rse.2017.06.031 (cit. on p. 88).

— (2017b). "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment* 202. Big Remotely Sensed Data: tools, applications and experiences, pp. 18–27. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.

rse.2017.06.031. URL: https://www.sciencedirect.com/science/article/pii/S0034425717302900 (cit. on p. 17).

Grill, Jean-Bastien et al. (2020). "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning". In: *CoRR* abs/2006.07733. arXiv: 2006.07733. URL: https://arxiv.org/abs/2006.07733 (cit. on pp. 45, 103, 125).

Gu, Albert and Dao, Tri (2024a). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. URL: https://openreview.net/forum?id=AL1fq05o7H (cit. on p. 125).

— (2024b). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. arXiv: 2312.00752 [cs.LG]. URL: https://arxiv.org/abs/2312.00752 (cit. on p. 135).

Gu, Albert, Goel, Karan, and Ré, Christopher (2021). "Efficiently Modeling Long Sequences with Structured State Spaces". In: *CoRR* abs/2111.00396. arXiv: 2111.00396. URL: https://arxiv.org/abs/2111.00396 (cit. on p. 135).

Gui, Jie et al. (Dec. 2024). " A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends ". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 46.12, pp. 9052–9071. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2024.3415112. URL: https://doi.ieeecomputersociety.org/10.1109/TPAMI.2024.3415112 (cit. on p. 106).

He, Kaiming, Chen, Xinlei, et al. (2021). "Masked Autoencoders Are Scalable Vision Learners". In: *CoRR* abs/2111.06377. arXiv: 2111.06377. URL: https://arxiv.org/abs/2111.06377 (cit. on p. 134).

He, Kaiming, Fan, Haoqi, et al. (2019). "Momentum Contrast for Unsupervised Visual Representation Learning". In: *CoRR* abs/1911.05722. arXiv: 1911.05722. URL: http://arxiv.org/abs/1911.05722 (cit. on pp. 40, 41, 61, 66, 103, 125).

He, Kaiming, Zhang, Xiangyu, et al. (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385 (cit. on pp. 24, 68, 86).

— (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 33).

Helber, Patrick et al. (2019). "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12.7, pp. 2217–2226. DOI: 10.1109/JSTARS.2019.2918242 (cit. on pp. 58, 66).

Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeffrey (2015). "Distilling the Knowledge in a Neural Network". In: *NIPS Deep Learning and Representation Learning Workshop*. URL: http://arxiv.org/abs/1503.02531 (cit. on p. 45).

Hochreiter, Sepp and Schmidhuber, Jürgen (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 24).

Hong, Danfeng et al. (2021). "More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification". In: *IEEE Transactions on Geoscience*

and Remote Sensing 59.5, pp. 4340–4354. DOI: 10.1109/TGRS.2020.3016820 (cit. on p. 32).

Hu, Weilin et al. (2015). "Deep convolutional neural networks for hyperspectral image classification". In: *Journal of Sensors* 2015, p. 258619. DOI: 10.1155/2015/258619. URL: https://doi.org/10.1155/2015/258619 (cit. on p. 32).

Huang, Wei et al. (2015). "A New Pan-Sharpening Method With Deep Neural Networks". In: *IEEE Geoscience and Remote Sensing Letters* 12.5, pp. 1037–1041. DOI: 10.1109/LGRS.2014.2376034 (cit. on p. 32).

Hütt, Christoph, Waldhoff, Guido, and Bareth, Georg (2020a). "Fusion of Sentinel-1 with Official Topographic and Cadastral Geodata for Crop-Type Enriched LULC Mapping Using FOSS and Open Data". In: *ISPRS International Journal of Geo-Information* 9.2. ISSN: 2220-9964. DOI: 10.3390/ijgi9020120. URL: https://www.mdpi.com/2220-9964/9/2/120 (cit. on p. 74).

— (2020b). "Fusion of Sentinel-1 with Official Topographic and Cadastral Geodata for Crop-Type Enriched LULC Mapping Using FOSS and Open Data". In: *ISPRS International Journal of Geo-Information* 9.2. ISSN: 2220-9964. DOI: 10.3390/ijgi9020120. URL: https://www.mdpi.com/2220-9964/9/2/120 (cit. on p. 106).

Inglada, Jordi et al. (2015). "Operational high resolution land cover map production at the country scale using satellite image time series". In: *Remote Sensing* 7.12, pp. 18062–18086 (cit. on p. 3).

Jasechko, Scott et al. (2013). "Terrestrial water fluxes dominated by transpiration". In: *Science* 340.6130, pp. 179–182. DOI: 10.1126/science.1233389 (cit. on p. 2).

Kingma, Diederik P. and Ba, Jimmy (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980 (cit. on p. 23).

Koh, Jing Yu, Salakhutdinov, Ruslan, and Fried, Daniel (2023). "Grounding Language Models to Images for Multimodal Inputs and Outputs". In: *ICML* (cit. on p. 54).

Kondmann, Lukas et al. (2021). "DENETHOR: The DynamicEarthNET dataset for Harmonized, inter-Operable, analysis-Ready, daily crop monitoring from space". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. URL: https://openreview.net/forum?id=uUa4jNMLjrL (cit. on pp. 88, 108).

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., pp. 1097–1105 (cit. on p. 24).

Kussul, Nataliia et al. (2017). "Deep learning classification of land cover and crop types using remote sensing data". In: *IEEE Geoscience and Remote Sensing Letters* 14.5, pp. 778–782 (cit. on p. 2).

Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: 10.1109/5.726791 (cit. on p. 24).

Lessig, Christian et al. (2023). *AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning*. arXiv: 2308.13280 [physics.ao-ph]. URL: https://arxiv.org/abs/2308.13280 (cit. on p. 107).

Li, Shen et al. (2020a). "PyTorch Distributed: Experiences on Accelerating Data Parallel Training". In: *CoRR* abs/2006.15704. arXiv: 2006.15704. URL: https://arxiv.org/abs/2006.15704 (cit. on p. 24).

— (2020b). "PyTorch Distributed: Experiences on Accelerating Data Parallel Training". In: *CoRR* abs/2006.15704. arXiv: 2006.15704. URL: https://arxiv.org/abs/2006.15704 (cit. on p. 125).

Liu, Chenfang et al. (2022). "Multi-Source Remote Sensing Pretraining Based on Contrastive Self-Supervised Learning". In: *Remote Sensing* 14.18. ISSN: 2072-4292. DOI: 10.3390/rs14184632. URL: https://www.mdpi.com/2072-4292/14/18/4632 (cit. on p. 103).

Liu, Xiao et al. (2021). "Self-supervised Learning: Generative or Contrastive". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1. DOI: 10.1109/tkde.2021.3090866. URL: https://doi.org/10.1109%5C%2Ftkde.2021.3090866 (cit. on p. 75).

Loveland, T.R. and Belward, A.S. (1997). "The International Geosphere Biosphere Programme Data and Information System global land cover data set (DISCover)". In: *Acta Astronautica* 41.4. Developing Business, pp. 681–689. ISSN: 0094-5765. DOI: https://doi.org/10.1016/S0094-5765(98)00050-2. URL: https://www.sciencedirect.com/science/article/pii/S0094576598000502 (cit. on p. 2).

Luo, Jiansong et al. (2024). "Early Crop Identification Study Based on Sentinel-1/2 Images with Feature Optimization Strategy". In: *Agriculture* 14.7. ISSN: 2077-0472. DOI: 10.3390/agriculture14070990. URL: https://www.mdpi.com/2077-0472/14/7/990 (cit. on p. 106).

Ma, Lei et al. (2019). "Deep learning in remote sensing applications: A meta-analysis and review". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 152, pp. 166–177. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2019.04.015. URL: https://www.sciencedirect.com/science/article/pii/S0924271619301108 (cit. on p. 33).

Ma, Shuang et al. (2020). "Learning Audio-Visual Representations with Active Contrastive Coding". In: *CoRR* abs/2009.09805. arXiv: 2009.09805. URL: https://arxiv.org/abs/2009.09805 (cit. on p. 54).

MacQueen, J. B. (n.d.). In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L. M. Le Cam and J. Neyman (cit. on p. 23).

Main-Knorn, Magdalena et al. (2017). "Sen2Cor for Sentinel-2". In: *Image and Signal Processing for Remote Sensing XXIII*. Ed. by Lorenzo Bruzzone. Vol. 10427. International Society for Optics and Photonics. SPIE, p. 1042704. DOI: 10.1117/12.2278218. URL: https://doi.org/10.1117/12.2278218 (cit. on p. 61).

Makantasis, Konstantinos et al. (2015). "Deep supervised learning for hyperspectral data classification through convolutional neural networks". In: *2015 IEEE International*

Geoscience and Remote Sensing Symposium (IGARSS), pp. 4959–4962. DOI: 10.1109/IGARSS.2015.7326945 (cit. on p. 32).

Mañas, Oscar et al. (2021). "Seasonal Contrast: Unsupervised Pre-Training from Uncurated Remote Sensing Data". In: *CoRR* abs/2103.16607. arXiv: 2103.16607. URL: https://arxiv.org/abs/2103.16607 (cit. on pp. 52, 58, 63, 67, 68, 103).

Masci, Jonathan et al. (2011). "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction". In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Ed. by Timo Honkela et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 52–59. ISBN: 978-3-642-21735-7 (cit. on p. 37).

Meier, Uwe et al. (Jan. 2009). "The BBCH system to coding the phenological growth stages of plants-history and publications". In: *Journal für Kulturpflanzen* 61, pp. 41–52. DOI: 10.5073/JfK.2009.02.01 (cit. on pp. 74, 106).

Mikolov, Tomás et al. (2013). "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1301.3781 (cit. on p. 53).

Misra, Ishan and Maaten, Laurens van der (2019). "Self-Supervised Learning of Pretext-Invariant Representations". In: *CoRR* abs/1912.01991. arXiv: 1912.01991. URL: http://arxiv.org/abs/1912.01991 (cit. on p. 39).

Mnih, Andriy and Kavukcuoglu, Koray (2013). "Learning word embeddings efficiently with noise-contrastive estimation". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., pp. 2265–2273 (cit. on p. 40).

Mou, Lichao, Ghamisi, Pedram, and Zhu, Xiao Xiang (2017). "Deep Recurrent Neural Networks for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7, pp. 3639–3655. DOI: 10.1109/TGRS.2016.2636241 (cit. on p. 32).

NASA Earth Data: Land Surface (2024). Accessed on 2024-02-19. URL: https://earthdata.nasa.gov/learn/land-surface (cit. on p. 2).

Noroozi, Mehdi and Favaro, Paolo (2017). *Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles*. arXiv: 1603.09246 [cs.CV] (cit. on p. 38).

Oord, Aäron van den, Li, Yazhe, and Vinyals, Oriol (2018). "Representation Learning with Contrastive Predictive Coding". In: *CoRR* abs/1807.03748. arXiv: 1807.03748. URL: http://arxiv.org/abs/1807.03748 (cit. on pp. 40, 75).

OpenAI (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL] (cit. on p. 54).

ORNL DAAC (2018). *MODIS and VIIRS Land Products Global Subsetting and Visualization Tool*. en. DOI: 10.3334/ORNLDAAC/1379. URL: https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1379 (cit. on p. 14).

Pathak, Deepak et al. (2016). "Context Encoders: Feature Learning by Inpainting". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544. DOI: 10.1109/CVPR.2016.278 (cit. on p. 37).

Patnala, Ankit et al. (2023). "Generating Views Using Atmospheric Correction for Contrastive Self-Supervised Learning of Multispectral Images". In: *IEEE Geoscience and Remote Sensing Letters* 20, pp. 1–5. DOI: 10.1109/LGRS.2023.3274493 (cit. on p. 57).

— (2024). "Bi-modal contrastive learning for crop classification using Sentinel-2 and Planetscope". In: *Frontiers in Remote Sensing* 5. ISSN: 2673-6187. DOI: 10.3389/frsen.2024.1480101. URL: https://www.frontiersin.org/journals/remote-sensing/articles/10.3389/frsen.2024.1480101 (cit. on p. 73).

PBC, Planet Labs (2017). *Planet Application Program Interface: In Space for Life on Earth*. Planet. URL: https://api.planet.com (cit. on p. 4).

Purushwalkam, Senthil and Gupta, Abhinav (2020). "Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases". In: *CoRR* abs/2007.13916. arXiv: 2007.13916. URL: https://arxiv.org/abs/2007.13916 (cit. on pp. 49, 52, 59, 107).

Račič, Matej et al. (Aug. 2020). "APPLICATION OF TEMPORAL CONVOLUTIONAL NEURAL NETWORK FOR THE CLASSIFICATION OF CROPS ON SENTINEL-2 TIME SERIES". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B2-2020, pp. 1337–1342. DOI: 10.5194/isprs-archives-XLIII-B2-2020-1337-2020 (cit. on pp. 74, 106).

Radford, Alec et al. (2021). "Learning Transferable Visual Models From Natural Language Supervision". In: *CoRR* abs/2103.00020. arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020 (cit. on p. 133).

Ray, Deepak K. et al. (2022). "Crop harvests for direct food use insufficient to meet the UN's food security goal". In: *Nature Food* 3.5, pp. 367–374. DOI: 10.1038/s43016-022-00504-z. URL: https://doi.org/10.1038/s43016-022-00504-z (cit. on p. 106).

Rombach, Robin et al. (2021). "High-Resolution Image Synthesis with Latent Diffusion Models". In: *CoRR* abs/2112.10752. arXiv: 2112.10752. URL: https://arxiv.org/abs/2112.10752 (cit. on p. 54).

Rosenblatt, Frank (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. DOI: https://doi.org/10.1037/h0042519 (cit. on p. 19).

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536 (cit. on p. 23).

Rumora, Luka, Miler, Mario, and Medak, Damir (2020). "Impact of Various Atmospheric Corrections on Sentinel-2 Land Cover Classification Accuracy Using Machine Learning Classifiers". In: *ISPRS International Journal of Geo-Information* 9.4. ISSN: 2220-9964. DOI: 10.3390/ijgi9040277. URL: https://www.mdpi.com/2220-9964/9/4/277 (cit. on p. 59).

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cit. on pp. 4, 5, 23, 64).

Rußwurm, Marc, Lefèvre, Sébastien, and Körner, Marco (2019). "BreizhCrops: A Satellite Time Series Dataset for Crop Type Identification". In: *CoRR* abs/1905.11893. arXiv: 1905.11893. URL: http://arxiv.org/abs/1905.11893 (cit. on pp. 90, 91).

Sagheer, Alaa and Kotb, Mostafa (2019). *Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems*. URL: https://doi.org/10.1038/s41598-019-55320-6 (cit. on p. 37).

Samuel, A. L. (1959). "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3, pp. 210–229. DOI: 10.1147/rd.33.0210 (cit. on p. 19).

Scheibenreif, Linus et al. (2022). "Self-supervised Vision Transformers for Land-cover Segmentation and Classification". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*. IEEE, pp. 1421–1430. DOI: 10.1109/CVPRW56347.2022.00148. URL: https://doi.org/10.1109/CVPRW56347.2022.00148 (cit. on pp. 103, 106).

Schmitt, M. et al. (2019). "SEN12MS – A CURATED DATASET OF GEOREFERENCED MULTI-SPECTRAL SENTINEL-1/2 IMAGERY FOR DEEP LEARNING AND DATA FUSION". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W7, pp. 153–160. DOI: 10.5194/isprs-annals-IV-2-W7-153-2019. URL: https://isprs-annals.copernicus.org/articles/IV-2-W7/153/2019/ (cit. on pp. 4, 58, 132).

Schneider, Maja et al. (Sept. 2023). "EuroCrops: The Largest Harmonized Open Crop Dataset Across the European Union". In: *Scientific Data* 10.1, p. 612. ISSN: 2052-4463. DOI: 10.1038/s41597-023-02517-0. URL: https://doi.org/10.1038/s41597-023-02517-0 (cit. on p. 132).

Shelhamer, Evan, Long, Jonathan, and Darrell, Trevor (2016). "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1605.06211. arXiv: 1605.06211. URL: http://arxiv.org/abs/1605.06211 (cit. on p. 32).

Shwartz-Ziv, Ravid and LeCun, Yann (2023). *To Compress or Not to Compress- Self-Supervised Learning and Information Theory: A Review*. arXiv: 2304.09355 [cs.LG]. URL: https://arxiv.org/abs/2304.09355 (cit. on pp. 37, 38).

Somepalli, Gowthami et al. (2021). "SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training". In: *CoRR* abs/2106.01342. arXiv: 2106.01342. URL: https://arxiv.org/abs/2106.01342 (cit. on p. 103).

Stone, T A et al. (May 1994). "Map of the vegetation of South America based on satellite imagery". In: *Photogrammetric Engineering and Remote Sensing* 60.5. URL: https://www.osti.gov/biblio/53459 (cit. on p. 2).

Sumbul, Gencer et al. (2019). "BigEarthNet: A Large-Scale Benchmark Archive For Remote Sensing Image Understanding". In: *CoRR* abs/1902.06148. arXiv: 1902.06148. URL: http://arxiv.org/abs/1902.06148 (cit. on pp. 4, 58, 64, 66, 132).

Szegedy, Christian et al. (2014). "Going Deeper with Convolutions". In: *CoRR* abs/1409.4842. arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842 (cit. on p. 29).

Szwarcman, Daniela et al. (2025). *Prithvi-EO-2.0: A Versatile Multi-Temporal Foundation Model for Earth Observation Applications*. arXiv: 2412.02732 [cs.CV]. URL: https://arxiv.org/abs/2412.02732 (cit. on p. 134).

Tao, Yumeng et al. (2016). "A deep neural network modeling framework to reduce bias in satellite precipitation products". In: *Journal of Hydrometeorology* 17.3, pp. 931–945. DOI: 10.1175/JHM-D-15-0075.1. URL: https://doi.org/10.1175/JHM-D-15-0075.1 (cit. on p. 33).

Tateishi, Ryutaro and Kajiwara, Koji (1991). "Land cover monitoring in Asia by NOAA GVI data". In: *Geocarto International* 6.4, pp. 53–64. DOI: 10.1080/10106049109354340. eprint: https://doi.org/10.1080/10106049109354340. URL: https://doi.org/10.1080/10106049109354340 (cit. on p. 2).

Tejankar, Ajinkya, Koohpayegani, Soroush Abbasi, Navaneet, K. L., et al. (2021). "Constrained Mean Shift Using Distant Yet Related Neighbors for Representation Learning". In: *CoRR* abs/2112.04607. arXiv: 2112.04607. URL: https://arxiv.org/abs/2112.04607 (cit. on p. 52).

Tejankar, Ajinkya, Koohpayegani, Soroush Abbasi, Pillai, Vipin, et al. (2020). "ISD: Self-Supervised Learning by Iterative Similarity Distillation". In: *CoRR* abs/2012.09259. arXiv: 2012.09259. URL: https://arxiv.org/abs/2012.09259 (cit. on p. 48).

Tian, Yonglong et al. (2020). "What makes for good views for contrastive learning". In: *CoRR* abs/2005.10243. arXiv: 2005.10243. URL: https://arxiv.org/abs/2005.10243 (cit. on p. 40).

Tong, Xin-Yi et al. (2018). "Learning Transferable Deep Models for Land-Use Classification with High-Resolution Remote Sensing Images". In: *CoRR* abs/1807.05713. arXiv: 1807.05713. URL: http://arxiv.org/abs/1807.05713 (cit. on p. 33).

Tseng, Gabriel et al. (2024). *Lightweight, Pre-trained Transformers for Remote Sensing Timeseries*. arXiv: 2304.14065 [cs.CV]. URL: https://arxiv.org/abs/2304.14065 (cit. on p. 132).

Tucker, CJ, Townshend, JR, and Goff, TE (1985). "African land-cover classification using satellite data". In: *Science* 227.4685, pp. 369–375. DOI: 10.1126/science.227.4685.369 (cit. on p. 2).

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (cit. on pp. 3, 7, 29, 31, 32, 53, 88).

Vescovo, Loris et al. (2012). "New spectral vegetation indices based on the near-infrared shoulder wavelengths for remote detection of grassland phytomass". In: *International Journal of Remote Sensing* 33.7, pp. 2178–2195. DOI: 10.1080/01431161.2011.607195. URL: https://doi.org/10.1080/01431161.2011.607195 (cit. on p. 59).

Vincent, Pascal et al. (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion." In: *J. Mach. Learn. Res.* 11, pp. 3371–3408. URL: http://dblp.uni-trier.de/db/journals/jmlr/jmlr11.html#VincentLLBM10 (cit. on pp. 37, 38).

Waldner, François and Diakogiannis, Foivos I. (2019). "Deep learning on edge: extracting field boundaries from satellite images with a convolutional neural network". In: *CoRR*

abs/1910.12023. arXiv: 1910.12023. URL: http://arxiv.org/abs/1910.12023 (cit. on p. 134).

Wang, Meng et al. (2023). "Nearest Neighbor-Based Contrastive Learning for Hyperspectral and LiDAR Data Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 61, pp. 1–16. DOI: 10.1109/TGRS.2023.3236154 (cit. on p. 51).

Wei, Yancong et al. (2017). "Boosting the accuracy of multi-spectral image pan-sharpening by learning a deep residual network". In: *CoRR* abs/1705.07556. arXiv: 1705.07556. URL: http://arxiv.org/abs/1705.07556 (cit. on p. 32).

Weikmann, Giulio, Paris, Claudia, and Bruzzone, Lorenzo (2021). "TimeSen2Crop: A Million Labeled Samples Dataset of Sentinel 2 Image Time Series for Crop-Type Classification". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14, pp. 4699–4708. DOI: 10.1109/JSTARS.2021.3073965 (cit. on p. 132).

Xiang, Jinyong et al. (2020). "Multi-time scale wind speed prediction based on WT-bi-LSTM". In: *MATEC Web Conf.* 309, p. 05011. DOI: 10.1051/matecconf/202030905011. URL: https://doi.org/10.1051/matecconf/202030905011 (cit. on p. 28).

Xiao, Tete et al. (2021). "What Should Not Be Contrastive in Contrastive Learning". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=CZ8Y3NzuVzO (cit. on pp. 43, 44, 59, 68).

Yalniz, I. Zeki et al. (2019). "Billion-scale semi-supervised learning for image classification". In: *CoRR* abs/1905.00546. arXiv: 1905.00546. URL: http://arxiv.org/abs/1905.00546 (cit. on p. 4).

Yan, King-Yin (2022). "AGI via Combining Logic with Deep Learning". In: *Artificial General Intelligence*. Ed. by Ben Goertzel, Matthew Iklé, and Alexey Potapov. Cham: Springer International Publishing, pp. 327–343. ISBN: 978-3-030-93758-4 (cit. on p. 30).

Yang, Xingyi et al. (2020). "Transfer Learning or Self-supervised Learning? A Tale of Two Pretraining Paradigms". In: *CoRR* abs/2007.04234. arXiv: 2007.04234. URL: https://arxiv.org/abs/2007.04234 (cit. on p. 75).

Yao, Jing et al. (2024). *SpectralMamba: Efficient Mamba for Hyperspectral Image Classification*. arXiv: 2404.08489 [cs.CV]. URL: https://arxiv.org/abs/2404.08489 (cit. on pp. 103, 125).

YM., Asano, C., Rupprecht, and A., Vedaldi (2020). "Self-labelling via simultaneous clustering and representation learning". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Hyx-jyBFPr (cit. on p. 44).

Yoon, Jinsung et al. (2020). "VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 11033–11043. URL: https://proceedings.neurips.cc/paper/2020/file/7d97667a3e056acab9aaf653807b4a03-Paper.pdf (cit. on p. 103).

Yuan, Qiangqiang et al. (2017). "A Multi-Scale and Multi-Depth Convolutional Neural Network for Remote Sensing Imagery Pan-Sharpening". In: *CoRR* abs/1712.09809. arXiv: 1712.09809. URL: http://arxiv.org/abs/1712.09809 (cit. on p. 32).

Yuan, Xin et al. (2021). "Multimodal Contrastive Training for Visual Representation Learning". In: *CoRR* abs/2104.12836. arXiv: 2104.12836. URL: https://arxiv.org/abs/2104.12836 (cit. on p. 75).

Zbontar, Jure et al. (2021). "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". In: *CoRR* abs/2103.03230. arXiv: 2103.03230. URL: https://arxiv.org/abs/2103.03230 (cit. on pp. 46, 103).

Zhai, Xiaohua et al. (2023). *Sigmoid Loss for Language Image Pre-Training*. arXiv: 2303.15343 [cs.CV]. URL: https://arxiv.org/abs/2303.15343 (cit. on p. 133).

Zhang, Richard, Isola, Phillip, and Efros, Alexei A. (2016a). *Colorful Image Colorization*. DOI: 10.48550/ARXIV.1603.08511. URL: https://arxiv.org/abs/1603.08511 (cit. on p. 37).

— (2016b). *Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction*. DOI: 10.48550/ARXIV.1611.09842. URL: https://arxiv.org/abs/1611.09842 (cit. on p. 37).

Zhao, Yanli et al. (2023). *PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel*. arXiv: 2304.11277 [cs.DC]. URL: https://arxiv.org/abs/2304.11277 (cit. on p. 24).

Zhao, Zhong-Qiu et al. (2018). "Object Detection with Deep Learning: A Review". In: *CoRR* abs/1807.05511. arXiv: 1807.05511. URL: http://arxiv.org/abs/1807.05511 (cit. on p. 32).

Zheng, Mingkai et al. (2021). "ReSSL: Relational Self-Supervised Learning with Weak Augmentation". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. URL: https://openreview.net/forum?id=ErivP29kYnx (cit. on p. 48).

Zhuang, Fuzhen et al. (2019). "A Comprehensive Survey on Transfer Learning". In: *CoRR* abs/1911.02685. arXiv: 1911.02685. URL: http://arxiv.org/abs/1911.02685 (cit. on p. 35).

Abbreviations

IGBP International Geosphere-Biosphere Program

NASA National Aeronautics and Space Administration

AVHRR Advanced Very High-Resolution Radiomete

MODIS Moderate Resolution Imaging Spectrora- diometer

ESA European Space Agency

SAR synthetic aperture radar

CNN Convolutional neural networks

DBN deep belief network

NLP Natural Language Processing

LLM Large Language Model

BERT Bidirectional Encoder Representations from Transformers

LSTM Long Short-Term Memory

NIR Near infrared

GEE Google Earth Engine

SWIR Shortwave infrared

NDVI Normalized Difference Vegetation Index

API Application programming interface

UTM Universal Transverse Mercator

ML Machine leaning

PDE Partial differential equation

MLP multi-layer perceptron

tanh hyperbolic tangent

ReLU rectified linear unit

MSE mean squared error

FSDP Fully Sharded Data Parallelism

OBIA object-based image analysis

PCA principal component analysis

NYU New York University

SimCLR Simple framework for contrastive learning of visual representations

MoCo momentum contrast

SeLa self-labelling

SwAV swapped assignment between multiple views

EMA Exponential Moving Average

BYOL Bootstrap your own latent

DiNo Self-distillation with no labels

W-MSE Whitening mean squared error

KL Kullback-Leibler

ReSSL Relational Self-Supervised Learning

ISD Iterative Similarity Distillation

CBOW continuous bag of words

MLM mask language modelling

NSP next sentence prediction

NCE Noise Contrastive Estimation

FIFO First In First Out

LOOC Leave one out contrastive

SeCo Seasonal contrast

GPT Generative pre-trained transformer

TOA Top of Atmosphere

BOA Bottom of Atmosphere

SCL Scene classification

AC Atmospheric correction

AOT Aerosol optical thickness

RNN Recurrent neural network

BPTT Backpropagation Through Time

SCARF Self-Supervised Contrastive Learning using Random Feature Corruption

CLC Corine Land Cover

SigLIP sigmoid loss for language image pre-training

CLIP contrastive language image pre-training

JEST Joint example selection

SSM State Space model

Acknowledgements

I would like to express my gratitude to all those who have contributed to the completion of my PhD thesis. First, I would like to thank my first supervisor Prof. Dr. Martin Schultz for his guidance, support, and encouragement throughout the research process. Without his initial research idea and ongoing assistance, this work would not have been possible. I would also like to extend my thanks to Prof. Dr. Juergen Gall, my second supervisor, for his invaluable feedback and insights throughout the course of my research work. Additionally, I would like to thank Dr. Scarlet Stadtler for her scientific and personal supervision, as well as her companionship, which has been essential in helping me navigate the challenges of my PhD journey. I would also like to express my gratitude to Dr. Michael Langguth for his thorough proofreading and the invaluable feedback he provided. I also want to express my appreciation to the Earth System Data Exploration group, our colleagues in the KISTE consortium, and other collaborators from different research projects for their support and friendship to make my PhD time memorable. I would also acknowledge Perplexity Standard, a LLM based search engine tool, used to rephrase and refine several texts in this thesis. Last but not least, I would like to thank my parents, my siblings, my partner, and my friends for their encouragement, and unwavering support.