**UiT** The Arctic University of Norway

Faculty of Science and Technology
Department of Physics and Technology

# Scalable computing for Earth observation

## Application on Sea Ice analysis

Salman Khaleghian

A dissertation for the degree of Doctor of Philosophy (PhD), December 2022

This thesis document was typeset using the *UiT Thesis LaTeX Template*.
© 2022 – `http://github.com/egraff/uit-thesis`

“Work for a better life as if you live forever, and work for a better end as if you die tomorrow.”

–Ali (pbuh)

# Abstract

In recent years, Deep learning (DL) networks have shown considerable improvements and have become a preferred methodology in many different applications. These networks have outperformed other classical techniques, particularly in large data settings. In Earth observation from the satellite field, for example, DL algorithms have demonstrated the ability to learn complicated nonlinear relationships in input data accurately. Thus, it contributed to advancement in this field. However, the training process of these networks has heavy computational overheads. The reason is two-fold: The sizable complexity of these networks and the high number of training samples needed to learn all parameters comprising these architectures. Although the quantity of training data enhances the accuracy of the trained models in general, the computational cost may restrict the amount of analysis that can be done. This issue is particularly critical in satellite remote sensing, where a myriad of satellites generate an enormous amount of data daily, and acquiring in-situ ground truth for building a large training dataset is a fundamental prerequisite.

This dissertation considers various aspects of deep learning based sea ice monitoring from SAR data. In this application, labeling data is very costly and time-consuming. Also, in some cases, it is not even achievable due to challenges in establishing the required domain knowledge, specifically when it comes to monitoring Arctic Sea ice with Synthetic Aperture Radar (SAR), which is the application domain of this thesis. Because the Arctic is remote, has long dark seasons, and has a very dynamic weather system, the collection of reliable in-situ data is very demanding. In addition to the challenges of interpreting SAR data of sea ice, this issue makes SAR-based sea ice analysis with DL networks a complicated process.

We propose novel DL methods to cope with the problems of scarce training data and address the computational cost of the training process. We analyze DL network capabilities based on self-designed architectures and learn strategies, such as transfer learning for sea ice classification. We also address the scarcity of training data by proposing a novel deep semi-supervised learning method based on SAR data for incorporating unlabeled data information into the training process. Finally, a new distributed DL method that can be used in a semi-supervised manner is proposed to address the computational complexity of deep neural network training.

# Acknowledgements

I would like to thank my supervisors, Andrea Marinoni, Einar Broch Johnsen, and Anders Andersen. Thank you for your time and energy, and your continuous motivation and encouragement during my PhD studies. It was a great opportunity for me to learn from all of you during this multidiscipline PhD project. Andrea, thank you for your constant support in the course of PhD.

I would also like to thank all members of the Earth observation group at UiT and the people at CIRFA. I learned many things from you through scientific and technical discussions.

I would like to state my appreciation to CIRFA center leader, Torbjørn Eltoft, who founded this project and gave me this opportunity to research interesting areas. I will not forget your support during all the ups and downs of my PhD.

Finally, I would like to state my heartfelt gratitude to my family for their unending support and presence, Thank you to my wife Shima, my daughters Kosar and Tasnim, my son Aliakbar and my parents even from far away.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**CNN** Convolutional Neural Network

**DDL** Distributed Deep Learning

**DL** Deep Learning

**DNN** Deep Neural Network

**EM** Electromagnetic

**EO** Earth Observation

**ESA** European Space Agency

**EW** Extra Wide Swath

**FC** Fully Connected

**GANs** Generative Adversarial Networks

**IA** Incident Angle

**IR** Infrared

**ISERV** Station SERVIR Environmental Research and Visualization System

**IW** Interferometric Wide Swath

**LULC** Land cover/land use mapping:and-use and land cover

**MIR** Mid-infrared

**NASA** National Aeronautics and Space Administration

**NIR** Near-infrared

**NIS** Norwegian Meteorological Institute

**RGB** Red, Green, Blue

**RS** Remote Sensing

**SAR** Synthetic Aperture Radar

**SGD** Stochastic Gradient Descent

**SM** Stripmap

**SOD** Stage of Development

**SRT** Shuttle Radar Topography

**SRTM** Shuttle Radar Topography Mission

**SSL** Semi-Supervised Learning

**TIR** Thermal Infrared

**UiT** University of Tromsø

**VAE** Variational Autoencoders

**WHO** World Health Organization

**WMO** World Meteorological Organization

**WV** Wave

# 1

# Introduction

In the last decade, Deep Neural Networks (DNNs) have shown remarkable performance in tackling various challenging machine learning problems [1]. Practically, a DNN may have hundreds of layers and millions of parameters. It has been demonstrated that deep neural networks outperform alternative methods, especially in big data problems [1]. Despite this, training a deep network architecture is a computationally expensive task. The reason for this costly computation is the high amount of data points to be handled and the complexity of the network structures [2]. In general, a larger training data set will improve the accuracy of the trained models. However, the computing cost of the training data may limit the amount of analysis that can be performed [3].

DNNs have been employed to address many Remote Sensing (RS) and Earth Observation (EO) challenges, and they have shown great success in solving a variety of satellite-based RS image analysis tasks, including land cover classification, object detection, and change detection [4]. Satellite images, which constitute a significant data source for Earth observation, allow us to measure and observe intricate features on the surface of the Earth. The amount of satellite images is rapidly increasing as a result of the development in spaceborne Earth observation technologies [5]. It is no accident that this field is now referred to as big data. The National Aeronautics and Space Administration (NASA)'s Landsat [6], and European Space Agency (ESA)'s Copernicus [7], respectively, offer high revisiting frequency data and data with large spectral-spatial coverage, allowing for near-real-time worldwide surveillance of the

Earth surface. Indeed, Copernicus is presently the world's biggest single EO program, with its fleet of Sentinel spacecraft.

To conduct large-scale, high-frequency monitoring of the Earth using deep learning architectures, we need scalable computing to train the models using a substantial quantity of labeled data [8]. However, these massive amounts of training data do not always exist, which is the case for the application focused in this thesis. In this dissertation, we consider various aspects of the deep learning-based analysis of Arctic Sea ice from satellite-based synthetic aperture radar (SAR) data. This application is challenging. The Arctic is remote, has long dark seasons and a very dynamic weather system, and the collection of reliable in-situ data is very demanding. In addition, SAR images of sea ice are inherently difficult to interpret and require extensive time and resources. Scarce training data is exacerbated when the trained model is to examine the dynamic sea ice in an Arctic-wide setting, dealing with a variety of seasonal and meteorological circumstances.

This thesis investigates three main topics in deep learning–based sea ice monitoring from SAR (Figure 1.1): deep learning architecture design, semi-supervised learning to cope with the training data, distributed systems, and high-performance computing.

- **Deep learning architectures design:** Deep neural networks are a holistic learning architecture for feature extraction and classification. We consider how different deep learning architectures cope with the sea ice classification task.

- **Semi-supervised learning:** We investigate several methods of label propagation and advance deep semi-supervised learning approaches to address the scarce training data problem in sea ice classification.

- **Scalable computing:** Distributed deep learning is considered to address computation complexity along with specific challenges when deep neural networks are trained on big Earth observation data.

In the analysis, we specifically consider the challenges associated with the properties of SAR data like scattering ambiguities, type-dependent incidence angle slopes, and the annoying additive noise pattern of Sentinel-1 SAR data, which are the principal data used in this study.

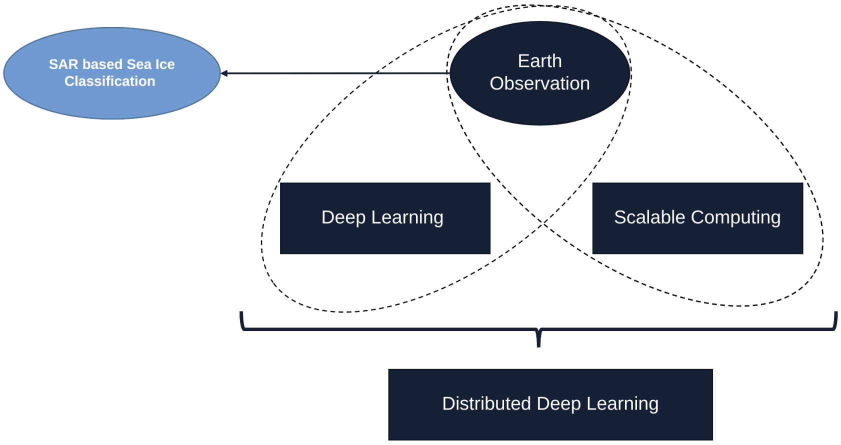


**Figure 1.1:** Dissertation Research areas.

This section briefly discusses the benefits of deep learning and its advantage for big data analysis, particularly for Earth observation applications. We describe the distributed deep learning setup and explain how this provides a scalable computing framework for deep learning analysis. Then, some major Earth observation applications are briefly listed to contextualize our chosen application. We explain our primary application, namely sea ice analysis from SAR, and add some perspectives on how the results may be exploited in operational sea ice charting. Finally, the thesis's objectives and structure are outlined.

## 1.1 Deep Learning

Deep learning models as feature learning hierarchies extract multiple layers of non-linear features and feed them to a classifier that integrates all the features to produce predictions. Hierarchically, DL algorithms directly learn the representative and discriminative features from the data. It is different from manual feature learning, which performs manual selection and extraction of features for each task. Features are automatically learned in a deep learning method to optimize the model's performance [9].

Thanks to DL theory [9], unsupervised feature learning from immense raw-image data sets has become conceivable. This unsupervised feature learning provides an alternative technique that autonomously learns practical features from the training set [10]. More specifically, when a large amount of data is available, it has been shown that deep neural networks perform better than other learning approaches [3], as we see in Figure 1.2.

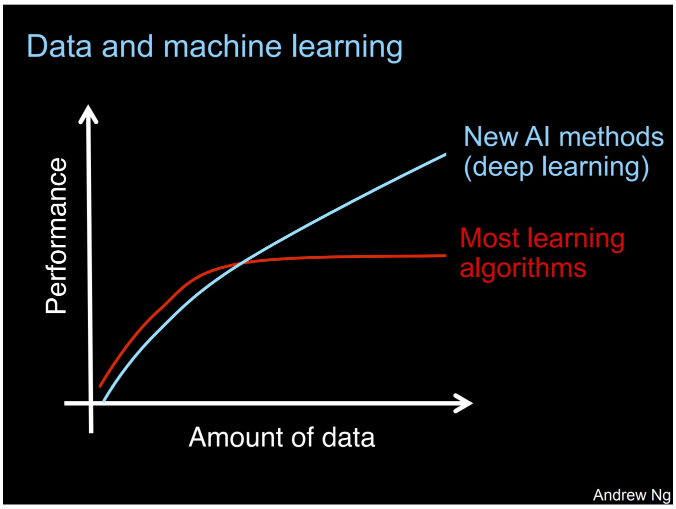


**Figure 1.2:** Deep learning performs better in comparison of other methods when available data is increased [3].

For example, Convolutional Neural Network (Convolutional Neural Network (CNN)) [9] is a well-known deep learning model that has been used to solve several computer vision tasks. A CNN employs convolutional layers to extract valuable information from the inputs. These convolutional layers contain learned parameters, allowing the filters to automatically extract the most valuable information for the task [9]. More details on DL and CNNs will be discussed in Chapter 3. Figure 1.3 shows a traffic sign image filtered by four convolutional kernels, which create four feature maps; These feature maps are sub-sampled by max pooling. The next layer applies ten convolutional kernels to these sub-sampled images. The final layer is a FC layers where all generated features are combined and used in the classifier.

### 1.1.1 Deep Learning for Earth Observation

DL has been utilized in Remote Sensing (RS) and Earth Observation (EO) for tasks ranging from image preprocessing, pixel-based classification, patch-based classification, and target recognition to high-level semantic feature extraction and RS scene interpretation [11]. In fact, deep learning is a novel and fascinating method that potentially can be the next step in the evolution of Earth observation and remote sensing image processing [11].

Extracting useful information from diverse forms of remote sensing data, and coping with ever-increasing data types and volumes, is a significant problem in

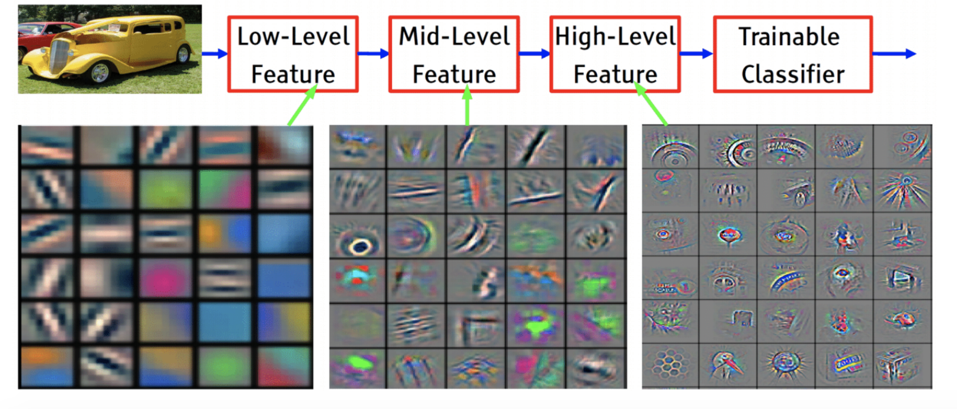


**Figure 1.3:** Example of learned features in each layer for a object detection task (from Yann LeCun, 2015).

Earth observation analysis [11]. Traditional techniques use feature engineering from RS images to build extracted and selected features to feed to different classification/regression models. Handcrafted features are shown to be successful in representing several spectral, textural, and geometrical properties of images [12, 13]. However, because these features cannot readily reflect the complexities of the actual data statistics, they cannot attain an ideal balance between discriminability and resilience. The dilemma is exacerbated when dealing with a large amount of remote sensing image data since imaging conditions fluctuate rapidly and may change dramatically in a short period [11].

### 1.1.2 Deep Learning Architectures Design

Different deep learning network architectures specially have been proposed for addressing computer vision issues. [14]. In computer vision, DL architectures have been made to work best for specific object detection and recognition by optimizing number layers, number filters, and many network hyperparameters. Optimizing the best architecture for new tasks, like RS and EO problems, is challenging. This issue is even more significant for more specific uses, such as SAR-based monitoring of sea ice in the Arctic.

When looking into DNN architectures, we can usually consider two main ways to deal with this problem. The first approach is to analyze the problem by making a model of a custom or ad hoc architecture. An ad hoc architecture is interesting because it is very flexible. However, it usually needs to have a lot of hyper-parameters optimized. The second approach relies on the use of an existing architecture that can either be fine-tuned by already trained parameters or trained from scratch. This approach reduces the time needed for the design of the deep learning architecture. For instance, we have employed this strategy in Chapter 6.

### 1.1.3 Scarce Training Data

Deep neural networks have been employed in many RS and EO data analysis challenges and showed the promising results [15]. They are well-known for their high efficiency and test-time performance. The disadvantage is that a significant number of training samples must be available to train the models. Also, these samples are in most cases labeled by humans. This issue is more serious in the case of the RS and EO domains since acquiring in-situ ground truth are very expensive, time-demanding, and often impossible. This issue becomes more significant in Arctic applications, where the volume of validated labeled data is typically low. The remoteness, long dark seasons, and exceedingly variable weather conditions make it difficult to acquire ground truth in these areas.

The scarcity of training data is well-known in the machine learning and deep learning domains and is recognized as a significant issue in big data applications [16]. New advanced approaches in deep semi-supervised learning, deep unsupervised learning, and deep self-supervised learning have been proposed to overcome this issue [17, 16]. More details on semi-supervised learning, one of our focus areas, will be discussed in Chapter 3.

### 1.1.4 Computational Complexity

In DL, increasing the quantity of training datasets often improves model performance (e.g., classification accuracy) [18, 19]. Nonetheless, as data amount and model complexity rise, the training process of DL is computationally costly and time-consuming. For instance, training a state-of-the-art ResNet-50 [20] model (in 90 epochs) on the ImageNet dataset [21] using the most recent Nvidia Tesla V100 GPU takes around two days [22]. In general, one must tweak the hyperparameters for certain task, which takes a great deal of effort and is necessary to get acceptable performance.

Figure 1.4 quantifies the computing requirements of frequently used deep learning models [23]. As shown in the graph, the highest performing trained architectures are those with very high computational complexity (such as NASNet-A-Large), which are located at the far right of the graph. In addition to this, it should be noted that they are not the ones with the most model complexity (as is evidenced by the size of the bubble). Due to the high compu-

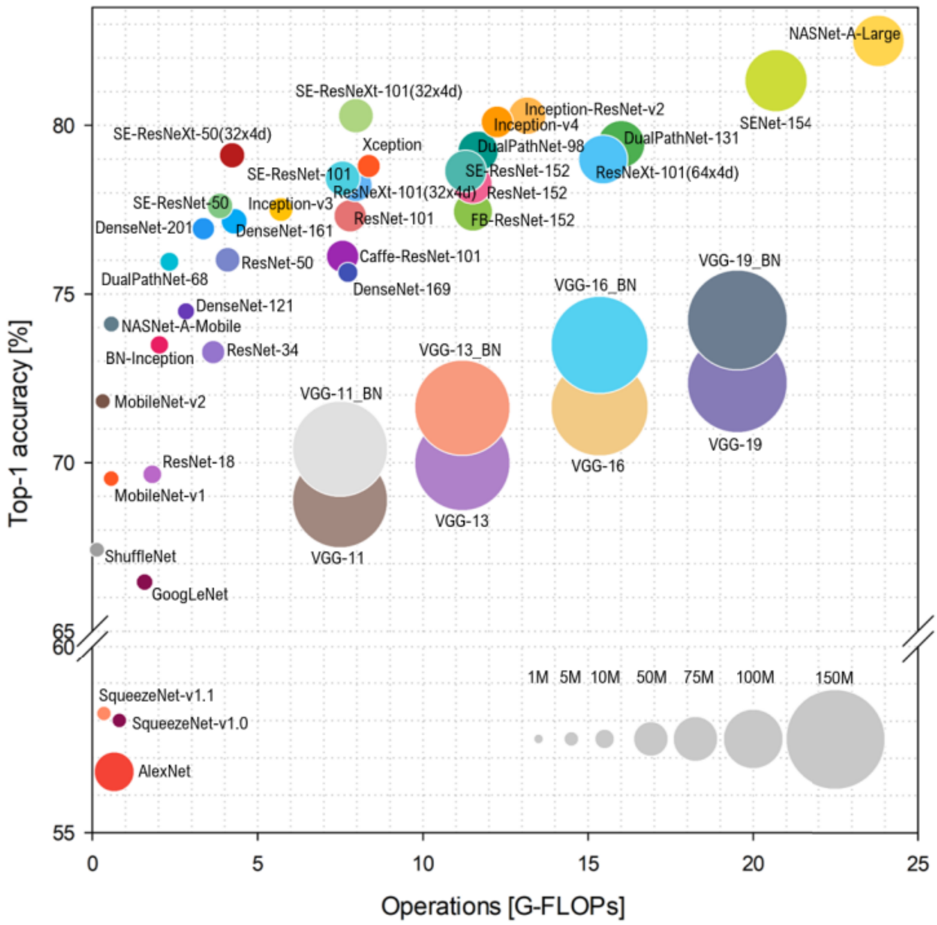


**Figure 1.4:** Computational complexity. The Top-1 accuracy versus the floating-point operations (FLOPs) needed for a single forward pass. The size of each ball depends model complexity [23].

tational cost of iterative DL training across a large quantity of data, substantial computer resources are required. As a result, single machines are not always capable of performing this job in the time provided.

## 1.2 Distributed Deep Learning for Big Data Analysis

The ascent in popularity of DNNs is closely tied to the amount of accessible processing power, which has made it possible to harness the fruits of the intrinsic parallelism of these networks [24]. Deep learning's computational

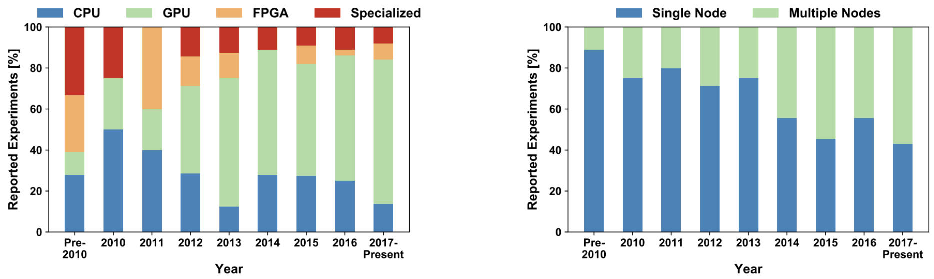


**Figure 1.5:** Parallel Architectures in Deep Learning using different hardware acceleration and single or multi nodes systems [24].

intensity and memory requirements rise in direct proportion to the size of the available datasets and the level of DNNs complexity. Training a DNN to an accuracy that is competitive in today's market is almost impossible without a high-performance computer cluster [25, 24]. Different components of training and inference (evaluation) of DNNs are adjusted to boost concurrency to use such systems.

In modern computer architectures, parallelism can be found both internally on the chip in the form of pipelining and out-of-order execution, as well as in the form of multi-core or multi-socket systems. Multi-core computers may be designed with either multiple processes, which use distinct memory domains, or multiple threads, which use shared memory domains. Alternatively, a combination of the two is also possible. The primary distinction between these two lies in the fact that multi-process parallel programming requires the programmer to think about the distribution of the data as a first-class concern, whereas multi-threaded programming only requires the programmer to focus about the parallelism and allows the hardware system to handle the data shuffling (typically through hardware cache-coherence protocols).

The process of training large-scale models requires a significant amount of computational resources. Therefore, single machines are not always capable of completing this work within the allotted amount of time. The computation might be split up and carried out simultaneously on several different computers all linked together through a network. Distributed deep learning by leveraging the computational resources of multiple devices (e.g., multiple GPUs) [24] is used to accelerate the training process of DNNs when working with a large amount of data.

Figure 1.5 provides a summary of the machine architectures that have been employed in technical literature. There is a discernible shift toward GPUs, which are the focus of the majority of articles starting from 2013. Despite this, even the most accelerated nodes are not enough to handle the massive computational demand. Figure 1.5 illustrates how the multi-node parallelism in those activities is rapidly expanding. Different approaches have been proposed to train deep learning models on multi-GPUs in distributed environments. More details on different approaches will be discussed in Chapter 4.

### 1.2.1 Communication Overhead

Latency, bandwidth, and message rate are the three most significant metrics for the interconnection network [24]. There are many performance measures available across the various network technologies. InfiniBand, for instance, provides much shorter latencies and more message rates than current Ethernet, even though both provide enormous capacity [26]. Interconnection networks designed specifically for high-performance computing may yield better results in all three performance criteria. However, communication via a network is often slower than the communication that occurs inside a single computer. In distributed deep learning, regardless of networking technology, the communication overhead directly affects the capability of distributed training [24]. This effect is because a large number of model parameters (full or partial) should be sent over the network in each training iteration. This issue will be discussed in more detail in Chapter 4.

### 1.2.2 Earth Observation as a Big Data Problem

International space agencies, like ESA and NASA, adhere to an open data policy and make available an enormous amount of multi-sensor data for free daily. Because of the rapid technological advancement that has been incorporated into Remote Sensing (RS) optical and microwave sensor technologies [5], the systems have made significant strides forward in the last several decades. In this sense, it is not a coincidence that remote sensing data are now being described using the big data terminology, which includes characteristics such as volume, velocity, variety, veracity, and value [27, 8].

Copernicus is the European Union's program [7] for Environmental monitoring. It is made up of a collection of systems that receive data from satellites and in-situ sensors, process and interprets this data, and then offer users accurate and up-to-date information on a variety of environmental and security concerns. The Sentinel satellites, in orbit for the particular purposes of the Copernicus program, and other contributing satellite missions run by national or international organizations, support Copernicus with data. Access to Sentinel data is governed by EU legislation and is complete, open, and unrestricted. Copernicus information is made accessible to consumers via Copernicus services covering six theme areas: land, marine, atmosphere, climate, emergency, and security. The Copernicus program is a vanguard of the Big Data paradigm resulting from the data and information processed and distributed. It also gives rise to the so-called five V's paradigm, briefly discussed below [5]:

- **Volume:** The European Space Agency's (ESA) Sentinel product repository has published over 5 million products to date, It has over 100000 users who have downloaded more than 50 PB of data since the system's inception. As additional Sentinel satellites are being launched, this volume will grow in the coming years.

- **Velocity:** The Copernicus data must be sent and processed quickly to provide 24/7 services to users that want immediate information. By the end of 2016, six TB of data had been generated, with 100 TB of data being broadcasted daily from the Sentinel product repository. As additional Sentinel satellites are being launched, these rates will rise in the following years.

- **Variety:** The Sentinel satellites have various types of sensors (e.g., radars, optical instruments), offering data products at multiple processing levels (from raw data to advanced products). Furthermore, in addition to satellite data (e.g., public government data), datasets utilized for geospatial applications might include aerial images, in-situ data, and other collateral information. To extract information and knowledge EO actors process this data. The information data is similarly sizable and faces the same Big Data issues mentioned above. For example, 1PB of Sentinel data may consist of around 750000 datasets which, when processed, can yield approximately 450TB of information and knowledge contents (e.g., classifications of items observed).

- **Veracity:** Reliable information is required for decision-making in operations. As a result, verifying data quality is critical for the entire information extraction chain.

- **Value:** The extraction of information from Copernicus data directly improves Europe's economy. Several economic assessments have found that the Copernicus initiative can substantially influence job generation, innovation, and growth. According to the Copernicus Market Report 2016, the overall investment in Copernicus will reach EUR 7.4 billion between 2008 and 2020, with a collective commercial benefit of roughly EUR 13.5 billion created during the same period; it also will provide 28.030 employment years in the EO industry.

For these reasons, EO problems represent a good platform to design, develop

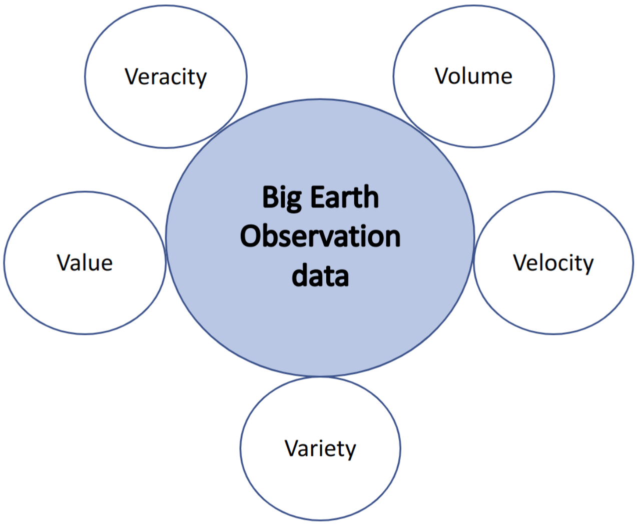


**Figure 1.6:** Five characteristics of a big data problem [28].

and test distributed deep learning architectures. Indeed, it is possible to expect that distributed deep learning frameworks can provide a remarkable added value to the implementation of EO analysis pipeline, so to substantially support the understanding of human-environment interactions by retrieving solid information from EO data analysis.

## 1.3 Earth Observation

The science of remote sensing (RS) is the ability to obtain information without physically touching an object or surface. In this process, the reflected or emitted radiation of the remote object or surface is observed and measured. Based on these measurements, the object and materials are identified and categorized by class or type, essence, and spatial characteristics. The Earth's physical, chemical, and biological systems can be measured and mapped from various remote sensing platforms, including satellites and aircraft, This technology is known as Earth Observation (EO) [29]. Plenty of phenomena, such as climate change, disasters, disease outbreaks, ship navigation, and fire and smoke observation, can be studied using Earth observation data [30]. The amount of EO data is steadily increasing due to the rapid development of this technology and the continuous launch of more satellites, this is one of the main reasons for which the EO domain has become a significant big data application area. [31].

The developments in the EO domain are now driven by various application areas and environmental and climate problems threatening our planet such as:

- **Mapping land cover and land use:** Land-use and land cover (LULC) map help assess climate change effects on hydrology, biodiversity, carbon dynamics, population, migration, and urbanization. This LULC maps can effectively be used for mobilizing decision-makers, industry, farmers, and the general public toward more sustainable use of resources [32, 33].

- **Carbon biomass assessment:** Forest biomass is a key in estimating carbon sequestration. the Forests sequester carbon in part by accumulating biomass since approximately half of the forests' dry biomass is carbon [34]. In carbon biomass assessment, Landsat Imagery, Shuttle Radar Topography Mission (SRT) and International Space Station SERVIR Environmental Research and Visualization System (ISERV) are examples of remote sensing platforms which provide principal images of the Earth [35].

- **Agriculture and food security:** In developing countries with subsistence farming, food security significantly impacts agriculture. Defined by the World Health Organization (WHO) is, food security characterized by three factors; (1) access to sufficient resources for obtaining a nutritious diet, (2) knowledge of primary nutrition, and (3) adequate water and sanitation. Satellite remote sensing is a prominent tool for acquiring food security [36].

- **Disaster management:** Global climate change severely impacts the already marginalized areas of the Earth, which are more susceptible to unpredictable weather patterns, floods, droughts, and rising sea levels. There is a need for chief short, medium, and long-term mitigation and warning strategies at national and regional levels to prepare for disasters like landslides, floods, and earthquakes. Satellite remote sensing is an important tool in establishing this preparedness [37].

- **Polar monitoring:** Despite its remoteness, the Arctic is home to 4 million people and has an economy exceeding 230 billion US$ (World Economic Forum, 2014). EO has a critical role to play in securing sustainable development in the Arctic [38]. Furthermore, this region directly impacts climate change and human life on the whole planet Earth. In the Arctic, satellite data are used to generate information about **sea ice conditions and weather**, which is of paramount importance for Arctic peoples, science, the commercial sector, and decision-making, marine navigation, safety, and climate change research [5].

### 1.3.1 Thesis Application: Sea Ice Analysis

The specific application studied in this thesis is sea ice classification. Sea ice is a critical environmental component of the Earth's climate system [39] and considerably impacts polar ecosystems. The Arctic has witnessed substantial climate change in the last decades, affecting its ecosystem, ecology, and meteorology. The changes are stronger in the Arctic than in any other place. These changes have been dubbed the Arctic augmentation, causing highly changeable Arctic weather and sea ice conditions. These harsh conditions pose challenges and hazards to high north maritime operations connected to resource extraction, fishing, and tourism [40, 41]. As a result, reliable and continuous monitoring of sea ice behavior, thickness, and ice type distribution are critical for effective and reliable human activities, along with for understanding changes over extended time frames [42, 43]. Therefore sea ice monitoring is a prominent EO application from a scientific and societal point of view.

Every year, many polar nations conduct scientific cruises to the Arctic to perform sea ice observations and surveys. Regular ice watch inspections from vessels [44, 45] are the most common and fundamental in-situ measurements. In-situ investigations on the ice include ice thickness and roughness measurements, temperature and salinity profiles, and the thickness and characteristics of snow cover on sea ice [46]. For validation of satellite RS products, airborne ice thickness measurements using electromagnetic EM induction devices are frequently used and referred to as in-situ data [47].

### 1.3.2 Synthetic Aperture Radar (SAR)

There are two primary forms of remote sensing sensors, active and passive. They are distinguished by generating the signals' sources used to examine an object. Active remote sensing systems generate their own illumination, whereas passive systems rely on reflected or backscattered radiation generated by other sources, like the sun. The electromagnetic radiation most commonly used for remote sensing differs in wavelengths, is classified as short (visible, NIR, and MIR) and long (visible, NIR, MIR and microwave). In the high north, microwave sensors are particularly prominent, because they are generally independent of weather and light conditions [48].

Synthetic Aperture Radars (SARs) are active imaging remote sensing sensors, commonly employed in Arctic monitoring. They are particularly useful for monitoring sea ice. It can obtain high spatial resolution by using the coherent character of the transmitted radar pulse by combining radar technology and advanced signal processing [49]. SAR imaging is not affected by sunlight or cloud cover. The difference between SAR and optical images is illustrated in

Figure 1.7.

Sea ice classification using SAR images is affected by several challenges that influence the performance of any machine learning system. These include incidence angle dependencies, seasonal changes, ambiguous radar scattering, and system noise. The SAR challenges will be discussed in more detail in chapter 2.

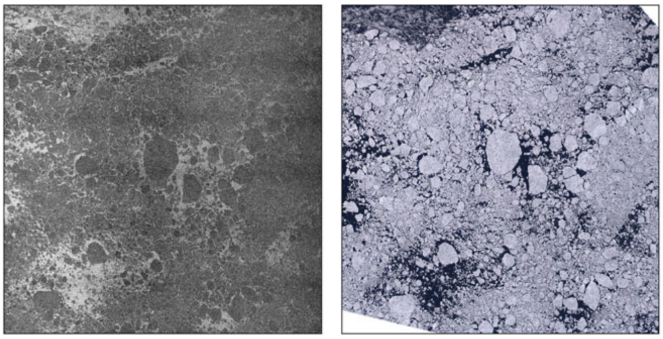


**Figure 1.7:** SAR vs Optical Earth observation. Left is a Sentinel-1 SAR scene and right is a optical image by Sentinel-2 from same location with small time gap [50].

### 1.3.3 Classification Problems

Focusing on the operational needs of stakeholders, communities and authorities in the Arctic ecosystem, RS research on sea ice monitoring has addressed two classification problems:

- Classification of sea ice as opposed to water: High-resolution ice masks can provide detailed information about the location of the ice edge and leads, and can be utilized to build large-scale ice concentration maps.

- Multi-class ice type classification: This entails the more general classification problem of constructing ice maps of multiple classes, including first and multi-year ice, thin ice, deformed ice, ridges, and leads.

The primary emphasis of this thesis is on the binary classification of sea ice versus water. However, the assumptions made in the suggested approaches are such that the algorithms described in this thesis may also be extended to multi-class ice type classification.

### 1.3.4 Exploitation: Operational Ice Charting

National sea ice services are responsible for producing operational sea ice charts, which are often updated daily. The ice charts offer information on the concentration of some ice categories and ice margins based on a range of satellite data, principally from SARs, passive microwave, and optical sensors. Visual observations and meteorological weather forecasts supplement the satellite data. Thermal infrared imaging satellite data are used to create contours of the sea surface temperatures. [51]. Currently, none of the ice services have reported deploying automated categorization methods on a year-round basis. All algorithms still require some level of human intervention. The criteria for ice chart generation vary greatly based on the end-user and their demands. There are both operational and scientific end-users. Scientific end-users employ ice charts for academic research, such as in climate, biology, or data assimilation in numerical models, On the other hand, operational end-users demand timely ice supporting their activities. Mariners, for example, require data to support navigation and safety in ice-covered waterways. In this respect, precise and consistently accurate information regarding the location of the ice edge, leads and ridges, areas of (thin) first-year ice, and areas of multi-year ice is critical. Mariners also demand short-term projections of the development of ice conditions in the region where they are or are headed. [52].

Current ice charts for European seas are provided after 1500 UTC on weekdays (Monday-Friday), and for the Antarctic on Mondays (October-April). [51]. An example of an ice chart for January 14th, 2022 is provided in Figure 1.8.

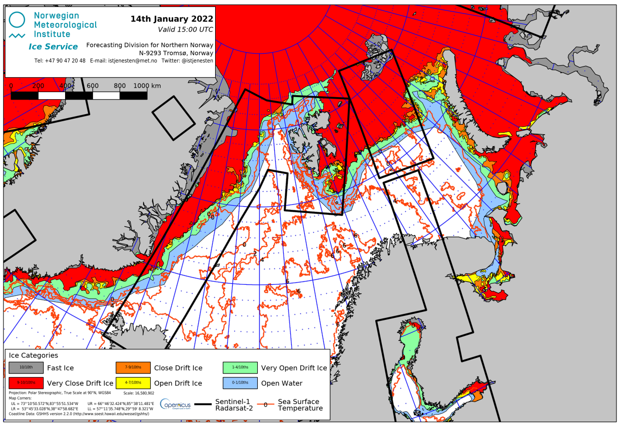

Norwegian
Meteorological
Institute
Ice Service
14th January 2022
Valid 15:00 UTC
Forecasting Division for Northern Norway
N-9293 Tromsø, Norway
Tel: +47 90 47 20 48 E-mail: istjenesten@met.no Twitter: @istjenesten
0 200 400 600 800 1000 km
Ice Categories
10/10th Fast Ice
9-10/10ths Very Close Drift Ice
7-9/10ths Close Drift Ice
4-7/10ths Open Drift Ice
1-4/10ths Very Open Drift Ice
0-1/10ths Open Water
Projection: Polar Stereographic, True Scale at 90°N, WGS84
Scale: 16,580,902
Map Corners:
UL = 73°10'50.572"N,83°55'51.534"W
UR = 66°46'32.424"N,85°38'11.481"E
LR = 53°45'33.028"N,38°47'58.682"E
LL = 57°11'35.748"N,29°59' 8.321"W
Coastline Data: GSHHS version 2.2.0 (http://www.soest.hawaii.edu/wessel/gshhs/)
Sentinel-1
Radarsat-2
Sea Surface Temperature

**Figure 1.8:** Example of an ice chart produced by the Ice Service of the Norwegian Meteorological Institute (NIS). The polygons depict different ice concentration zones. The ice charts also depict the ice edge to indicate open water which is important for ship navigation. Detailed information can be found in [51].

## 1.4 Objectives of this Thesis

Sea ice is a dynamic and complicated target surface, making it difficult to develop a reliable algorithm for autonomous classification based on SAR images. Heavy speckle noise, incidence angle impact, range-dependent noise pattern, and sensitivity of SAR signals to the sea surface all contribute to the difficulty of sea ice analysis [53]. Robust ice charting and ice edge identification become increasingly difficult as a result of the intricate interaction between SAR signals, the imaging geometry, and underlying physical processes [54]. To accomplish automated sea ice analysis, rather than using feature engineering or hand-crafted features (i.e. features that have been designed to highlight specific properties of the surface), this project will use deep neural network architectures as a feature learning scheme to achieve more robust features and better classification.

In this regard, the objectives of this thesis can be summarized as follows:

**Main objective:** The main objective of this thesis is to investigate the capabilities of deep neural networks and propose new methods to improve sea ice analysis by considering the scarce labeled data and computation complexity issues.

- **Objective 1:** Investigating the capabilities of deep neural networks based on self-designed and previously proposed famous architectures considering networks depth, different inputs, and learning strategies (Paper 1). This investigation includes identifying specific issues and requirements related to sea ice analysis, which will subsequently be addressed by semi-supervised learning and scalable computing.

- **Objective 2:** Investigating different semi-supervised methods to address scarce training data considering the specific requirements of sea ice analysis by deep neural networks. We propose a new architecture for semi-supervised sea ice classification in Paper 2.

- **Objective 3:** Investigating distributed deep learning in relation to sea ice analysis. We propose a new distributed deep learning method to address the computational complexity of deep neural networks' training for the sea ice analysis application (Paper 3).

The thesis considers deep neural networks as a holistic learning architecture for feature extraction and classification problems. It proposes new methods in deep supervised (Paper 1, Chapter 6) and semi-supervised (Paper 2, Chapter 7) learning to improve the accuracy and generalization of sea ice analyses. Moreover, to mitigate the computational complexity of training deep learning architectures, the thesis proposes a new distributed deep learning method for sea ice analysis (Paper 3, Chapter 8).

### 1.4.1 Thesis Outline

The three journal articles, which form the core of this thesis, are discussed in detail in chapters 6, 7, and 8. In these three studies, supervised and semi-supervised learning for SAR-based sea ice classification are discussed, and a novel distributed deep learning algorithm is suggested. Chapter 2 provides essential background material, covering the basic principles in Synthetic Aperture Radar (SAR) image formation, emphasizing sea ice categorization, to provide the necessary context. Chapter 3 discusses several learning techniques, including supervised and semi-supervised deep learning. The core idea and techniques of distributed deep learning are covered in detail in Chapter 4. Chapter 5 provides a summary of the scientific papers. Finally, Chapter 9 summarizes the findings and discusses possible future studies.

# 2

# Synthetic Aperture Radar Imagery for Sea Ice Classification

Synthetic Aperture Radar images are the primary data source for operational ice services [55], and the scientific study is presented in this dissertation. An imaging Synthetic Aperture Radar, abbreviated as SAR, is a radar that can be operated from the ground, aircraft (airborne) or satellites (spaceborne) that can provide two-dimensional images of the Earth's surface [56]. This chapter discusses the fundamental concepts of spaceborne SAR, emphasizing on its use for the RS of sea ice in polar regions.

Generally speaking, radar (radio detection and ranging) systems are designed on the principle of echolocation. An antenna sends an Electromagnetic (EM) signal and analyses the bounces returned from a given target. Assuming that the signal's speed is known, the distance between the antenna and the target may be estimated by multiplying the signal's travel time by speed [29].

The Earth is now orbited by several SAR instruments, each operating at a different frequency range, with various polarization channels, and divergent spatial resolution. They provide a critical contribution to observations of distant and inaccessible locations, such as the high northern latitudes. The measurements are independent of sunlight or any other naturally released radiation since the

SAR is an active system that generates its own signals [57]. Furthermore, the usual SAR wavelengths penetrate clouds with little or no loss of energy [58]. As a result, SAR sensors may capture pictures continually, regardless of whether or not there is sunshine or good weather. This matter is particularly paramount in the Arctic, where there is prolonged darkness throughout the polar night and generally approximately 70-80 percent of cloud cover throughout the year [59]. Because of its capacity to capture images throughout the day and in all weather conditions, SAR has become the chief data source for sea ice charting in operational ice services across the globe [60]. On the other hand, SAR images are significantly different from optical images and may be difficult to decipher and comprehend.

The geometry of a side-looking imaging radar system is sketched in Figure 2.1.

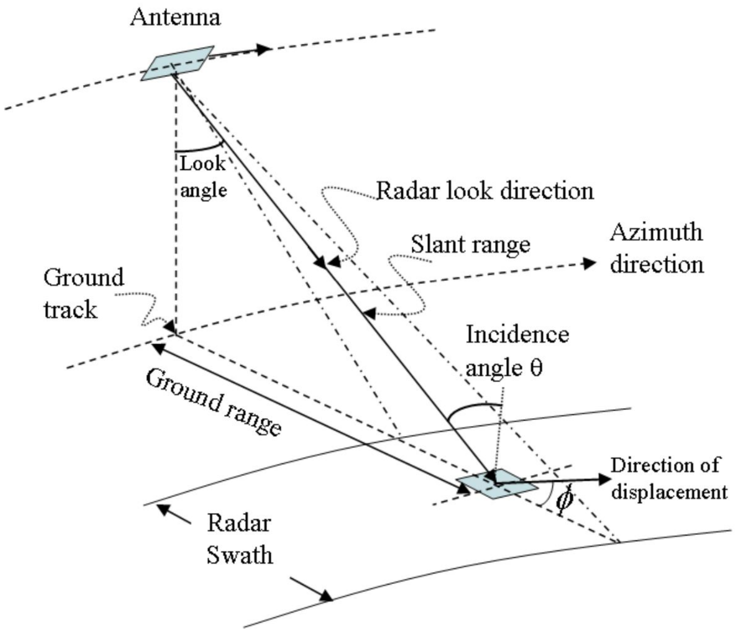


**Figure 2.1:** A geometric model for a SAR system.

The main concepts of SAR imaging are defined below: **Swath**: The swath width is the coverage of the image in range direction, which is the direction perpendicular to the flight direction. The along-track direction is often referred to as the azimuth direction (Figure 2.1). **Slant range**: Slant range is the length between the antenna and grounds of a pixel. **Ground range**: The ground range is the distance between the ground track and the ground pixel (Figure 2.1). **Incidence Angle (IA)**: The incidence angle is the angle formed between the incident SAR beam and the axis normal to the geodetic ground surface of the area where the beam is incident.

## 2.1 Spaceborne SAR System Properties

SAR systems can image the Earth's surface from space with a high spatial resolution. They are furthermore characterized by properties like frequency, polarimetric capabilities, and the temporal resolution given by the revisit time period of the satellite platform. The following are the definition of these characteristics:

**Spatial Resolution:** Spatial resolution is a prominent metric for measuring image quality because it indicates the distance between two objects on the ground which can be separated in a SAR image. SAR spatial resolution is characterized in the azimuth and range directions separately. Range refers to the across-track direction, perpendicular to the satellite's flight direction, while azimuth refers to the along-track dimension, parallel to the satellite's flight direction. In a real aperture radar, the range resolution is dependent on the length of the pulse. Two distinct targets on the surface will be resolved if their separation is greater than half the pulse length. Most SAR systems use a linear frequency modulated pulsed waveform, called a chirp, and achieve, after compression, a range resolution that is inversely proportional to the chirp bandwidth, i.e., $\delta_r = \frac{c_0}{2B_r}$, where $\delta_r$ is the range resolution, $c_0$ is the velocity of light, and $B_r$is the chirp bandwidth.

In a realistic real aperture radar, the azimuth resolution is determined by the angular width of the radiated microwave beam, and the slant range distance [61]. This beam width is a measure of the width of the illumination pattern. As the radar's distance to the surface increases, the azimuth resolution gets worse. A SAR exploits the relative movement of the antenna compared to the imaged surface, which in essence generates a linear frequency modulation in the azimuth direction. This feature is exploited to improve the resolution dramatically.

In fact, after compression, the azimuth resolution becomes $\delta_a = \frac{L_a}{2}$, where $\delta_a$ is the azimuth resolution, and $L_a$ is the antenna length in the azimuth direction. The motion of the platform is used to synthesize a long antenna in order to get a high azimuth resolution [61]. Hence, the name synthetic aperture radar, and we note that both the range and the azimuth resolutions are independent of the range to the target.

**Frequency**: Most spaceborne SAR systems operate at wavelengths between 0.5

and 75 cm [62]. Using this spectral range has the primary advantage of allowing the atmosphere to be virtually transparent. The atmosphere is transparent for wavelengths in microwave bands. The backscattered radiation is generally dependent on the surface roughness at the scale of the radar wavelength. The wavelength selection should therefore be made to match the size of surface features on the targeted object. Furthermore, longer wavelengths can penetrate through the snow cover and the ice, resulting in a higher contribution of volume scattering. The most commonly used frequency bands for spaceborne SAR observations of sea ice are L-band, C-band, and X-band [63].

**Polarimetry**: An EM wave is made up of an electric and magnetic field. These fields are orthogonal to each other and the propagation direction of the wave. The direction of the electric field characterizes the signal's polarization, and it may be expressed in terms of two orthogonal basis vectors [64]. EM waves are polarized, with linear or circular polarization in particular cases [57]. Most SAR satellites use linear polarization on both the transmission and reception. The polarization direction in the plane of wave propagation is either horizontal (H) or vertical (V). An individual SAR channel might also be defined by a two-letter combination, with the first letter indicating the polarization of the sent signal and the second letter indicating the polarization of the received signal. The HV channel, for example, sends in horizontal polarization and receives in vertical polarization.

**Temporal Resolution**: When building a time series of satellite images, the revisit cycle determines the temporal resolution. The revisit cycle equals the time it takes between two observations of the same object on the Planet's surface. These observations may occur at different orbits of the SAR. The revisit cycle should not be confused with the repetition cycle, which is defined as the duration between two passes of a satellite along the same orbit, according to [65]. While the repetition cycle is solely determined by the satellite orbit configuration, the revisit duration is determined by the target location and swath width. The majority of SAR sensors in space are borne by polar orbiting spacecraft. The orbit track spacing for these satellites is closer at higher latitudes, resulting in a much higher revisit rate in Polar regions.

**Data Acquisition Modes**: The majority of spaceborne SAR sensors can operate in a variety of acquisition modes. The geographic coverage, resolution, and polarimetric capability all change across these modes. For example, the SAR instrument onboard Sentinel-1 may operate in: Stripmap (SM), Interferometric Wide Swath (IW), Extra Wide Swath (EW) and Wave (WV). The best acquisition mode and image product for a certain application must be selected based on its individual aims and needs, along with the overall availability of data. The Stripmap (SM) and ScanSAR modes are the most used acquisition modes. The antenna footprint is fixed to one swath in SM mode, and a continuous strip on the Earth's surface is scanned. By directing the antenna to various elevation angles and merging numerous subswaths, the Scan SAR mode obtains more sizable spatial coverage. This increased geographical coverage comes at the expense of reduced spatial resolution.

## 2.2 Sentinel-1

There are several spaceborne platforms operated by different countries for sea ice monitoring and analysis. They include satellites like, Radarsat-2 (Canada), TerraSAR-X/TanDEM-X (Germany), HJ-1C (China), ALOS-2 (Japan) and Sentinel-1A/1B (Europe/ESA). These sensors take images at different frequencies and polarization constellations.

The Sentinel-1 C-band SAR instrument supports operation in single polarization (HH or VV) and dual polarization (HH+HV or VV+VH) modes, implemented through one transmit chain (switchable to H or V) and two parallel receive chains for H and V polarisation. SM, IW, and EW products are available in single (HH or VV) or dual polarization (HH+HV or VV+VH). WV is single polarisation only (HH or VV). The primary conflict-free modes are IW, with VV+VH polarization over land, and WV, with VV polarization, over the open ocean. Having the Interferometric Wide swath mode as the chief operational mode satisfies most current service requirements, avoids conflicts, preserves revisit performance, simplifies mission planning, decreases operational costs, and builds a consistent long-term archive [56]. Figure 2.2 shows Sentinel-1 acquisition modes.

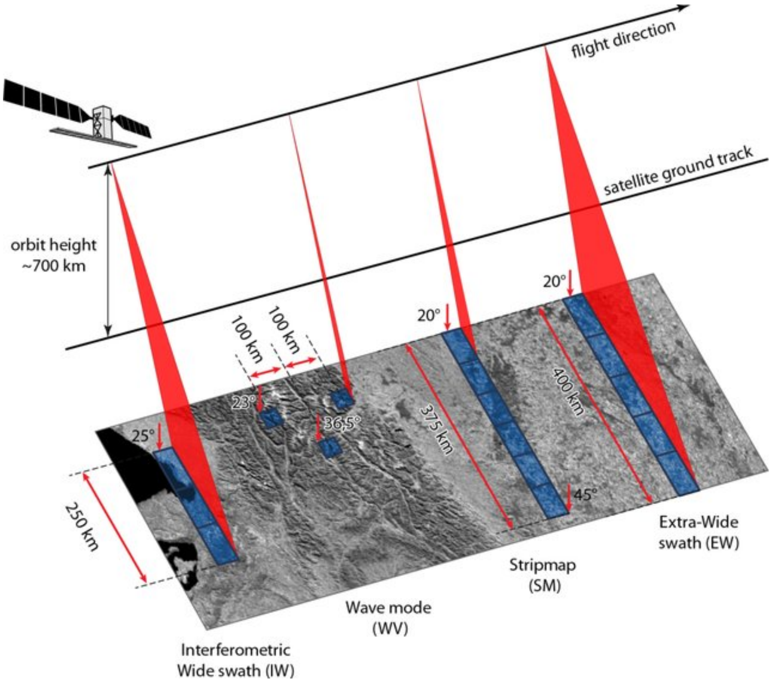


**Figure 2.2:** Illustration of Sentinel-1 acquisition modes [66].

Generally, The Sentinel EW mode is utilized for wide-area coastal monitoring, such as ship traffic, oil spill, and sea-ice monitoring, among others [65, 67]. Strictly speaking, SM mode is only utilized on tiny islands and upon request for special occasions like disaster relief. At least some operational sea ice charting criteria can be met by the EM mode's extended coverage at a resolution for various applications [56]. Sentinel-1 EM mode data have been used as the primary source in Paper 1 and Paper 2 of this dissertation to develop deep learning models for sea ice classification.

Sentinel-1 has a temporal revisit period of twelve days and completes 175 orbits every cycle, giving it a spatial resolution of twelve days. Because Sentinel-1a and Sentinel-1b circle in the same orbit plane with a 180° orbital phase difference, the effective repetition constellation cycle is decreased to six days instead of the previous twelve.

## 2.3 Sea Ice Classes in SAR Images

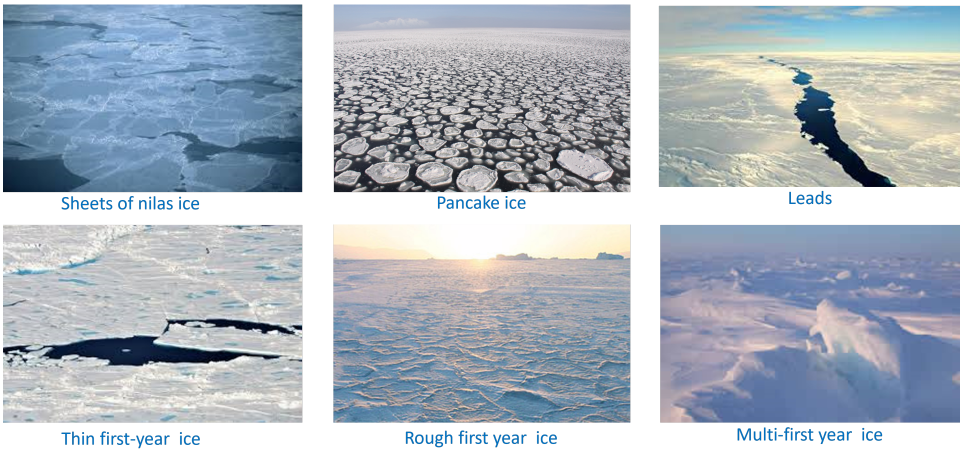
Sea ice surfaces may be classified into different ice types. Distinct user communities use different ice class definitions. The most used categorization criteria

**Table 2.1:** Different water and ice classes.

| WMO code | Classes |
|---|---|
| 02 | Open Water/ Leads with Water |
| 01–02 | Brash/Pancake Ice |
| 83 | Young Ice (YI) |
| 86–89 | Level first-year ice FYI |
| 95 | Old/deformed Ice |

are based on the ice thickness, where ice type is associated with a defined thickness range. In general, ice thickness is related to the ice age. Therefore, these criteria may be regarded as age-based [68]. A term used to describe the distinct thickness/age-based classes of sea ice is the sea ice stage of development (SOD). [69] describes a categorization method that has been accepted by the World Meteorological Organization (WMO) and is frequently used in operational ice charts. Figure 2.3 shows various types of ice.

In this dissertation, we use a dataset with five classes based on the WMO definition. The Table 2.1 shows these classes and their respective codes.



Sheets of nilas ice

Pancake ice

Leads

Thin first-year ice

Rough first year ice

Multi-first year ice

**Figure 2.3:** Ice types in optical observation.

Ice types appear very differently in SAR compared to optical images. Their signatures in SAR depend on the backscatter of the radar signal at the ice and water surfaces and the signal's penetration into the ice. Figure 2.4 displays an optical image from the same region as a SAR image, acquired with a short time interval between the two images. To take both HV and HH polarization channels into account, they are displayed as the red and green channels, respectively, while the blue channel is zero.

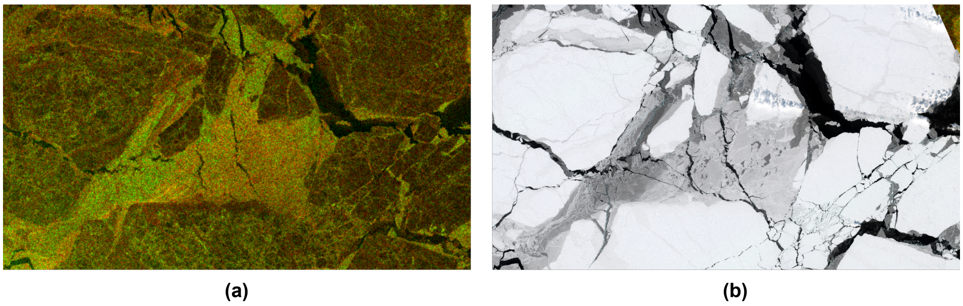


**Figure 2.4:** Sentinel-1 HH+HV SAR image, (a), versus Sea Ice in optical image, (b).

## 2.4 Challenges

Sea ice classification has a variety of challenges that might impact the performance of any machine learning classification system. Several particular issues associated with this dissertation are addressed in this section.

**Incidence angle dependency**: In the case of spaceborne SAR systems, the IA can span a wide range from the side-looking aspect of imaging geometry, where the local IA changes dramatically from the near to the far range views. While this significantly impacts the resulting image intensity values, it also represents an important physical phenomenon related to radar scattering, which is influenced by the physical properties of the sea ice surface and volume. The IA gradient is dependent on the ice type. It is also very different for a sea ice surface and open water

**Seasonal changes**: Seasonality significantly influences the dielectric characteristics of ice and snow covers, and also, as a result, the intensities of the backscattered signals. During the winter, the surface scattering from young ice and the volume scattering from the bubble structure in the top layer of multi-year ice strongly contribute to backscattered signals [70]. Melting alters the physical qualities and dielectric properties of sea ice and snow, resulting in a drastic reduction in the radar signal from a particular ice type consequence of increased humidity. Hence, it becomes increasingly difficult to distinguish between different ice forms in the summer, as seen in Figure 2.5. This figure shows two Sentinel-1 SAR images from the fast ice region off the east coast of Greenland before (on the left), and after (on the right) melting starts.

**Ambiguous scattering**: The roughness and dielectric characteristics of the target surfaces are responsible for the backscattering seen from sea ice and water surfaces. In the ocean, small-scale roughness (on the order of the radar

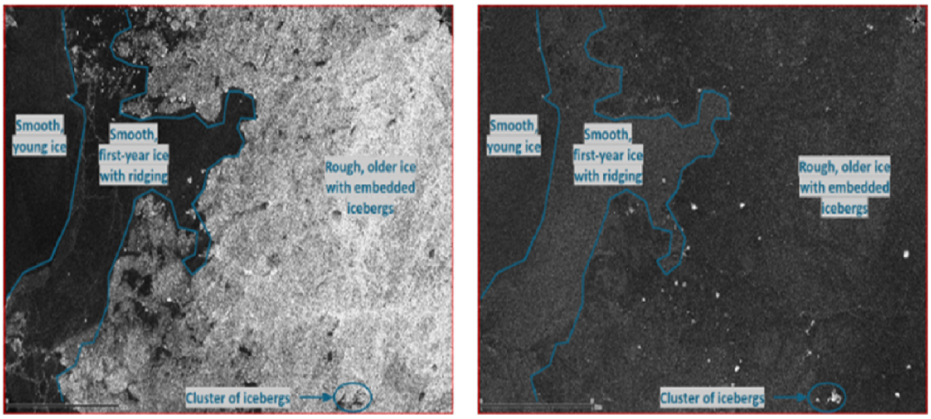

Smooth, young ice
Smooth, first-year ice with ridging
Rough, older ice with embedded icebergs
Cluster of icebergs
Smooth, young ice
Smooth, first-year ice with ridging
Rough, older ice with embedded icebergs
Cluster of icebergs

**Figure 2.5:** The difference between Sentinel-1 IW-mode photos (HH polarisation) for a) pre-melt sea ice (17 April 2018) and b) full-melt sea ice (17 April 2018) is shown in the figure (16 June 2018). Due to the fact that this is a region of fast ice, Belgica Bank off the coast of north-east Greenland, the sea ice and icebergs are identical in both photographs (courtesy of Nick Hughes, the Norwegian Meteorological Institute).

signal's wavelength size) is the dominant source of scattering, and they are modified by the larger waves that go through the water. As a result, excessive wind and turbulence will produce high backscatter, with texture linked with random directions of the wind turbulence. The scattering process is more complicated for sea ice. Level or deformed ice surfaces, dry or wet, snow-covered or bare, are all possible conditions on an ice surface. There are also incidence angle dependencies, which depend on ice type, to consider. All these phenomena combine to make the SAR returns confusing, to the point that many diverse ice types may provide almost identical signal characteristics. Even distinguishing between open ocean and sea ice may be problematic. This problem is depicted in Figure 2.6.

**Noise characteristics of Sentinel-1:** Internal noise on Sentinel-1 has an impact on the performance of SAR systems. A structured but class-independent intensity noise signal is introduced throughout the images as a result of this, which is particularly noticeable in the lower intensity HV and VH channels. As a function of range (and hence IA), this noise floor is often more prominent with the five sub-swath assemblies of Terrain Observation with Progressive Scans SAR Sentinel-1 EW imagery. The varying noise floor often has undesired classification effects comparable, but distinct, from the IA dependence. It has proven difficult to efficiently filter out this noise component, in part because the spatial correlation properties of the noise are very similar to the spatial correlation properties of speckle, which is a well-known phenomenon associ-

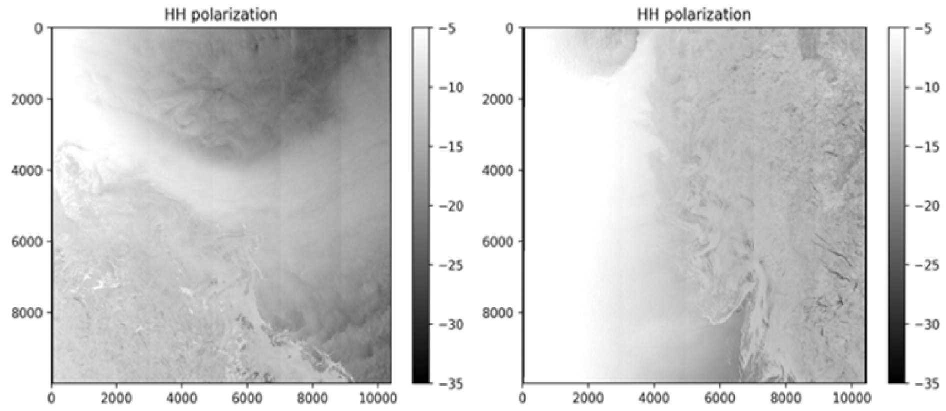


**Figure 2.6:** Two HH-polarization sentinel-1 SAR from the marginal ice zone.

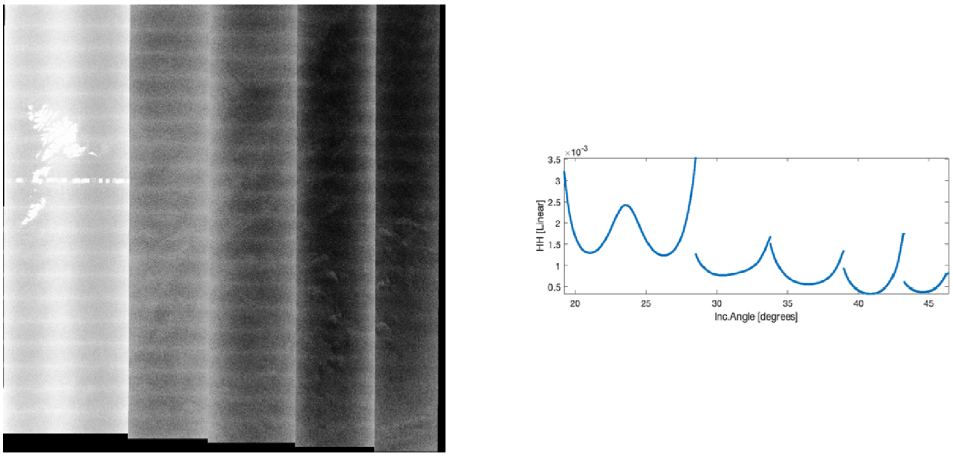


**Figure 2.7:** An example of a noise HV SAR image, with corresponding noise floor profile on the right.

ated with coherent radar imaging and has been studied extensively. Figure 2.7 shows an example of a noise pattern in a Sentinel-1 EW-mode HV SAR scene (on the left), and the associated noise floor profile (on the right).

**Ground truth labeling**: Acquiring in-situ ground truth for Sea Ice classification is very costly, time-consuming, and in certain cases, not even practicable. There are many reasons for this, including the difficulties of acquiring ground truth in this particular region of the Earth, which has extended dark seasons and very variable weather conditions. Even though numerous sensors are often utilized to label the data, manual labeling suffers from the above issues. Manual labeling is restricted in diversity since humans can only assess a small number of locations, resulting in tiny and unbalanced training datasets. It is important to note that this is due to the nature of the application and that even with great efforts, we will only have a limited quantity of labeled data compared with the volume and diversity of data available in the Arctic. Thus, it is a critical problem that should be considered, particularly in deep learning training. The issue of scarce training data is well-known in the machine learning and deep learning domains. It has resulted in significant hurdles in the field of big data applications. For this reason, new advanced approaches in deep learning, such as deep semi-supervised learning, deep unsupervised learning, and deep self-supervised learning, have been developed to solve this problem at hand.

**Lack of automatic learning methods**: The majority of Sea ice classification techniques rely on traditional and statistical approaches, while deep learning models and architectures for sea ice classification are still not very mature, especially for operational use. When a deep learning model is trained on sea ice records, it may uncover a slew of previously undisclosed difficulties and confederations. This issue is much more crucial when the question of the explainability of deep learning is still being debated. Thus, describing deep learning models' behaviors based on newly found physical features might be challenging.

# /3

# Deep Neural Networks Learning Approaches

Deep Learning (DL) is a machine learning technique that creates artificial neural networks designed to imitate the structure and function of the human brain. Deep Learning is a technique used to train computers to recognize patterns in data. Deep learning, also known as deep structured learning or hierarchical learning, is a type of nonlinear processing that, in practice, employs a large number of hidden layers - typically more than 6, but often much higher - to extract features from data and transform the data into different levels of abstraction or representations [71].

This concept is motivated by the fact that the mammalian brain is structured in a deep architecture, with each input percept encoded at numerous layers of abstraction, particularly in the monkey visual system reference [72]. DL researchers have built unique deep architectures as an alternative to shallow architectures, inspired by the architectural depth of the human brain. Traditionally, machine learning needs human feature extraction from training data by a subject matter expert. However, as shown in Figure 3.2, a deep learning framework automatically learns relevant features from training data such as images, text, patterns, and digital signals.

The optimization techniques for training deep models differ numerously from typical optimization strategies. Typically, machine learning operates directly. In

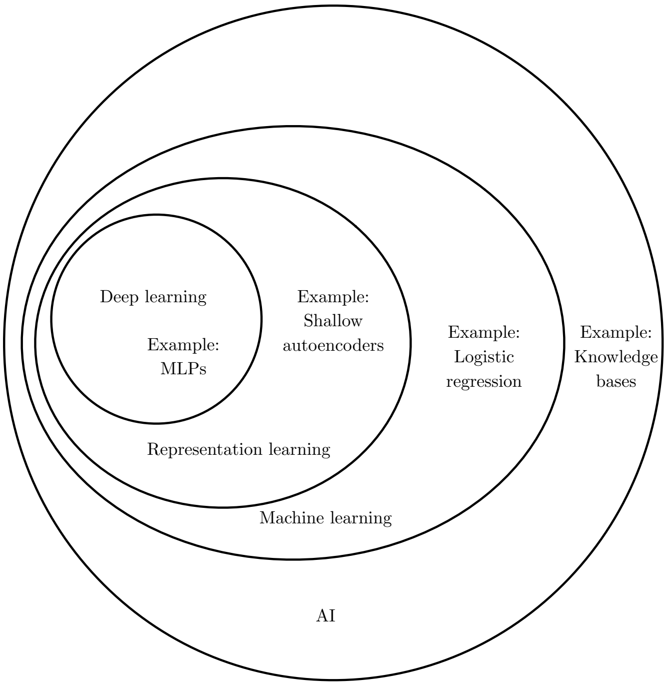


**Figure 3.1:** Deep Learning is a machine learning technique [9].

the majority of machine learning cases, we are interested in some performance measure P that is defined relative to the test set and may be intractable. As a result, we optimize P indirectly. We lower a different cost function $J(\theta)$ to improve P. This contrasts with pure optimization, in which reducing J is a self-contained objective. Additionally, optimization techniques for deep model training often incorporate some specialization in the form of machine learning objective functions. Typically, the cost function is expressed as a weighted average across the training set, as follows:

$$J(\theta) = \mathbb{E}_{(x,y)\sim\hat{P}_{data}} L(f(x,\theta), y) \tag{3.1}$$

where $L$ is the per-example loss function, $f(x,\theta)$ is the predicted output when the input is $x$, and $\hat{P}_{data}$ is the empirical distribution. In the supervised learning case,y is the target output.

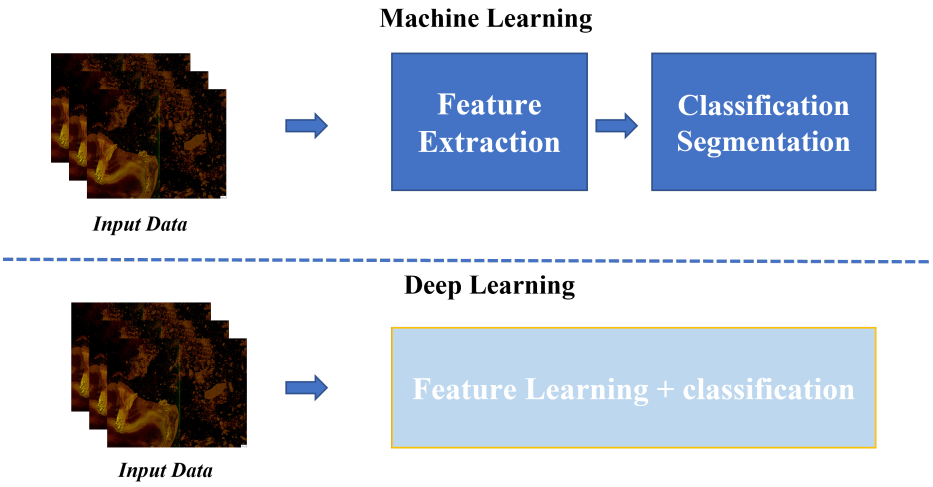


**Figure 3.2:** Deep Learning vs Traditional machine learning.

## 3.1 Learning Approaches

The learning techniques used in deep learning methodologies are generally classified as follows: 1) Supervised Learning, 2) Semi-supervised Learning (abbreviated as SSL), and 3) Unsupervised Learning. Supervised learning creates a knowledge base from previously identified patterns, which aids in the classification of new patterns. The primary objective of this learning is to translate the input characteristics into a class output. The result of this learning is the construction of a model from input patterns. The model may be used to categorize previously unknown occurrences accurately. In general, it may be expressed as a function f(x) with patterns as input and a class $y$ as output. The effectiveness of deep learning and CNNs is often contingent on the availability of a vast amount of labeled data, where millions of photos are tagged to train deep neural networks [14, 20] to allow these models to perform on par with or even better than humans.

While visual data are abundant, data that has been accurately annotated by humans is very sparse. Not only obtaining vast volumes of annotated training data for each job is difficult and perhaps expensive, but also it proves to be mistake-prone. This problem is even more essential in the case of remote sensing and Earth observation, since acquiring in-situ ground truth is prohibitively costly, time demanding, and in some cases impossible. This problem will be far more severe when it comes to Arctic data analysis [73]. While the amount of labeled data would be incredibly tiny, the amount of unlabeled data may be enormous. The distribution of such unlabeled data contains critical information constructing resilient representations that are generalizable to novel learning tasks. depending on the number of labeled instances used to train models, Unlabeled data may be used in an unsupervised or semi-supervised way. Unlabeled data may also help models bridge the domain divide between multiple tasks, resulting in a plethora of unsupervised and semi-supervised techniques [16]. These techniques are shown in Figure 3.3.

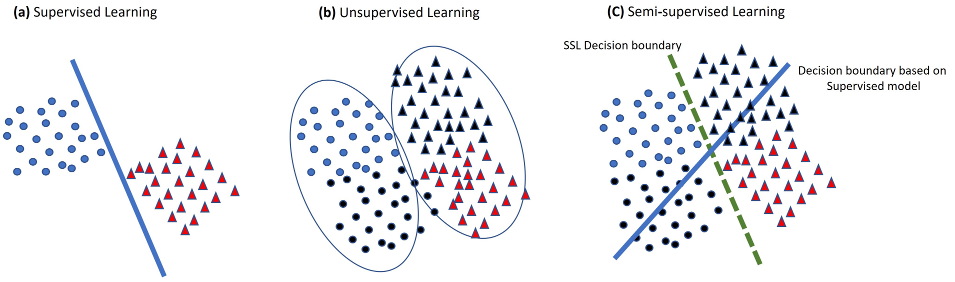

(a) Supervised Learning
(b) Unsupervised Learning
(C) Semi-supervised Learning
SSL Decision boundary
Decision boundary based on Supervised model

**Figure 3.3:** Basic example of binary classification considering supervised, unsupervised, and semi-supervised learning. We have two classes, blue circles, and red triangles. Unlabeled data belonging to each class are shown in black circles and triangles. (a) refers to supervised learning in which all training samples are labeled. (b) is unsupervised learning in the absence of labeled training data. (c) is semi-supervised learning, in which some training samples contain labels and some do not [74].

On the other hand, unsupervised learning does not need a predefined output value. Rather than that, one attempts to deduce some underlying structure from the inputs. Unsupervised approaches aim to develop representations that are sufficiently generalizable to be used for various future learning challenges. For example, in unsupervised clustering, the objective is to infer a mapping from provided inputs (e.g., vectors of real numbers) to groups in such a way that comparable inputs are mapped to the same group [75].

Semi-supervised learning seeks to bridge the divide between these two objectives [76, 77]. While attempting to solve a classification issue, extra data points with unknown labels may be introduced to help the classification process. On the other hand, for clustering algorithms, the learning process may profit from the fact that some data points belong to the same class. Semi-supervised classification techniques are especially useful when labeled data is in little supply [75]. The majority of semi-supervised learning research is focused on classification.

## 3.2 Convolutional Neural Networks

As the most representative supervised DL model, the Convolutional Neural Networks (Convolutional Neural Network (CNN)s) [78] have outperformed most algorithms in visual recognition. The deep structure of CNNs allows the model to learn highly abstract feature detectors and to map the input features into representations that can boost the performance of the subsequent classifiers.

The CNN is a trainable multilayer architecture made from many feature-extraction stages that may be trained. Usually, each level is made from three layers, as follows: a convolutional layer, a nonlinearity layer, and a pooling layer. When designing the architecture of a CNN, it is crucial to consider how the two-dimensional structure of the input picture will be used. A CNN is composed of the feature extraction process including one or more conventional layers, followed by FC layers that serve as the classifier. Figure 3.4 is a diagram that depicts the overall architecture of a CNN .

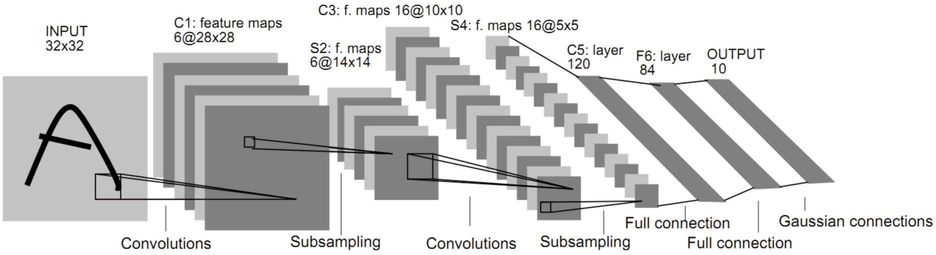


**Figure 3.4:** A general CNN architecture for image analysis, here for character recognition [78].

The convolutional layer receives as input a three-dimensional array of $r$ two-dimensional feature maps of size $m \times n$. Additionally, a three-dimensional array $m \times n \times k$ is produced, consisting of $k$ feature maps of size $m \times n$. The convolutional layer has $k$ trainable filters of size $l \times l \times q$, collectively referred to as the filter bank $W$, which links the input and output feature maps. The convolutional layer generates the output feature maps seen in Equation 3.2.

$$z^s = \sum_{i=1}^{q} W_i^s * x^i + b_s \tag{3.2}$$

where $x^i$ represents each input feature map, * represents a two-dimensional discrete convolution operator, and b represents a trainable bias parameter.

This layer is just a pointwise nonlinearity function applied to each component in a feature map in a conventional CNN. The nonlinearity layer determines the output feature map $a^s = g(z^s)$, where g(.) is often selected to be a rectified linear unit $(ReLU)g(x) = max(0, x)$. This function is often referred to as the activation function. However, for the last layer of completely linked layers, the SoftMax function is utilized to get the probability distribution.

$$Softmax(z^s) = \frac{exp(z^s)}{\sum_j exp(z^j)} \tag{3.3}$$

The pooling layer is composed of a grid of pooling units spaced s pixels apart, each of which summarizes a small spatial area of size $p * p$ centered on the pooling unit's position.

Following numerous feature extraction steps and using a Fully Connected (FC) layers as a classifier, the complete network is trained using backpropagation of a supervised loss function such as the traditional least-squares output as Equation 3.3.

$$J(\theta) = \sum(\|f(x, \theta) - y\|^2) \tag{3.4}$$

where f denotes the network output after the operation of the SoftMax function to the trainable parameter $\theta$. The output vector y is a 1-of-K vector. The objective is to minimize J($\theta$) as a function of $\theta$. The stochastic gradient descent with backpropagation [79] algorithm is investigated for this optimization.

Gradient-based optimization methods are the most often utilized in DL to solve the above optimization issue. Due to the computational difficulty of second-order gradient descent techniques, first-order gradient descent methods, particularly Stochastic Gradient Descent (SGD) with mini-batch and variations, are often employed in DL.

Generally, the iterative process consists of the following steps: 1) It takes a mini-batch of data and samples it. 2) It conducts feed-forward calculations to determine the objective function's loss value (Equation 3.4). 3) It uses backward propagation to determine the gradients relative to the model parameters. 4) Finally, it does a model parameter update. It takes time to train a deep model, much more so for big models or datasets. It is becoming more popular to use distributed training approaches to expedite the training process by using several processors [25].

## 3.3 Semi-Supervised Learning

Several semi-supervised classification techniques have been proposed throughout the last two decades. These approaches vary in their underlying semi-supervised learning assumptions, their treatment of unlabeled data, and their relationship to supervised algorithms. The primary differences between SSL techniques are due to the various assumptions they make. The underlying marginal data distribution p(x) over the input space must include information about the posterior distribution p(y|x). This is a crucial condition for semi-supervised learning. If this is the case, unlabeled data may be used to elicit information about p(x), and hence about p(y|x). If, on the other hand, this requirement is not satisfied and p(x) includes no information about p(y|x), it is intrinsically impossible to enhance the accuracy of predictions using the extra unlabelled data [77].

Fortunately, the previously indicated criteria seem to be fulfilled in many real-world learning situations, as seen by the effectual application of semi-supervised learning approaches in practice. However, the interaction between p(x) and p(y|x) is not necessarily the same. As a result, semi-supervised learning assumptions such as smoothness, low-density assumption, and manifold have emerged, which codify the forms of predicted interaction [76].

It is assumed that the associated labels $y$ and $y$ should be the same for two input points $x, x^{'} \in X$ that, according to the smoothness assumption, are near to each other in the input/feature space. This assumption is also often employed in supervised learning, but it has an additional advantage in the semi-supervised context: the smoothness assumption may be applied transitively to unlabeled data, which is practical in many situations. Consider the following scenario: there is a labelled data point $x_1 \in X_L$, and two unlabeled data points $x_2, x_3 \in X_U$, and $x_1$ is near to $x_2$ and $x_2$ is close to $x_3$, but $x_1$ is not close to $x_3$. If the smoothness condition is met, we may still expect $x_3$ to have the same label as $x_1$, since closeness (and hence the label) is transitively transmitted via $x_2$.

The low-density assumption states that the decision border of a classifier should, wherever possible, travel through low-density areas in the input or feature space of the classification problem. In other words, the decision border should not travel through densely populated neighborhoods. The assumption is specified over the actual distribution of the input data, which is represented by the function $p(x)$. When just a small number of samples from this distribution are considered, it simply indicates that the decision boundary should be located in an area where only a small number of data points are seen. Consequently, the low-density assumption is closely connected to the smoothness assumption; in fact, it may be thought of as the smoothness assumption's equivalent for the data distribution underpinning the assumption of smoothness.

When dealing with machine learning situations in which the data may be represented in Euclidean space, the observed data points in the high-dimensional input space $R^d$ are often clustered in lower-dimensional substructures of the input space $R^d$. Manifolds are topological spaces that are locally Euclidean. For example, if we examine a 3-dimensional input space in which all points are located on the surface of a sphere, then the data may be said to be located on a 2-dimensional manifold in which the points are located. Semi-supervised learning is based on the manifold assumption, which asserts that (a) the input/feature space consists of numerous lower-dimensional manifolds on which all data points sit and (b) data points lying on the same manifold are assigned the same label. If we can figure out which manifolds exist and which data points are located on which manifold, we may infer class assignments for unlabeled data points from the class assignments of labeled data points located on the same manifold. This is very useful when dealing with large datasets.

Various deep semi-supervised learning methods have been proposed considering these assumptions and advanced deep learning approaches. These approaches are shown in Figure 3.5.

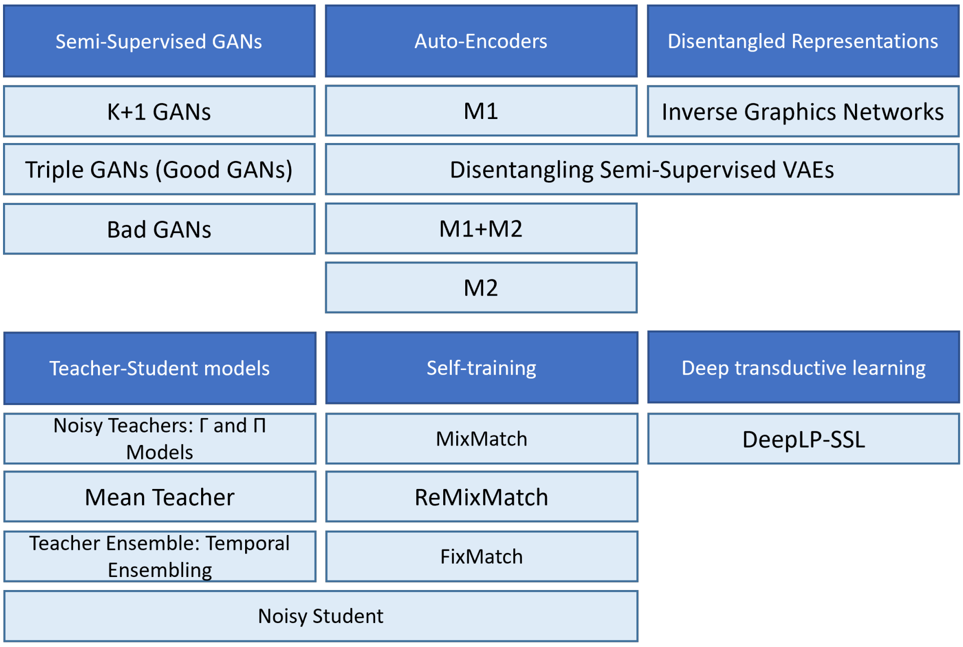

Semi-Supervised GANs
Auto-Encoders
Disentangled Representations
K+1 GANs
M1
Inverse Graphics Networks
Triple GANs (Good GANs)
Disentangling Semi-Supervised VAEs
Bad GANs
M1+M2
M2
Teacher-Student models
Self-training
Deep transductive learning
Noisy Teachers: Γ and Π Models
MixMatch
DeepLP-SSL
Mean Teacher
ReMixMatch
Teacher Ensemble: Temporal Ensembling
FixMatch
Noisy Student

**Figure 3.5:** An overview of the landscape of semi-supervised methods [16].

### 3.3.1 Semi-supervised GANs

Generative Adversarial Networks GANS have been used multifariously to address different problems. For semi-supervised learning, they have represented two distinct approaches to the use of GANS in semi-supervised learning. One suggests training a K+1 classifier using K predefined labels and a false class to represent generated samples. It investigates the distribution of unlabeled instances by classifying them into the first K real classes [80]. The second paradigm interprets the generator of a learned GANs model as the (local) parameterization of the data manifold. This parameterization allows for the description of label invariance along the manifold's tangents. This is strongly connected to the Laplace-Beltrami operator [81], which is only approximated in conventional graph-based semi-supervised models by the graph Laplacian [16].

### 3.3.2 Auto-Encoders

These approaches expand the unsupervised Variational Autoencoders (VAE) to two types of semi-supervised models, which are described in more detail below. The first latent-feature discriminative model (M1), [82], is a basic model to understand. A classifier is trained to predict the label of sample x on top of the latent representation z of a sample x created by a VAE model. While the VAE is trained on both the labeled and unlabeled portions of a training set, the classifier is only trained on the labeled portions of a training set. The second generative semi-supervised model (M2), [82], is more difficult to understand and implement. An additional class variable y, which is latent for unlabeled x but visible for labeled x, is used to construct sample x. This class variable y is generated in addition to the latent representation z. In addition to the M1 and M2 models and the hybrid model, attempts have been made in the literature to include supervision information in VAE ([83]).

### 3.3.3 Disentangling Representations

Through the development of an inverse graphical representation of the visual model, these approaches construct the semi-supervised model based on the VAE model. In the first method, it is intended to learn a collection of "graphics codes," which are used to manipulate and render pictures similar to that of graphics software [84]. Images are represented as disentangled representations of graphics codes in this context. The second technique [85], proposes an extended form of semi-supervised VAEs to separate interpretable variables from latent representations, similar to the previous approach.

### 3.3.4 Teacher-Student Models

The idea behind teacher-student models for semi-supervised learning is to obtain a single or an ensemble of teachers, and use the predictions on unlabeled examples as targets to supervise the training of a student model. Consistency between the teacher and the student is maximized to improve the student's performance and stability in classifying unlabeled samples. Various ways of training the teacher and maximizing the consistency between the teacher and the student lead to a variety of semi-supervised models of this category. Several famous approaches can be categorized in this group, such as mean teacher [86] and Noisy teacher [87, 88].

Obtaining a single or an ensemble of instructors and using their predictions on unlabeled instances as objectives to oversee the training of a student model is the concept underlying teacher-student models for semi-supervised learning. Maximizing consistency between the teacher and the student is crucial to increase the student's performance and stability while categorizing unlabeled samples. Diverse approaches to training the instructor and optimizing consistency between the teacher and the student result in an array of semi-supervised models. In this category, there are numerous well-known techniques, such as the mean teacher [86] and the Noisy teacher [87, 88].

### 3.3.5 Self-training

These methods are based on Consistency regularization by considering noisy data (usually data augmentation) by leveraging the idea that a classifier should output the same class distribution for an unlabeled example even after it has been changed by adding some noise (augmentation). These approaches can be considered as a single teacher-student model in which the model is trained by itself. Therefore, these methods can also be considered the Teacher-Student models. Several famous approaches can be categorized in this group, such as MixMatch [89], FixMatch [90] and [91].

### 3.3.6 Deep Transductive Learning

In transductive learning, label inference that is limited to a certain set of unlabeled examples is crucial. This act may be accomplished, for example, by combining classification loss on labeled data with unsupervised aims on all data, the latter of which serves as a regularizer. In the deep learning field, [73] first employs efficient transductive label propagation [92] to infer pseudo-labels for unlabeled data that are then utilized to train the classifier. Label propagation is a graph-based technique, and in this study, the graph is generated using the classification network's embeddings. As a result, this procedure consists of two phases. The network is first trained using labeled and pseudo-labeled data. The second phase constructs the closest neighbor graph using the embeddings of the network trained in the previous step.

## 3.4 Thesis Approach: DL and Sea Ice Classification

In order to employ DL for sea ice classification, two main challenges, including network architecture and scarce training data, are considered.

**Network Architectures:** Network architecture is an important factor in extracting meaningful features for any application. Hyperparameters such as the number of layers, the number of kernels, the size of kernels and etc. are significantly important to designing a network architecture. However, it is interesting to employ the proposed architectures for other applications and fine-tune them for sea ice applications. Customizing a proposed network for sea ice application could be a shortcut approach. To encounter this issue, we consider the following questions:

- What is the proper network architecture in terms of number layers, number kernels and etc. to be used for sea ice classification?

- Is it beneficial to use existing well-known architectures for sea ice classification?

- Which type of proposed architectures are more suitable for this application?

These questions were significantly important since DL networks are not very mature, and there are few works in technical literature. Therefore, these issues were targeted in the first publication. we proposed the following contribution to address these questions:

- A dataset for deep learning analysis is published.

- Different approaches in deep learning, including ad hoc architecture design, transfer learning, and re-training from scratch, have been studied.

- A self-design ad hoc architecture is designed and trained for sea ice classification.

- Different proposed architectures including MobileNetV2 [93], RestNet50 [20], and DenseNet121 [94] are trained for sea ice classification.

- Considering transfer learning and re-training from scratch, especially the VGG-16 model [95] is trained for sea ice classification.

- Effects of max-pooling layers in the VGG-16 architecture are studied, and a modified VGG-16 model is proposed.

- Scarce training data issue is highlighted through the reported experiments.

- Data augmentation technique is used, and it is highlighted that this technique is led to some uncertainty on preserving the physical properties of signals.

- Based on our finding using self-design ad hoc architecture and proposed architectures, a specific 13-layer is considered for further analysis.

**Scarce training data:** To encounter this issue, we consider the following questions:

- What is the efficient approach to increasing the number of samples in the training data set?

- How advanced approaches can be used to address this issue for sea ice classification?

- Which approach is more appropriate to preserve physical properties?

Scarce training data is an important obstacle toward employing DL models for sea ice classification. To address this issue, we might consider human labeling or an algorithm that can use the advantage of labeled and unlabeled data:

- **Increasing the dataset:** Adding more training data depends on visual manual annotation, which is very time-consuming. Moreover, considering the ambiguity of  image, even with the help of optical images, it is not always visually possible to label data.

- **Semi-supervised learning:** Semi-supervised learning seeks to bridge between labeled and unlabeled data to improve classification accuracy. In this method, the semi-supervised algorithm itself tries to include information from unlabeled data without human interference.

To address the scarce training data issue, a new semi-supervised approach is proposed in the second publication:

- A teacher-student learning based on label propagation for deep semi-supervised learning is proposed.

- The method is based on the feature space of the trained CNN.

- the method is not dependent on heavy augmentation technique that is not desired in this application.

- A limited number of labeled samples starting from 15 samples and unlabeled samples are considered to efficiently train the models.

- Our method reduced the dependence on labeled samples, which is very time-consuming and costly to collect for sea ice analysis.

- We have also shown that by adding more unlabeled samples, the performance of the inference results has improved.

Since unlabeled data is available more than labeled data, the size of training data can be increased easily. Regarding the computational intensity of deep learning, the training time can be significantly increased in this situation. It leads to a consequential problem for hyperparameter optimization. This issue is considered in the third publication.

# 4

# Distributed Deep learning for Scalable Computing

Deep Neural Networks DNNs are rapidly taking over various aspects of our daily lives and apply for many different applications. In particular, the disruptive trend toward big data has led to an explosion in the size and availability of training datasets for machine learning tasks. Training such models on large datasets to convergence can easily take weeks or even months on a single GPU [20, 95]. However, DNNs rise into prominence is tightly coupled to the available computational power. Often, A single machine can't finish a training task in the desired time frame [96].

An effective remedy to this problem is to utilize multiple GPUs to speed up training. Scale-up approaches rely on tight hardware integration to improve the data throughput. These solutions are effective but costly. Furthermore, technological and economic constraints impose tight limitations on scaling up [97]. In contrast, distributed deep learning (DDL) aims at scaling out to train large models using the combined resources of clusters of independent machines [97].

DDL can be studied from various perspectives, including the type of parallelism, type of concurrency, type of aggregation, and type of communication. In this section, different proposed methods are discussed based on different views.

## 4.1 Type of Parallelism

In general, distributed deep learning can be divided into two main approaches: model parallelism and data parallelism.

### 4.1.1 Model parallelism

Model parallelism distributes model parameters to several computational workers [98]. Each computer worker is in charge of distinct model parameters or layers. Because various neurons in the deep model are highly dependent on one another, each worker should share its output results with the other workers before proceeding with the calculation of the next layer. One prominent advantage of model parallelism is that training large models becomes viable since each worker may only keep a subset of the model, resulting in a reduced memory need.

However, the model parallelism approaches have numerous significant flaws, including unbalanced parameter sizes and significant computing reliance in different levels of the deep model [98]. It isn't easy, if not NP-complete [99], to divide the learning model into appropriate sections and allocate them to distinct compute nodes. Furthermore, an intricate technique must be designed to ensure the durability of model parallelism, which is challenging owing to computational reliance. Because of these difficulties, using model parallelism to expedite training is not straightforward [100].

This approach can conserve memory (since the complete network is not stored in one place) but incurs additional communication after every layer [24]. Model partitioning can be conducted by applying splits between neural network layers (=vertical split) or by splitting the layers (=horizontal split), as depicted in Figure 4.1.

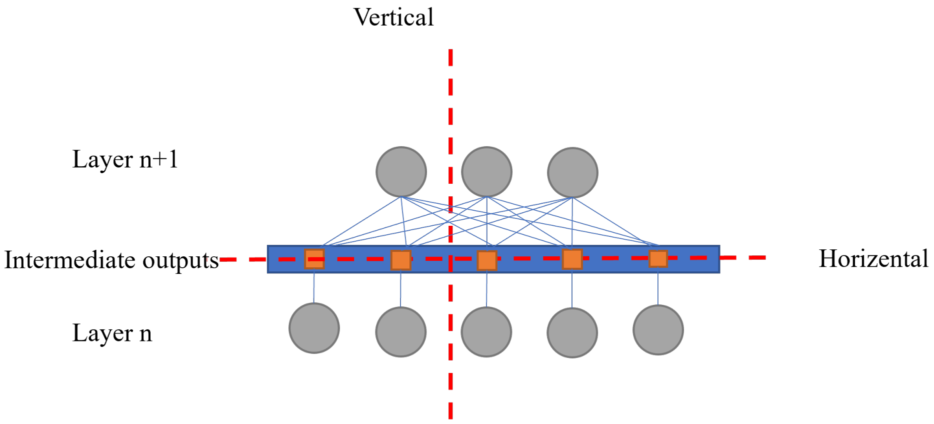


**Figure 4.1:** Model parallelism. It can achieve through vertical or horizontal splits [97].

### 4.1.2 Data Parallelism

The other prominent form of distributed training is data parallelism [101, 102, 103, 104], which is seen in Figure 4.2. In this case, the model parameters are duplicated to all computer workstations, resulting in a faster training time. Each computing worker analyses separate mini-batches of data in a single iteration to calculate the local gradient changes, which are then exchanged with the other workers before updating the model parameters to the latest values.

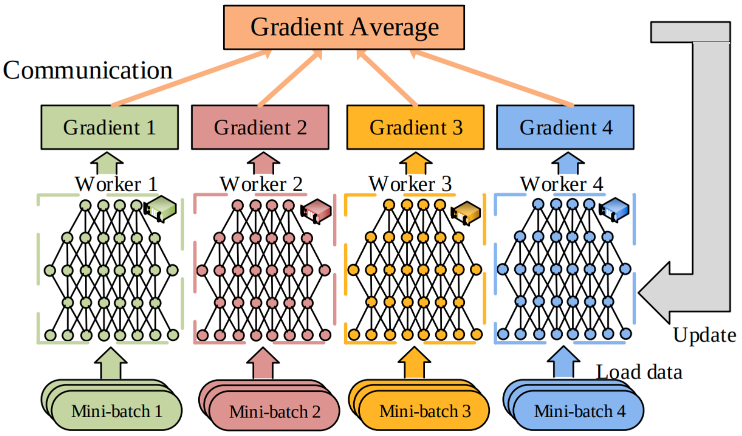


**Figure 4.2:** Data parallelism [25].

## 4.2 Type of Aggregation

From an aggregation view, we can have two different architectures, centralized and decentralized approaches.

In a centralized approach, one or several central servers, often called parameter server, is responsible for updating a specific model parameter [103]. Parameter servers depend on the gradients computed by cluster nodes that perform back-propagation (workers). Figure 4.1 Illustrates the data flow during training in such a system. Centralized optimization allows the expensive task of computing per-parameter gradients to be distributed across the cluster machines. It also elegantly handles updating the model by pooling all communication at the parameter server. Thereby, per-parameter gradients for large amounts of training, samples can be computed quickly. For large clusters, this frequent need for communication focused on the same network endpoints can quickly become a bottleneck [105, 106]. Therefore, most centralized-optimization-based DDL implement communication patterns where the parameter server role is distributed [97].

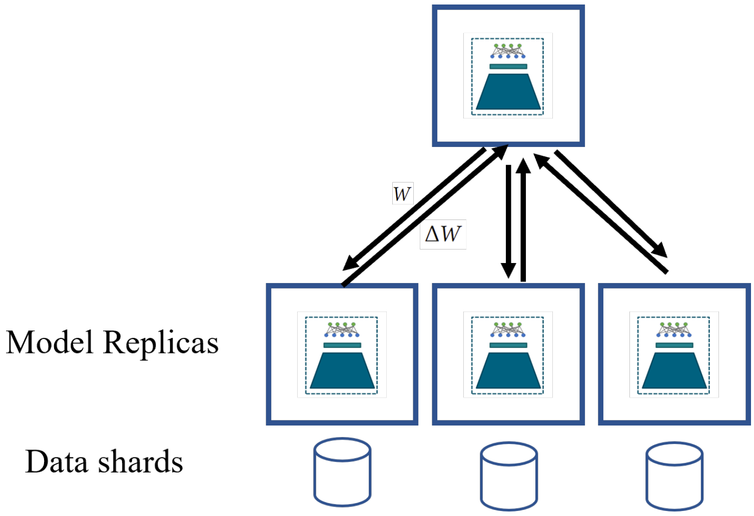


**Figure 4.3:** Centralized approach.

On the other hand, the decentralized approach treats its workers as a swarm, in which each worker independently probes the loss function to find gradient descent trajectories to minima that have good generalization properties [107]. The famous approach in decentralized approaches are collective algorithms like Allreduce, and Ring-Allreduce [108].

### 4.2.1 Allreduce Algorithm

To describe Allreduce, we consider P as the total number of the processes, and each process is uniquely identified as a number between 1 and P. We consider each process has an array of values (network parameters) that should be aggregated with other arrays on other processes. In this case, each process divides its own array into P subarrays, which we refer to as “chunks.” Let chunk[p] be the p-th chunk. These processes communicate in a ring mode, as in Figure 4.4. Each process sends chunk[p] to the next process while simultaneously receiving chunk[p-1] from the previous. Then, process p performs the aggregation operation to the received chunk[p-1] and its own chunk[p-1] and sends the aggregated chunk to the next process p+1. In other words, every chunk travels all around the ring and accumulates a chunk in each process. After visiting all processes once, it becomes a portion of the final result array, and the last-visited process holds the chunk. Finally, all processes can obtain the complete array by sharing the distributed partial results. This process is achieved by circulating step again without aggregation operations, i.e., merely overwriting the received chunk to the corresponding local chunk in each process. The AllReduce operation completes when all processes obtain all portions of the final array [109, 110].

The aggregation operations of SUM, MAX, and MIN are frequently used. In distributed deep learning, the SUM operation is used to compute the mean of gradients. By repeating the receive-aggregate-send steps P-1 times, each process obtains a different portion of the resulting array.

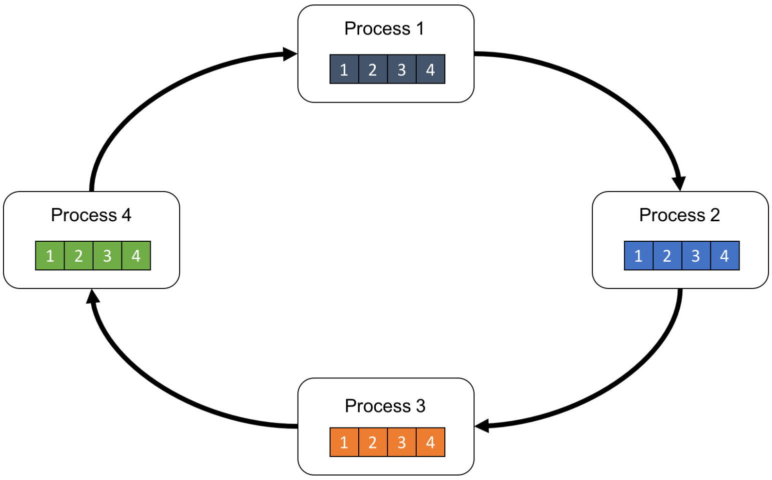


**Figure 4.4:** Illustration of Ring Allreduce.

## 4.3 Type of Concurrency

### 4.3.1 Concurrency in Network

In this category, we compute the output of the layers or the whole network in a concurrent mode for the forward evaluation and backpropagation phases. For example, model parallelism divides the work according to the neurons in each layer. Different parts of the DNNs are computed on distinct processors in various machines [24, 111]. With data parallelism, several replicas of a neural network model are created during training, each on a different worker (processor). For example, the replicas of the model are synchronized (i.e., either average gradients or parameters) at every step by communicating either with a centralized parameter server [103, 112] or decentralized using Allreduce [113, 106, 109].

### 4.3.2 Concurrency in Training

In this category, concurrency is used in the training stage. Multiple instances of training processes run independently on different machines. In this sense distributed training of ensembles is an entirely parallel process, requiring no communication between the workers [114]. Ensemble learning requires more memory and computational power in the training and inference phases. Therefore, knowledge distillation has been used in a two-step training to transfer knowledge of an ensemble with several networks to a single network [115, 116, 117].

To handle the problem of two-step training, Zhang et al. [118] investigated how an ensemble of students can learn collaboratively and teach each other throughout the training process. Kim et el. [119] introduced a fusion learning method that trains a robust classifier by integrating feature maps. Park et el. [120] used a feature-level ensemble for knowledge distillation by transferring the ensemble knowledge between multiple teacher networks. Although these methods can be trained in parallel, their main problem is accuracy, where the number of epochs is not taken into account. Codistillation [121] takes advantage of ensemble and mutual learning to speed up the training. Codistillation uses a distillation-like loss that penalizes predictions made by one model on a batch of training samples for deviating from the predictions made by other models on the same batch.

## 4.4 Type of Communication

### 4.4.1 Synchronous

From the communication perspective, DDL can also be divided into synchronous and asynchronous [24] methods. In synchronous systems, all computations happen simultaneously. A global synchronization barrier prevents workers from progressing until all workers are in the same position. By avoiding deviations in progression between workers, synchronous systems can achieve efficient collaborative training at the expense of potentially underutilizing resources. In synchronize approach, the replicas synchronize (i.e., average either gradients or parameters) at every step by communicating either with a centralized parameter server [103] or using all reduce [113]

### 4.4.2 Asynchronous

Using asynchronous systems takes a more relaxed approach to organizing collaboration and avoids delaying the execution of one worker for another. However, asynchronous systems favor a higher utilization of hardware and regard deviations between workers as manageable side effects that can, in certain circumstances, even prove helpful. On the other hand, by relaxing the synchronization restriction and creating inconsistent models, training workers can update gradients asynchronously [122]. The Figure 5.3 depicts synchronous and asynchronous approaches in centralized data parallelism distributed deep learning.

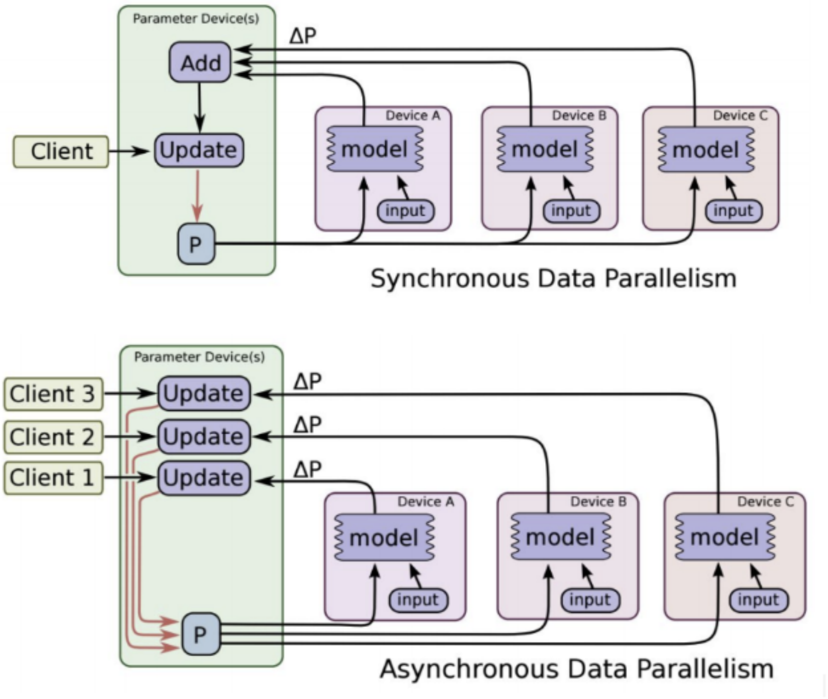


**Figure 4.5:** synchronous and asynchronous in centralized data parallelism distributed deep learning.

## 4.5 Communication Compression

Regarding what to communicate, the exchanged gradients or models may be compressed by quantization or sparsification before transmission over network connections to minimize communication traffic while maintaining model convergence.

### 4.5.1 Quantization

Quantization is a kind of compression technique that utilizes fewer bits to represent data previously encoded in 32 bits on each dimension of the transmitted gradient. As a result, the gradients utilized to improve the models are imprecise after the quantized transmission. Quantization of transmitted gradients is directly related to low-precision deep learning. Deep learning with low precision evolved in an environment where the CPU and GPU need faster calculation and less memory to train DNNs. Quantized communication is feasible if the low accuracy of gradients is sufficient to ensure training convergence. Numerous studies have previously addressed the convergence of deep learning under low-precision gradients [123, 124, 125, 126, 127, 125].

### 4.5.2 Sparsification

Sparsification techniques seek to minimize the number of items conveyed during each repetition. The essential concept of sparsification is that during Stochastic Gradient Descent (SGD) updates, only "significant" gradients are necessary to update the model parameter and ensure training convergence [128]. As seen in Fig. 6, a considerable part of the gradient vector's coordinates are wiped out, eliminating the need to transmit zero-valued components. Gradient sparsification is a more severe compression method than quantization for reducing communication traffic [129, 130, 128, 131].

## 4.6 Communication Overhead Issue

To end this section, we present an example to show the main challenge in distributed deep learning. Most DL models contain a large number of parameters to capture the complex features of the input data and their impact on the prediction results. In the parameter server approach, each worker needs to push the gradient of every parameter and pull every updated parameter from others through a parameter server. As a result, distributed training entails a substantial amount of data exchange between workers and servers.

For example, Table 4.1 shows an experiment using one parameter server and three workers on 1 PS – 3 workers P3.2xlarge Instance Amazon EC2 – Tesla V100 with 10Gb Bandwidth for 100 Iteration [132]. An estimation of the communication time and training time in each iteration is shown considering different network architectures. As we can see, most of the time spends on transferring the models between machines. In fact, different approaches in distributed deep learning try to address communication overhead from different aspects.

## 4.7 Thesis Approach: Distributed Deep Learning

Expanding the deep learning techniques to process big data can significantly improve the overall performance. However, time constraints to training deep learning architectures are one serious obstacle to training a complex DL model.

**Table 4.1:** Communication time versus training time in the parameter server [132].

| Model | # Param. | Data Size (MB) | Training Time | Comm. Time |
|---|---|---|---|---|
| AlexNet [14] | 61.1M | 488 x 4 | 1.99s | 1.56s |
| VGG-16 [95] | 138M | 1104 x 4 | 4.93s | 3.53s |
| VGG-19 [95] | 143M | 1104 x 4 | 5.15s | 3.66s |
| ResNet-152 [133] | 60.2 M | 481.6 x 4 | 1.86s | 1.54s |

It is more significantly important when we have a large amount of training data along with complex DL architectures, especially in semi-supervised learning where we can increase unlabeled data to be involved in the training process easier. Table 4.2 shows the training time considering various semi-supervised methods and our proposed method in paper 2. Although our training dataset is not so sizable, we can observe high training time.

It is important to consider which distributed learning method is more suitable for our proposed supervised and semi-supervised (proposed semi-supervised method is based on extracted feature space of the model) methods. Paper 3 specifically considered the scalability issue and proposed a new method for distributed deep learning.

Our method is a new distributed learning approach based on knowledge distillation. Our method is interesting because:

- Considering the feature space of distributed models, it is related to our proposed method in supervised and semi-supervised, and we can apply this method to extend them to distributed training.
- Our method provides more parallelism while reducing the communication cost.
- It can be used on commodity hardware in contrast to other approaches that need high-speed and low latency networks.

**Table 4.2:** Training time for different methods using 60 labeled data and 10000 unlabeled data on the UiT training dataset [134] on a single GPU (Quadro RTX 5000 16GB).

| Methods | # Batch Size. | Training time |
|---|---|---|
| LP-DeepSSL(WideResNet) [73] | 100 | 2h |
| LP-DeepSSL [73] | 100 | 3h |
| TSLP-SSL [135] | 100 | 6h |
| MixMatch (WideResNet) [89] | 100 | 10h |
| SGANs [80] | 64(best) | 1h |

# 5

# Overview of Publications

In this chapter, we present a summary of three publications and other contributions.

## 5.1 Paper Summaries

### 5.1.1 Paper 1

Salman Khaleghian, Habib Ullah, Thomas Kræmer, Nick Hughes, Torbjørn Eltoft, Andrea Marinoni, **"Sea Ice Classification of SAR Imagery Based on Convolution Neural Networks"**, Remote Sensing, 2021, 13(9).

In the first paper, different training approaches in deep learning have been studied and a modified version of famous architecture called VGG-16 has been proposed. Most State-of-the-art methods are largely based on traditional statistical and machine learning methods. DL architectures for sea ice classification are in an early stage of development. In this sense, most works use basic simple shallow architecture and trivial learning methods. In this paper, the most recent deep architecture networks, and a self-design ad hoc architecture, have been studied. These methods are depicted in Figure 5.1. We consider two main approaches the first consists of modeling a custom or ad hoc architecture to analyze the problem. An ad hoc architecture is interesting as it offers high flexibility, but it generally requires the optimization of many hyper-parameters.

In the second approach, where a given existing DL architecture is used in a new application domain, the existing architecture can either be fine-tuned based on already pre-trained parameters or trained from scratch. This approach significantly reduces the time to design the deep learning architecture. We explore the VGG-16 model [95], well-known network architecture developed for image recognition at the University of Oxford and has achieved high performance in many applications. Different training approaches, including transfer learning and re-training from scratch along with data augmentation, are discussed. We also studied the effects of max-pooling layers in the VGG-16 architecture and proposed a modified VGG-16 model for sea ice classification. We also compared it with three other reference models to show the stability and robustness of our modified VGG-16 model for sea ice classification. These reference models are MobileNetV2 [93], RestNet50 [20], and DenseNet121 [94].

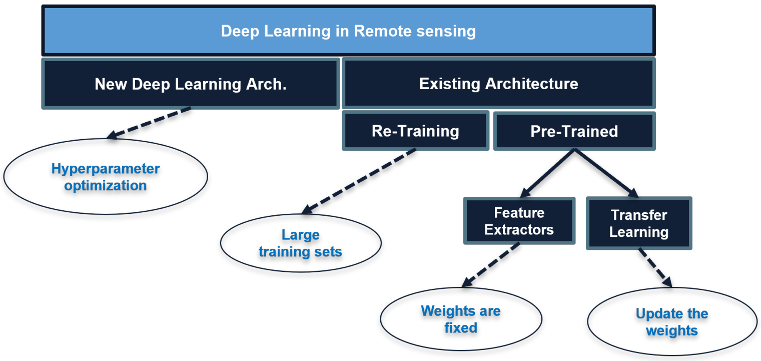


**Figure 5.1:** Paper 1 considered deep learning approaches.

We tested and assessed the results both qualitatively and quantitatively. The results showed that these complex architectures (such as those based on the VGG network) typically obtain promising classification results. Moreover, based on the evaluation of data augmentation, even if the quantitative performance improvement was only minor, the data extension technique seemingly can prevent over-fitting caused by a scarce training dataset. We also assessed the robustness of the trained CNN models when applied to SAR scenes collected at different spatial locations and times. We also found that the additive system noise in the SAR imagery is challenging in obtaining refined sea ice maps. Both the computational requirements and the additive system noise are crucial issues for the operational use of SAR data for sea ice classification. This work highlights the scarcity issue in sea ice classification and shows the significance of involving more training data. This finding was a base for our second paper on semi-supervised learning.

### 5.1.2 Paper 2

S. Khaleghian, H. Ullah, T. Kræmer, T. Eltoft, and A. Marinoni, "**Deep Semi-supervised Teacher–Student Model Based on Label Propagation for Sea Ice Classification**" in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10761-10772, 2021, doi: 10.1109/JSTARS.2021.3119485.

Semi-Supervised Learning (SLL) [16] is employed, to extract accurate information from large-scale datasets, when a limited amount of labeled data are available. These methods aim to combine labeled data with unlabeled records. In the past few years, semi-supervised models have presented performance improvement in various fields of remote sensing research. To address the scarce train data in sea ice classification, we propose a teacher–student-based label propagation deep semi-supervised learning (TSLP-SSL) method. The overall architecture is illustrated in Figure 5.2. Our architecture consists of two models: a teacher and a student model. The teacher model is trained in a two-step procedure. Initially, we trained the teacher model in a supervised fashion utilizing only the labeled data. We then feed both the labeled and unlabeled samples to the trained teacher model and consider the feature space embedding to engender pseudo-labels for the unlabeled data through a label propagation procedure [136, 92, 73]. In the next step, the original and the pseudo-labels are used to train the student model, which is subsequently used during the inference stage. Hence, our proposed method effectively exploits a relatively large amount of unlabeled data to improve the final classification performance.

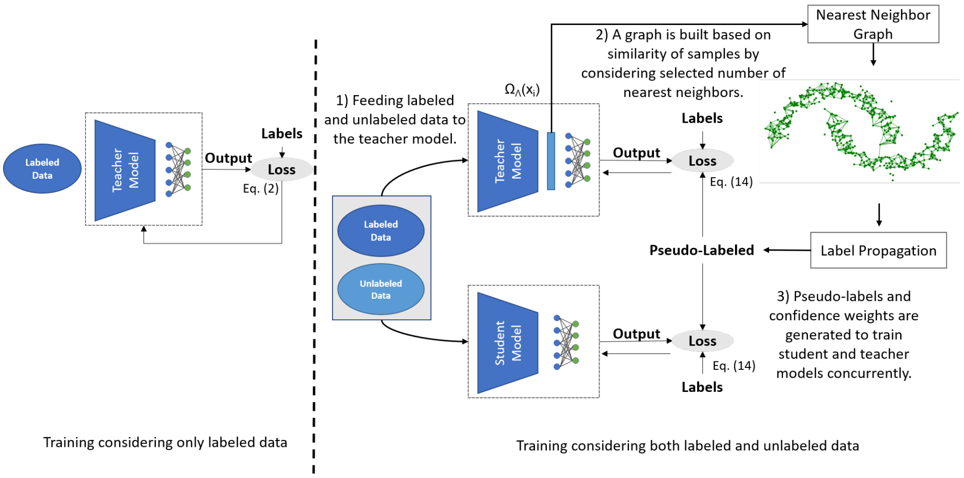

Labeled Data
Teacher Model
Output
Labels
Loss
Eq. (2)
Training considering only labeled data
1) Feeding labeled and unlabeled data to the teacher model.
$\Omega_\Lambda(x_i)$
2) A graph is built based on similarity of samples by considering selected number of nearest neighbors.
Nearest Neighbor Graph
Labeled Data
Unlabeled Data
Teacher Model
Output
Labels
Loss
Eq. (14)
Pseudo-Labeled
Label Propagation
Student Model
Output
Loss
Eq. (14)
Labels
3) Pseudo-labels and confidence weights are generated to train student and teacher models concurrently.
Training considering both labeled and unlabeled data

**Figure 5.2:** Overall architecture of proposed method in paper 2.

To show our proposed method's capabilities, we considered a limited number of labeled samples starting from 15 samples and unlabeled samples to efficiently train the models. In fact, our proposed method was characterized by the ability to learn practical information from both labeled and unlabeled data. Our method reduced the dependence on labeled samples, which is very time-consuming and costly to collect for sea ice analysis. Therefore, this property of our method makes it a good fit for the community of sea ice analysis, where limited labeled data are available.

We have also shown that by adding more unlabeled samples, the performance of the inference results has improved our method can be extended to other problem areas considering the semi-supervised aspect, where a limited number of labeled samples are available. Since unlabeled data is available more than labeled data, the size of training data can be increased easily. Regarding the computational intensity of deep learning, the training time can be significantly increased in this situation. It leads to a consequential problem for hyperparameter optimization. In this regard, the first and second papers show the need for scalable computing, especially when unlabeled data is available. Therefore, the third paper addresses this issue to reduce the training time when a large amount of labeled and unlabeled data is available.

### 5.1.3 Paper 3

S. Khaleghian, H. Ullah, E. B. Johnsen, A. Andersen and A. Marinoni, "AFSD: Adaptive Feature Space Distillation for Distributed Deep Learning," in IEEE Access, vol. 10, pp. 84569-84578, 2022, doi: 10.1109/ACCESS.2022.3197646.

Deep Neural Networks (DNNs) training is very computationally intensive. A single machine often can't finish finishing a training task in the desired time frame. It is more significantly important when we have a large amount of training data along with complex DL architectures, especially in semi-supervised learning where we can increase unlabeled data to be involved in the training process easier. In fact, expanding the deep learning techniques to process big data can significantly improve the overall performance. However, time constraints to training deep learning architectures are one serious obstacle to training a complex DL model.

Through the Ph.D. project, different  methods have been investigated. Paper 3 specifically considered the scalability issue and proposed a new method for distributed deep learning.

Our method is based on knowledge distillation. This method is interesting because: 1) it is related to our proposed method in supervised and semi-supervised, and we can apply this method to extend to distributed training. 2) this method provides more parallelism while reducing the communication cost. And 3) it can be used on commodity hardware in contrast to other approaches that need high-speed and low latency networks.

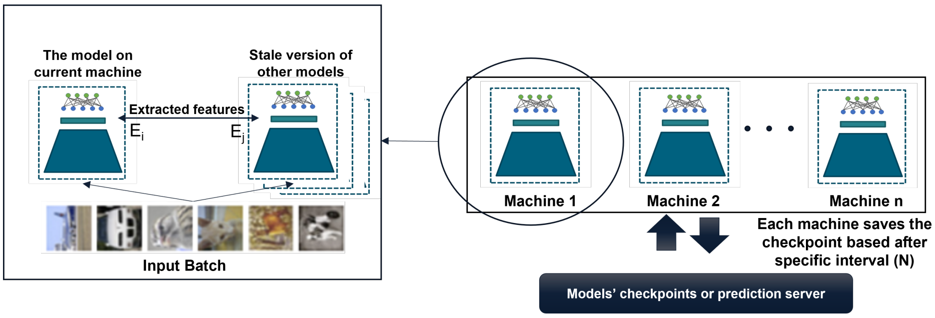

The model on current machine
Stale version of other models
Extracted features
$E_i$
$E_j$
Input Batch
Machine 1
Machine 2
Machine n
Each machine saves the checkpoint based after specific interval (N)
Models' checkpoints or prediction server

**Figure 5.3:** Overall architecture of proposed method in paper 3.

In contrast to the two-step distillation, our method trains n copies of a model in parallel and starts distillation early in the training process. We proposed a new loss function using an additional distillation term based on extracted features by the models. Our proposed method can tolerate longer update interval rates. We rarely update the models but still, we achieve the same accuracy with fewer epochs. In our method, distilling the knowledge between the models less frequently provides flexibility to the models in terms of learning diverse variations in the data. The overall architecture of this method is shown in Figure 5.3.

The proposed method mainly targets distributed deep supervised learning. However, a possible extension of the semi-supervised version has been described in future works.

## 5.2 Other Contributions

**Dataset:** A dataset for sea ice classification based on deep learning methods was published during the PhD Project.

**S. Khaleghian**, J. P. Lohse, and T. Kræmer, “Synthetic-aperture radar (SAR) based ice types/ice edge dataset for deep learning analysis,” 2020. [Online]. Available: https://doi.org/10.18710/QAYI4O.

**Conference Papers:**

**S. Khaleghian**, T. Kræmer, A. Everett, Å. Kiærbech, N. Hughes, T. Eltoft, A. Marinoni, “Synthetic aperture radar data analysis by deep learning for automatic sea ice classification”, EUSAR, Leipzig, Germany, June 2021.

H. Ullah, **S. Khaleghian**, T. Kromer, T. Eltoft and A. Marinoni, "A Noise-Aware Deep Learning Model for Sea Ice Classification Based on Sentinel-1 Sar Imagery," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, 2021, pp. 816-819, doi: 10.1109/IGARSS47720.2021.9553971.

A. Marinoni, G. C. Iannelli, **S. Khaleghian** and P. Gamba, "On the Optimal Design of Convolutional Neural Networks for Earth Observation Data Analysis by Maximization of Information Extraction," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 3505-3508, doi: 10.1109/IGARSS39084.2020.9323521.

Manolis Koubarakis, George Stamoulis, Dimitris Bilidas, Theofilos Ioannidis, George Mandilaras, Despina-Athanasia Pantazi, George Papadakis, Vladimir Vlassov, Amir H. Payberah, Tianze Wang, Sina Sheikholeslami, Desta Haileselassie Hagos, Lorenzo Bruzzone, Claudia Paris, Giulio Weikmann, Daniele Marinelli, Torbjørn Eltoft, Andrea Marinoni, Thomas Kræmer, **Salman Khaleghian**, Habib Ullah, Antonis Troumpoukis, Nefeli Prokopaki Kostopoulou, Stasinos Konstantopoulos, Vangelis Karkaletsis, Jim Dowling, Theofilos Kakantousis, Mihai Datcu, Wei Yao, Corneliu Octavian Dumitru, Florian Appel, Silke Migdall, Markus Muerth, Heike Bach, Nick Hughes, Alistair Everett, Ashild Kiærbech, Joakim Lillehaug Pedersen, David Arthurs, Andrew Fleming, Andreas Cziferszky.” Artificial Intelligence and Big Data Technologies for Copernicus Data: The ExtremeEarth Project.” Conference on Big Data from Space (BiDS21) 2021. Virtual event, 18-20 May 2021.

**Journal papers:**

Desta Haileselassie Hagos, Theofilos Kakantousis, Vladimir Vlassov, Sina Sheikholeslami, Tianze Wang, Jim Dowling, Claudia Paris, Daniele Marinelli, Giulio Weikmann, Lorenzo Bruzzone, **Salman Khaleghian**, Thomas Kræmer, Torbjørn Eltoft, Andrea Marinoni, Despina-Athanasia Pantazi, George Stamoulis, Dimitris Bilidas, George Papadakis, George Mandilaras, Manolis Koubarakis, Antonis Troumpoukis, Stasinos Konstantopoulos, Markus Muerth, Florian Appel, Andrew Fleming, and Andreas Cziferszky. "ExtremeEarth Meets Satellite Data From Space." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2021

**Presentations:**

**Salman Khaleghian**, Thomas Kræmer, Alistair Everett, Åshild Kiærbech, Nick Hughes, Torbjørn Eltoft, Andrea Marinoni. Deep learning for enhanced sea ice understanding. Arctic Frontiers 2020, Tromsø, Norway, January 26 - 30, 2020

**Salman Khaleghian**, Habib Ullah, Thomas Kræmer, Torbjørn Eltoft, Andrea Marinoni, "A deep semi-supervised learning method based on transductive label propagation for sea/ice classification", NORA Conference 2021, Bergen, Norway.

This work is funded in part by Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA) and the Research Council of Norway (RCN Grant no. 237906), the European Union's Horizon 2020 research and innovation programme ExtremeEarth project, grant agreement no. 825258 (http://Earthanalytics.eu/) and by the Fram Center under the Automised Large-scale Sea Ice Mapping (ALSIM) "Polhavet" flagship project. In this regards, most of the findings has been presented in the project meetings.

# 6

# Paper 1: Sea Ice Classification of SAR Imagery Based on Convolution Neural Networks

*remote sensing*

MDPI

*Article*

# Sea Ice Classification of SAR Imagery Based on Convolution Neural Networks

**Salman Khaleghian [1,*], Habib Ullah [1], Thomas Kræmer [1], Nick Hughes [2] Torbjørn Eltoft [1] and Andrea Marinoni [1]**

[1] Department of Science and Technology, UiT the Arctic University of Norway, NO-9037 Tromsø, Norway; habib.ullah@uit.no (H.U.); thomas.Kramer@uit.no (T.K.); torbjorn.eltoft@uit.no (T.E.); andrea.marinoni@uit.no (A.M.)

[2] Norwegian Ice Service, Norwegian Meteorological Institute, P.O. Box 6314 Langnes, NO-9293 Tromsø, Norway; nick.hughes@met.no

* Correspondence: salman.khaleghian@uit.no

**Abstract:** We explore new and existing convolutional neural network (CNN) architectures for sea ice classification using Sentinel-1 (S1) synthetic aperture radar (SAR) data by investigating two key challenges: binary sea ice versus open-water classification, and a multi-class sea ice type classification. The analysis of sea ice in SAR images is challenging because of the thermal noise effects and ambiguities in the radar backscatter for certain conditions that include the reflection of complex information from sea ice surfaces. We use manually annotated SAR images containing various sea ice types to construct a dataset for our Deep Learning (DL) analysis. To avoid contamination between classes we use a combination of near-simultaneous SAR images from S1 and fine resolution cloud-free optical data from Sentinel-2 (S2). For the classification, we use data augmentation to adjust for the imbalance of sea ice type classes in the training data. The SAR images are divided into small patches which are processed one at a time. We demonstrate that the combination of data augmentation and training of a proposed modified Visual Geometric Group 16-layer (VGG-16) network, trained from scratch, significantly improves the classification performance, compared to the original VGG-16 model and an ad hoc CNN model. The experimental results show both qualitatively and quantitatively that our models produce accurate classification results.

**Keywords:** convolutional neural network; ice edge detection; polar region; Sentinel-1; sea ice classification; synthetic aperture radar

**Citation:** Khaleghian, S.; Ullah, H.; Kræmer, T.; Hughes, N.; Eltoft, T.; Marinoni, A. Sea Ice Classification of SAR Imagery Based on Convolution Neural Networks. *Remote Sens.* **2021**, *13*, 1734. https://doi.org/10.3390/rs13091734

Academic Editor: John Paden

Received: 1 March 2021
Accepted: 22 April 2021
Published: 29 April 2021

## 1. Introduction

Sea ice is a key environmental factor [1] that significantly affects polar ecosystems. Over the past decade, the Arctic has experienced dramatic climate change that affects its environment, ecology, and meteorology. The trends are more pronounced than in other regions, and this has been called the Arctic amplification [2] resulting in increasingly variable Arctic weather and sea ice conditions. These are already more extreme than at lower latitudes, and present challenges and threats to maritime operations related to resource exploitation, fisheries, and tourism in the northern areas [3,4]. Therefore, reliable and continuous monitoring of sea ice dynamics, coverage, and the distribution of ice types is important for safe and efficient operations, in addition to supporting detection of how the conditions are changing over longer timescales [5,6]. For example, Ren et al. [7] classified sea ice and open water from synthetic aperture radar (SAR) images using the U-Net model, and integrated a dual-attention mechanism into the original U-Net to improve the feature representations. Han et al. [8] introduced a method for sea ice image classification based on feature extraction and a feature-level fusion of heterogeneous data from SAR and optical images. Song et al. [9] proposed a method based on the combination of spatial and temporal features, derived from residual convolutional neural networks (ResNet) and long short-term memory (LSTM) networks that allowed the extraction of spatial feature vectors for a time series of sea-ice samples using a trained ResNet network. Then, using the feature vectors as inputs, the LSTM network further learnt the temporal variation of the set of sea-ice samples. Subsequently, they fed the high-level features into a Softmax classifier to output the most recent ice type.

For sea ice data analysis, SAR imaging plays a key role as the images acquired by air and satellite-borne platforms provide information that is not restricted by environmental factors and, importantly for Arctic monitoring, can continue to be collected during all weather conditions and through the polar night [10].

Recently, Deep Learning (DL) based methods have shown promising results in many application areas, including computer vision [11], information theory [12], and natural language processing [13]. These have been shown to have excellent generalization capabilities, particularly when properly trained on large datasets. These developments have therefore led to a belief that deep neural networks (DNNs) could lead to a significant improvement of automatic sea ice classification, considering the specific challenges related to this task. However, no applications based on this approach have yet made it into operational use.

In our paper, we explore the performance and efficiency of some DL-based methods for sea ice classification from SAR imagery. DNNs are trainable multi-layer architectures composed of multiple feature-extraction stages, succeeded by a fully connected classification module. DNNs may consist of hundreds of layers, and their architecture can be feed-forward or recurrent, having different types of layers and activation functions, and the training can be achieved through many different optimization strategies. A DNN can be built from different combinations of fully connected, convolutional, maxpooling (sub-sampling), or recurrent layers. Due to their deep nature, they are often trained on large datasets, and in general are able to achieve low generalization errors.

A convolutional neural network (CNN) [14,15] is a feed-forward network consisting of only convolutional layers, pooling layers, and fully connected layers. A CNN [16] is the type of DNN which is most commonly applied to analyzing visual imagery. In the convolutional layers, a CNN extracts features from the image in a hierarchical way by using multiple filters. Each filter consists of a set of weight parameters, which are iteratively adjusted and optimised using an optimization algorithm. These filters are applied to an input image to create a feature map that summarizes the presence of detected features in the input. The CNN learns the filter coefficients during training in the context of the specific problem, and uses pooling layers to sub-sample the output in such a way that the most prominent pixels are propagated to the next layer, dropping the rest. Here it provides a fixed sized output matrix, which is translation and rotation invariant.

Sea ice classification based on SAR imagery is a very challenging task because, in addition to the sea ice characteristics, the radar signals are sensitive to imaging geometry, speckle noise [17], and the blurring of edges and strong anisotropies that may be produced by the SAR imaging process based on the backscattering of signals. In the literature, different methods [18–21] for sea ice classification based on SAR imagery have been presented and typically consider traditional machine learning and probabilistic approaches based on shallow learning strategies. Generally, shallow learning relies on handcrafted features like intensities, polarization ratios, and texture features, which may not encode well the large variations that sea ice may display. Therefore, their generalization capabilities are limited.

To address these challenges, we explore deep learning networks for sea ice classification. Inspired by the success of DNNs in general, and CNNs in particular in many applications, we consider two main approaches when exploring DL networks. The first consists of modeling a custom or ad hoc architecture to analyze the problem. An ad hoc architecture is interesting as it offers high flexibility, but it generally requires optimization of many hyper-parameters. In the second approach, where a given existing DL architecture is used in a new application domain, the existing architecture can either be fine-tuned based on already pre-trained parameters, or trained from scratch. This approach significantly reduces the time to design the deep learning architecture. We explore the VGG-16 model [14], which is a well-known network architecture developed for image recognition at University of Oxford, and has achieved high performances in many applications. This architecture is the core in other architectures like the Fully Convolutional Networks (FCN) [22]. Different training approaches including transfer learning and re-training from scratch are discussed in Section 3.1. We also studied effects of maxpooling layers in the VGG-16 architecture and propose a modified VGG-16 model for sea ice classification. The main contributions of this paper are:

1. We present a deep learning based models for sea ice classification based on SAR imagery. One of the major attractions of these models is their capability to model sea ice and water distinctively in SAR images representing different geographic locations and timing.
2. We extensively evaluate the models on our collected dataset and compare it to both a baseline method and a reference method. Our results show that our explored model outperforms these methods.
3. We categorize state-of-the-art methods and present a comprehensive literature review in this area in the next section.

The rest of the paper is organized as follows. In Section 2, related work is presented. Section 3 reports our proposed deep models and training strategies. Section 4 presents the experimental results on multiple SAR scenes. Finally, Section 5 outlines the conclusion and final remarks.

## 2. Related Work

Sea ice type classification is a major research field in the exploitation of SAR images and has been the subject of research for more than 30 years [23]. The literature on this topic is quite extensive, and here we highlight only a few of the more recent studies.

In general, sea ice classification methods fall into three categories: probabilistic/statistical methods, classical machine learning methods, and deep learning based methods.

In the first category, Moen et al. [24] investigated a Bayesian classification algorithm based on statistical and polarimetric properties for automatic segmentation of SAR sea ice scenes into a specified number of ice classes. Fors et al. [25] investigated the ability of various statistical and polarimetric SAR features to discriminate between sea ice types and their temporal consistency within a similar Bayesian framework, finding that the relative kurtosis, geometric brightness, cross-polarisation ratio and co-polarisation correlation angle are temporally consistent, while the co-polarisation ratio and the co-polarisation correlation magnitude are temporally inconsistent. Yu et al. [26] presented a sea ice classification framework based on a projection matrix, which preserves spatial localities of multi-source images features from SAR and multi-spectral images. By applying a Laplacian eigen-decomposition to the feature similarity matrix, they obtained a set of fusion vectors that preserved the local similarities. The classification was then obtained in a sliding ensemble strategy, which enhances both the feature similarity and spatial locality. In a recent paper, Cristea et al. [27] proposed to integrate the target-specific incidence-angle-dependent intensity decay rates into a non-stationary statistical model. The decay of the intensities of co-polarized SAR signals with incidence angle is dependent on the nature of the targets, and this decay impacts the segmentation result when applied to wide-swath images. By integrating the decay into the Bayesian segmentation process, this deteriorating effect is alleviated and cleaner segmentation results are obtained.

In the second category of classical machine learning, Orlando et al. [28] used a multi-layer perceptron classifier to perform multi-class classification considering first-year ice, multi-year ice, icebergs, and the shadows cast by icebergs. Alhumaidi et al. [29] trained a neural network classifier using polarization features alone, and polarization features plus multi-azimuth “look” Ku-band backscatter for sea ice edge classification. Their method demonstrated a slight advantage in combining polarization and multi-azimuth ‘look’ over using only co-polarized backscatter. Bogdanov et al. [30] also used a multi-layer perceptron classifier for sea ice classification in the winter season, based on a multi-sensor data fusion using coincident data from both the ERS-2 and RADARSAT-1 SAR satellites, low-resolution television camera images, and image texture features. They assessed the performance of their method with different combinations of input features and concluded that a substantial improvement can be gained by fusing the three different types of data. Leigh et al. [31] proposed a support vector machine (SVM) based ice-water discrimination algorithm considering dual polarization images produced by RADARSAT-2, and extracting texture features from the gray-level co-occurrence matrix (GLCM), in addition to backscatter features. Lit et al. [32] introduced a sea ice classification method based on the extraction of local binary patterns, and subsequently used a bagging principal component analysis (PCA) to generate hashing codes of the extracted features. Finally, these hashing codes were fed into an extreme learning machine for classification. Park et al. [33] extracted texture features from SAR images and trained a random forest classifier for sea ice classification. Their method classifies a SAR scene into three generalized cover types, including ice-free water, integrated first-year ice, and old ice. Zhang et al. [34] introduced a conditional random fields classifier for sea ice classification for Sentinel-1 (S1) data that has been applied to SAR scenes from the melt season in the Fram Strait region in the Arctic, and is based on the modeling of backscatter from ice and water to overcome the effects of speckle noise and wind roughened open water.

In the third category, we present DL-based methods for sea ice classification. These methods have been widely used in analyzing Earth observation (EO) data, but the literature is very limited when it comes to the analysis of sea ice data. Previous work can be categorized into two main approaches, namely ad hoc architectures and well-established, existing architectures. For ad hoc architectures, one can, for example, freely determine hyperparameters, including the number of layers, the number of nodes in a particular layer, and the training technique. Many researchers have created ad hoc architectures for handling specific problems [35–39]. Some of the popular existing architectures are the AlexNet [40], the VGG net [14], and the GoogLeNet [41]. There are three sub-approaches on how to train the network when considering the use of existing architectures: (1) re-training the architecture from scratch, (2) using transfer learning and fine-tuning the architecture based on problem specific training data, or (3) applying feature extractors. In the case of re-training from scratch on a new training dataset, the weights of the architecture are randomly initialized. In the case of transfer learning, pre-trained weights are copied and fine-tuned with the new data. All weights may be adjusted, or only some of the network's layers are re-trained and fine-tuned with new training data [42]. For example, Castelluccio et al. [43] fine-tuned two existing architectures to perform semantic classification of remote sensing data, namely the CaffeNet and the GoogLeNet, and showed significant performance improvements. Wang et al. [38,39] used deep ad hoc CNNs for ice concentration estimation. Kruk et al. [44] used DenseNet [45] for finding ice concentration and ice types considering dual-polarization RADARSAT-2 SAR imagery by fusing the HH and HV polarizations for the input samples. Han et al. [46] introduced a hyperspectral sea ice image classification method based on spectral-spatial-joint features with deep learning. Initially, they extracted sea ice texture information from the GLCM and then a three-dimensional deep network to extract deep spectral-spatial features of sea ice for classification. Gao et al. [47] proposed a deep fusion network for sea ice change detection based on SAR images. They exploited the complementary information among low, mid, and high-level feature representations, and for optimizing the network's parameters, they used a fine-tuning strategy. Petrou and Tian [48] used a DL approach [49] to predict sea ice motion for several days in the future, given only a series of past motion observations. Their method is based on an encoder-decoder network and to calculate motion vectors, they used sea ice drift derived from daily optical images covering the entire Arctic. Their model learnt long-time dependencies within the motion time series and captured spatial correlations among neighboring motion vectors.

## 3. Method

Our work falls in the third category, namely DL-based methods, and is inspired by the success of the versatile CNNs in many different applications [14]. We perform both binary and multi-class classifications. In binary classification, we categorize different types of sea ice into one class and water into another class. In multi-class classification, we consider four different ice types that correspond to the World Meteorological Organization (WMO) ice types classification [50]. To model the effects of incidence angle, we create a patch-based training dataset which includes incidence angle as a separate image channel. We explore three different CNN models for sea ice classification, including an *ad hoc CNN architecture* designed from scratch, a *VGG-16 model* [14], considering both transfer learning and re-training from scratch, and a *modified version of the VGG-16 model*. The ad hoc architecture is a new CNN, where we explore different numbers of convolutional and maxpooling layers to examine the impact on their classification performance. We also studied effects of maxpooling layers in the VGG-16 architecture and propose a modified VGG-16 model for sea ice classification.

When it comes to the training of the CNN architectures, it is worth noticing that there is no pre-prepared, publicly available sea ice classification datasets. Therefore, in our work, we train and test all the architectures considering a sea ice dataset that we have carefully generated ourselves from a combination of overlapping SAR and optical satellite images, supported by expert evaluations from sea ice analysts. Our dataset consists of 31 SAR images from north of the Svalbard archipelago collected between March and September during 2015–2018. In order to reduce the effect of overfitting of the models during the training process due to scarce training data, we use an augmentation technique to extend the training set.

### *3.1. CNN Models for Classification*

The ad hoc CNN model we investigate consists of three convolution layers along with an equal number of maxpooling layers. For these layers, the number of kernels/filters are 32, 64, and 64, respectively. Our model also consists of three fully connected layers with 1024, 512, and 2 nodes, respectively. We also use dropout, a regularization technique to avoid overfitting, where we set the dropout probability equal to 0.5. The specification of the ad hoc model is depicted in Figure 1.

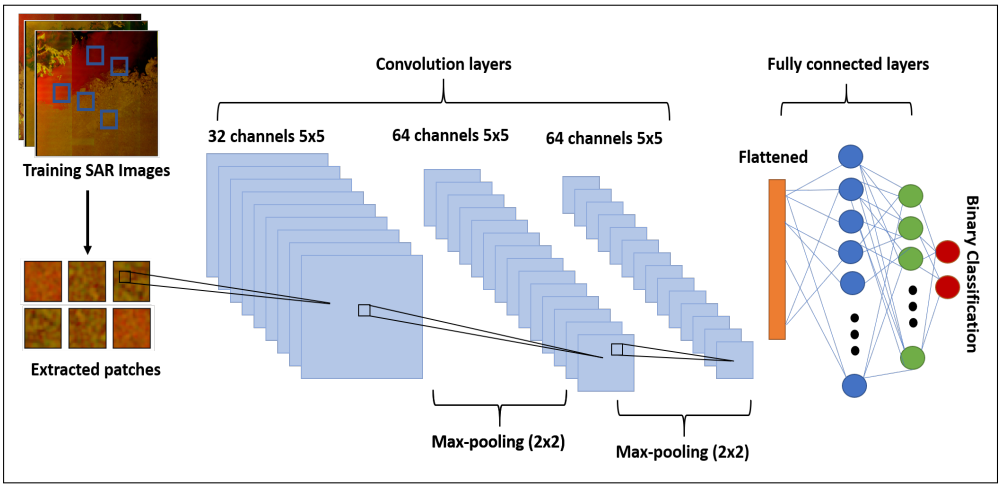


**Figure 1.** Ad hoc CNN. Our proposed CNN architecture consists of three convolution layers and three fully connected layers. At the input, there are training images. We extract patches from these images and feed them to the network during the training process.

We also explore the VGG-16 model [14] for sea ice classification. The architecture of this model is depicted in Figure 2. The fully connected layers are the same for both architectures, but the convolution layers are different, and hence the extracted output features from the convolution layers are not the same. In both architectures, the rectified linear unit (ReLU) activation functions [51] have been used in all layers, except for the last layer, where the SoftMax function [52] is used. We use the cross-entropy loss function and Adam optimizer [53] in the training process. The batch size, which refers t o the number of training examples utilized in one iteration, was set to 50 patches.

In case of the VGG-16 network, we adopt both training from scratch using our sea ice training dataset and a transfer learning strategy. Training from scratch provides insight on the impact of a deeper network in relation to the sea ice classification task. In this case, we adjust all the weights during the training process, starting from a random initialization. For transfer learning, we readjust the weights during training, following the setup obtained for a specific application. We tested each network model with different sizes of the input patches.

Furthermore, we use an augmentation technique to extend our labelled dataset. Here it is supposed to consolidate the architectures both in the feature extraction and the classification stages. In data augmentation, the training data is processed using multiple patch-wise operators and transformations. We used the augmentation strategy of Buslaev et al. [54]. According to the strategy, we perform horizontal flip, rotation with 90 degrees, blurring, and random changes to both brightness and contrast. The data augmentation technique aims to improving the robustness of both architectures by focusing on the structure of the classes, and should help both architectures to be independent of changes in brightness and contrast.

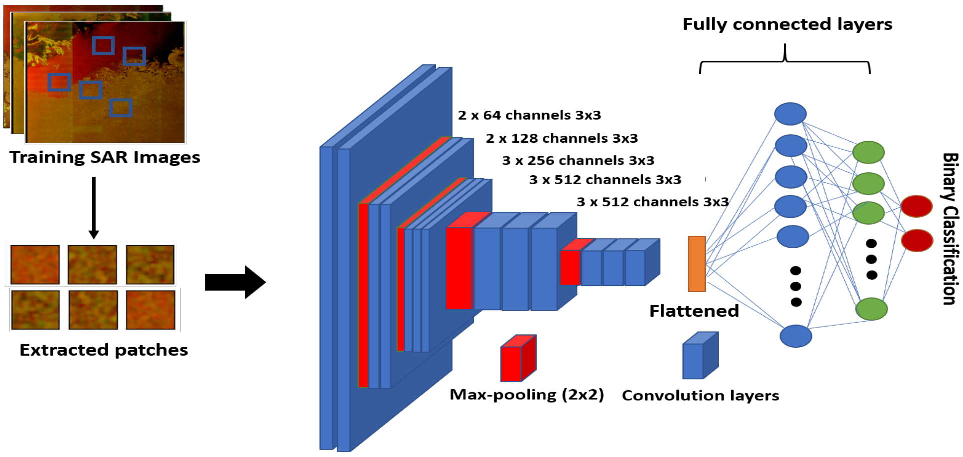


**Figure 2.** VGG-16 CNN. The architecture of this network consists of several convolution layers and three fully connected (FC) layers. The FC layers are represented by dense layers. At the input, there are training images. We extract patches from these images and feed them to the network during the training process.

### 3.2. Modified CNN Model for Classification

We also propose a modified VGG-16 model [14] for sea ice classification. In general, convolutional neural networks introduce equivariance to translation. It means that if an object moves along the height or width axis of an image, the activation translated to the output will be the same. However, this is not true for rotations and changes in the illumination. It can be described intuitively by thinking about a filter that picks up horizontal edges. This filter can find all the horizontal edges in the image, but it cannot detect vertical edges. To this end, a maxpooling layer adds translational invariance [55]. If we consider a pooling layer with a window size of 2 × 2 and a stride equal to 2, it does not matter in which of the four locations the big activation is, the output of the pooling layer will be the same. However, this is not always desired. For example, translation invariance is not desired for face-recognition where the exact distances between eyes and the nose are crucial. In this case, you would not want to reduce the use of pooling.

In sea ice classification, we are not looking for a specific object, and the texture of the classes is important. The use of many maxpooling layers severely affects the networks ability to encode texture characteristics. In a large network, a maxpooling layer shrinks the image size and saves computation. On the other hand, this process limits the minimum input patch size which can be used. For example, the smallest input image that can be used in VGG-16 is 32 × 32. If smaller patches are fed to the network, there will be no output image after the convolution and maxpooling layers to feed to the fully connected layers. We investigate the effect on sea ice classification of *removing the last maxpooling layer in the VGG-16 architecture*. By reducing the maxpooling layers, we suppress the the translational invariance property of the VGG-16 network, and simultaneously reduce the minimum input size that is allowed to be used.

## 4. Experimental Results

### 4.1. Dataset

To test the deep CNN models for sea ice classification, we created an annotated dataset building on the work of Lohse et al. [56]. This is based on 31 Sentinel-1A Extended Wide (EW) Level-1 Ground Range Detected (GRD) scenes, with a spatial resolution of 40 m × 40 m, that were acquired north of the Svalbard archipelago in winter months between September and March during the period 2015–2018. Four sample images from our dataset are shown in Figure 3. Our dataset can be accessed from the provided link https://dataverse.no/dataset.xhtml?persistentId=doi:10.18710/QAYI4O (accessed on 1 March 2021). The images were pre-processed by applying a thermal noise removal algorithm in the European Space Agency (ESA) Sentinel Application Platform (SNAP) software [57], calibrated using the $\sigma_0$ look-up table, and multi-looked using a 3 × 3 boxcar filter. After conversion to dB scale, the images were clipped and scaled linearly in the range [0, 1], considering the dual-polarization intensity channels individually, and including a third input-channel representing the incident angle. The range for the co-polarization (HH) is $-30$ to 0 dB, for the cross-polarization (HV) it is $-35$ to $-5$ dB, and for the incidence angle 19 to 46 degrees. A set of polygons representing homogeneous sea ice types was subsequently manually annotated with labels for those types, taking into account additional information from co-located and nearly temporally coincident optical image data from Sentinel-2. Patches were then extracted from these polygons for 5 different classes representing: Water (including ice-free water (windy), ice-free water (calm), and open water in leads), Brash/Pancake Ice, Young Ice, Level First-Year Ice, and Deformed Ice (including both first-year and multi-year ice). The stride between patches was 10 pixels. In Table 1, we provide the code values for the ice types related to the stage of development (ice age), as defined by the SIGRID-3 vector archive format for sea ice georeferenced information and data ([58]), the class names, and the number of samples for each class for a patch size of 32 × 32. It is worth noticing that we have an imbalanced dataset, where the number of samples for each class has considerable variation. This is a result of the effort we made to accurately annotate the polygons, and hence the number of polygons was small and not representing all classes equally.

For binary Water/Ice classification, we grouped the samples into two classes, namely Water and Ice. Our motivation for performing binary classification is to investigate if deep models can distinguish between sea ice and water, which would subsequently allow for quantitative sea ice concentration mapping. The number of samples of water and ice for different patch sizes are shown in Table 2, and there is not a class imbalance problem in this case. For all the tests, 80 percent of the dataset was used for training and 20 percent for validation.

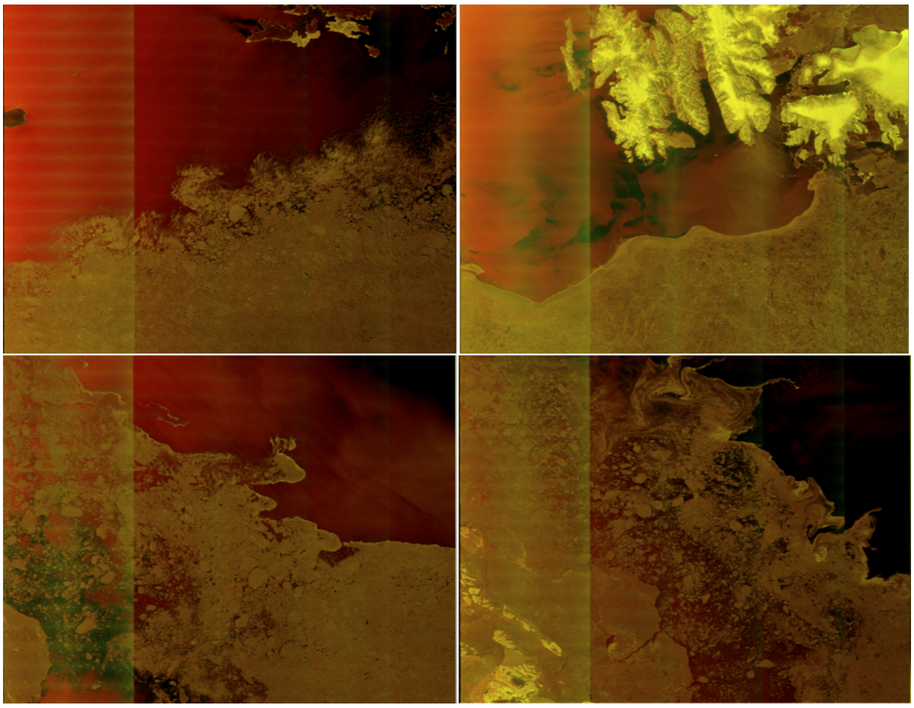


**Figure 3.** Sample images. Four different input SAR images are presented in both the rows from our collected dataset. The light vertical lines in all the images represent SAR additive noise.

**Table 1.** Number of samples for different classes namely open water/leads with water, brash/pancake ice, young ice (YI), level first year ice (FYI), and old/deformed ice.

| Codes | Classes | 32 × 32 |
|---|---|---|
| 02 | Open Water/Leads with Water | 9318 |
| 01–02 | Brash/Pancake Ice | 159 |
| 83 | Young Ice (YI) | 202 |
| 86–89 | Level First-Year Ice (FYI) | 213 |
| 95 | Old/Deformed Ice | 9137 |

**Table 2.** Number of samples for sea/ice classification in different patch sizes. For example, the total number of patches are 19,029 in case of patch size equal to 32 × 32.

| Patch Size | Total | Ice | Sea |
|---|---|---|---|
| 10 × 10 | 22,999 | 12,723 | 10,276 |
| 20 × 20 | 21,020 | 11,301 | 9719 |
| 32 × 32 | 19,029 | 9711 | 9318 |
| 36 × 36 | 18,469 | 9255 | 9214 |
| 46 × 46 | 17,237 | 8255 | 8982 |

We would like to mention that in the inference experiment, we used completely different images. These were another 4 scenes from north of Svalbard, and 8 scenes from Danmarkshavn, East Greenland that were each collected during separate months in 2018.

## *4.2. Model Accuracies*

### 4.2.1. Patch Channels and Sizes

In the first study case, we report the validation accuracy by considering three different channel compositions. We calculate the validation accuracy for a patch by checking if the predicted class is the same as the true class, and by comparing the index of the highest scoring class in the predicted vector with the index of the actual class in the ground truth vector. It is interesting to use the HH polarization alone since it generally has a stronger signal and is less affected by additive noise. However, the HV polarization is more sensitive to ice types during freezing conditions and provides information about the different classes [56]. Furthermore, it is well-known that the radar backscatter from sea ice is dependent on the incidence angle, with lower incidence angles appearing brighter [56]. In order to study the importance of this effect for different classes, we included the incidence angle as a separate input channel. Hence, we consider three alternatives. First, we extracted one-channel patches using only the HH polarization. Secondly, we extracted two-channel patches, with both the HH and HV polarizations as inputs. Finally, we extracted three-channel patches by considering the HH and HV channels, plus the incidence angle. The results for these channel compositions are summarized in Table 3 for the ad hoc CNN, using a patch size of 32 × 32. As can be seen, the composition of the input patches affects performance of the model, with a large improvement due to adding the HV channel to the HH, and another small improvement by adding the incidence angle. The improvement of adding the incidence angle is surprisingly small. However, based on the validation results for the ad hoc CNN, we will use all three channels in our next experiments.

**Table 3.** Validation accuracy of ad hoc CNN for different Patch compositions including HH, HH-HV, and HH-HV-incidence angle. The patch size is equal to 32 × 32 with spatial resolution 1440 $m^2$.

| | **HH** | **HH, HV** | **HH, HV, Incidence Angle** |
|---|---|---|---|
| Validation Accuracy | 88.4% | 98.2% | **98.4**% |

Next, we studied the effect of using different patch sizes for the three-channel case. We consider the ad hoc architecture and input patch sizes of 10 × 10, 20 × 20, 32 × 32, 36 × 36 and 46 × 46, respectively. The validation results in Table 4 show that the accuracy improves with the increase in the patch size. However, this improvement comes at the cost of a lower spatial resolution as larger patches cover wider areas of the surface. Note that for S1 EW GRD images each pixel covers 40 × 40 meters on the Earth surface and, for example, a patch size equal to 46 × 46 covers a 1840 × 1840 square meters area. This patch will be classified as water if the majority of the pixels represent water and would be a problem at ice edges as classification based on larger patches would lead to coarser or non-smooth edges. Hence, there is a trade-off between accuracy and resolution. We used smaller patch sizes in our other experiments.

**Table 4.** Validation accuracy using ad hoc CNN with different patch sizes including 10 × 10, 20 × 20, 32 × 32, 36 × 36, and 46 × 46.

| | **10 × 10** | **20 × 20** | **32 × 32** | **36 × 36** | **46 × 46** |
|---|---|---|---|---|---|
| Validation Accuracy | 95.54% | 97.49 % | 98.53 % | 98.75 % | 99.09% |
| Spatial Resolution (meter) | 400 | 800 | 1280 | 1440 | 1840 |

#### 4.2.2. Different Training Strategies

In this section, we study the performance of the VGG-16 architecture for sea ice classification from SAR images under different training strategies. These strategies include: (a) training the network by transfer learning, where the pre-trained network is trained on the ImageNet dataset, (b) training the network from scratch, (c) training the network from scratch, with an augmented dataset, (d) training the modified VGG-16 network from scratch considering the augmented dataset with a patch sizes equal to 32 × 32, and (e) similar to (d) with a patch size of 20 × 20. Transfer learning and data augmentation are well-known learning strategies that have been successfully applied in computer vision applications. The image formation process for SAR images is fundamentally different from optical images, and our objective here is to understand if these techniques are also suitable for the sea ice classification task using SAR data. For training our model, we consider the learning rate equal to 0.001 and batch size equal to 20.

The number of convolutional layers of the VGG-16 network is different from the ad hoc network. Therefore, the extracted features are different in these architectures, and presumably also their classification performances. We present the classification results related to the VGG-16 network with different training approaches in Table 5. As can be seen, when the network is trained by transfer learning, the validation accuracy is equal to 97.9%, whereas when the same network is trained from scratch, the accuracy is 99.5%. Figure 4 shows the training and validation losses for these two cases, transfer learning in the left panel and 'from scratch' learning in right, respectively. We note that the validation losses in both panels show increasing trends after the point where the training losses indicate conversion, meaning these networks suffer from overfitting.

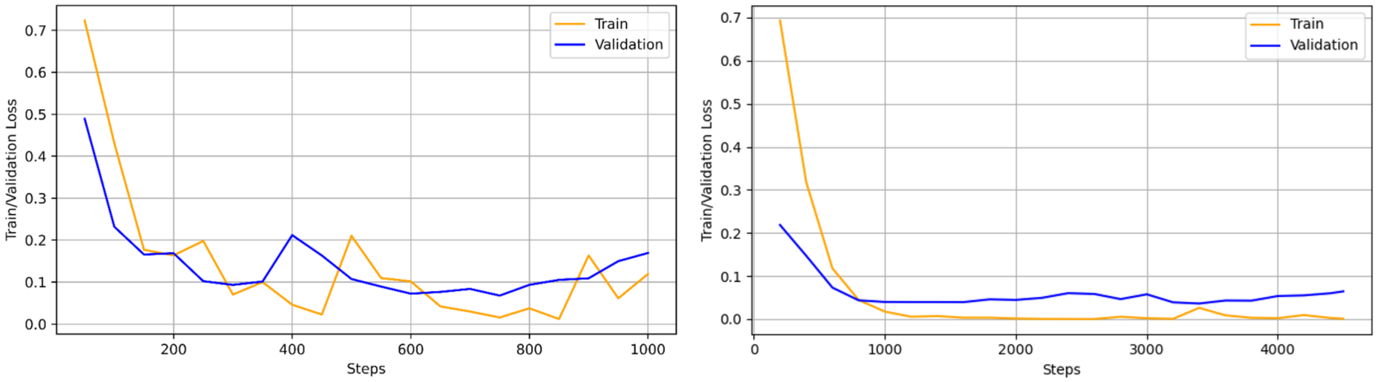


**Figure 4.** Validation and training losses. Considering the VGG-16 network trained with transfer learning (**left**), the validation loss is increasing and the training loss is decreasing. Considering the VGG-16 network trained from scratch (**right**), the Validation loss is increasing and the training loss is decreasing. Therefore, both the networks are facing the problem of overfitting.

**Table 5.** The comparison of the validation accuracy of different models: VGG-16 transfer learning, VGG-16 trained from scratch, VGG-16 trained from scratch with augmentation, VGG-16 modified with augmentation considering two different pixel resolutions.

| Training Strategies | Validation Accuracy | Resolution in Pixels | Resolution in Meters |
|---|---|---|---|
| VGG-16 transfer learning | 97.9 | 32 × 32 | 1280 |
| VGG-16 trained from scratch | 99.5% | 32 × 32 | 1280 |
| VGG-16 trained from scratch with augmentation | 99.79% | 32 × 32 | 1280 |
| VGG-16 Modified + augmentation | 99.89% | 32 × 32 | 1280 |
| VGG-16 Modified + augmentation | 99.30% | 20 × 20 | 800 |

The issue of overfitting is often related to sparse training data, and can be remedied by extending the training set using data augmentation. We demonstrate this for the strategy of training the network from scratch, since as shown in Table 5 it has the best performance. We train the VGG-16 network from scratch with the augmented data according to the augmentation strategy described above. Figure 5 presents the corresponding training and validation loss curves, and, as can be noted, both the validation and training losses are decreasing, hence, showing better generalization capabilities. Table 5 shows that data augmentation also improves the classification results. In fact, we achieve a validation accuracy equal to 99.79%, which is remarkably good.

We also report the validation accuracies of the modified VGG-16 network trained from scratch using the augmented dataset considering two different patch sizes, namely 32 × 32 and 20 × 20. We remind the reader that this architecture is designed to have a reduced number of maxpooling layers, and hence would allow for better texture preservation and smaller input patch sizes. Table 5 also displays the validation accuracies for these models, and as can be noted, the modified VGG-16 networks achieves very high validation accuracies. We also perform a comparison with three other reference models to show the stability and robustness of our modified VGG-16 model for sea ice classification. These reference models are MobileNetV2 [59], RestNet50 [60], and DenseNet121 [45]. The performance of our model in comparison with these reference models is presented in Figure 6 in the form of validation accuracy over time. As can be seen, our model presents higher and consistent validation accuracy.

Based on these experimental analyses, we observe that the modified VGG-16 network trained from scratch with the augmented data provides the highest accuracies. This leads us to conclude that in the case of sea ice classification from SAR data, training the network from scratch with an augmented dataset enables better adjustment and learning of the sea ice characteristics. Transfer learning, with pre-training on ImageNet data, which is fundamentally different from SAR data, does not allow the same adaptation to the data. Moreover, by reducing the number of maxpooling layers, the network better preserves the structure of the data and shows improved performance.

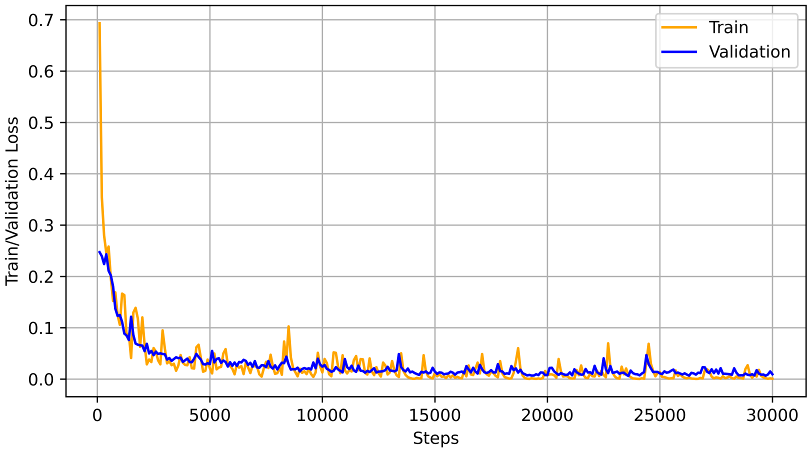


**Figure 5.** Validation and training losses. Considering the VGG-16 network trained from scratch with the augmented data, both the validation and training losses are decreasing showing the better generalization capability of the network.

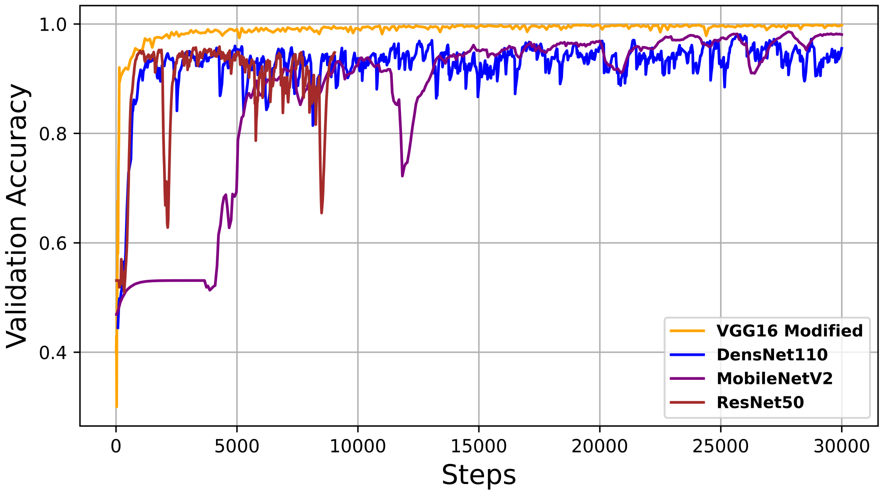


**Figure 6.** Validation accuracy. We present the performance of our model in comparison with MobileNetV2 [59], RestNet50 [60], DenseNet121 [45] in the form of validation accuracy. As can be seen, our model shows higher and consistent validation accuracy over time.

### *4.3. Inference Results*

In order to assess the robustness of the proposed approaches, we investigated the classification results for four new SAR scenes from north of Svalbard, i.e., scenes that are not part of the training data, by presenting the results as qualitative ice versus water maps. To this aim, we set up the inference experiment in a patch-wise manner, where the images are partitioned into non-overlapping patches, and the classification is performed on the entire patches.

Figure 7 shows the four input images from north of Svalbard in the first row. In the same figure, the patch-wise results of the ad hoc CNN are presented in the second row, the results of the VGG-16 model trained with transfer learning are presented in the third row, the results of the VGG-16 model retrained from scratch without the augmented data are presented in the fourth row, the results of the VGG-16 model retrained from scratch with the augmented data are presented in the fifth row, and the results of the modified VGG-16 model trained from scratch with augmented data are presented in the sixth row. Areas consisting of water are annotated in blue and areas consisting of sea ice are annotating in white. For better visualization, we applied a land mask to detect land areas, and the black regions in the images represent land areas. We zoom in on parts of some images to highlight specific details. The classification results obtained with ad hoc CNN (second row) are not satisfactory. The classified images are severely affected by the banding additive noise pattern, as can be clearly seen in columns two and three. The VGG-16 trained with transfer learning (third row) does not classify sea ice areas properly. In fact, open water and newly formed sea ice often have lower radar backscatter values in HV than in HH channels.These cross-polarization values are closer to the noise floor and therefore often have a lower signal-to-noise ratio producing artifacts due to different noise patterns. It can lead to problems during the interpretation of sea ice maps because the added intensity corrupts the true back scattered signal of the sea ice region.

In Figure 7, The VGG-16 retrained from scratch without using the augmented data (fourth row) is better than ad hoc CNN and VGG-16 trained with transfer learning. However, there are still some misclassifications, as can be seen in the first column. The second last row presents the results obtained with the VGG-16 model retrained from scratch with the augmented data. The last row presents the results obtained with the modified VGG-16 retrained from scratch with the augmented data. For the modified VGG-16 model, we reduced the number of maxpooling layers. In this case, the noise seems to be quite well handled, as can be seen in the second column of the last row. However, there is still some noise effects in the third column. Hence, it is worth noticing how the results are affected by the additive noise, which can be seen in the original images (row one) as distinct bands marking the different sub-swaths, and in particular the case when the ad hoc CNN and VGG-16 with transfer learning are considered. Nevertheless, the results obtained by using VGG-16 trained from scratch appear to be more robust against the noise. From this experimental analysis, we conclude that the patch-wise classification results seem to be better when the training data obtained from data augmentation is used to train the VGG-16 model from scratch. The improvement is evident in the last row of Figure 7.

To further show the generalization performance of the CNN models for ice versus water classification, we also tested the models on images acquired from a different Arctic region, the area offshore of Danmarkshavn, East Greenland (76°46′ N, 18°40′ W). Here the Norwegian Meteorological Institute provided vector polygon data representing manually interpreted sea ice areas for the SAR data [61], which consisted of eight images, corresponding to eight different months of the year. These including both the freezing and melting seasons, and were then analyzed with the trained architectures. Figure 8 displays the classification results corresponding to the modified VGG network, trained from scratch with data augmentation, using patch sizes of 32 × 32 and 20 × 20.

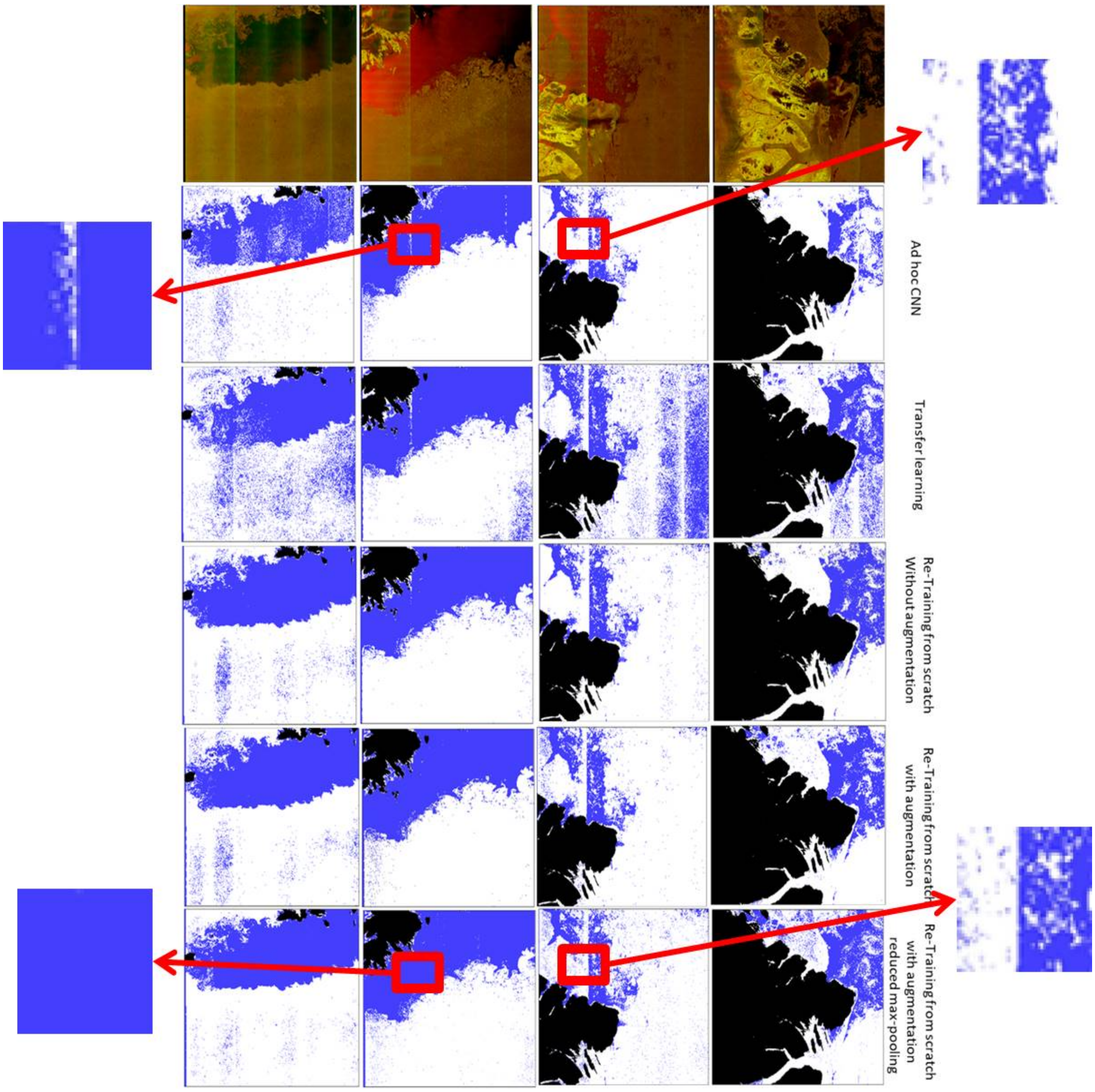

Ad hoc CNN
Transfer learning
Re-Training from scratch Without augmentation
Re-Training from scratch with augmentation
Re-Training from scratch with augmentation reduced max-pooling

**Figure 7.** Patch-wise results considering patch size equal to 32 × 32. The first row presents the original images in two-bands. The second row presents results using ad hoc CNN, the third row presents results using VGG-16 with transfer learning. The fourth row presents results using VGG-16 trained from scratch without augmentation. The fifth row presents results obtained using VGG-16 trained from scratch with augmentation. The sixth row presents the results of modified VGG-16 trained from scratch considering patch size equal to 20 × 20. Ice is annotated in white and water is annotated in blue. The land mask is annotated in black.

As can be seen, the overall performance is good. It is also noticed that the results obtained with patch size equal to 32 × 32 are better than the results obtained with patch size equal to 20 × 20. The larger patch-size seems to be less affected by the noise and therefore we conclude that a patch size equal to 32 × 32 is a better choice for Sentinel-1 SAR images corrupted by additive noise. Overall, our experimental analysis shows that the VGG-16, when trained from scratch with augmented data, presents very good classification results when trained in a supervised fashion.

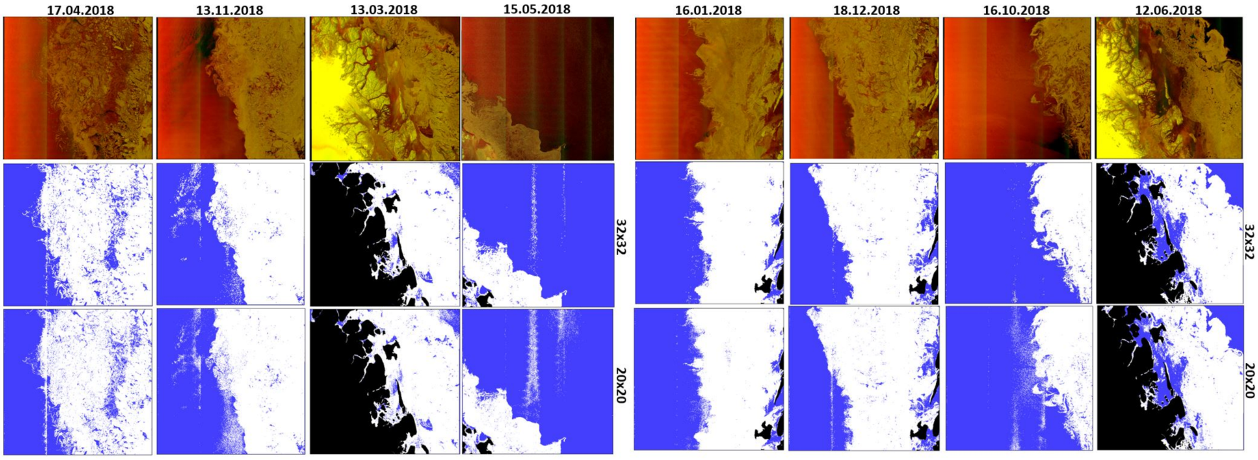


**Figure 8.** Patch-wise inference results of VGG-16 trained from scratch with 32 × 32 and 20 × 20 patches. The network is trained on north of Svalbard images and tested on the new region, Danmarkshavn, East Greenland. Different images in freezing and melting seasons are shown.

To better characterize the quality of the sea ice classification, it is important to distinguish between ice edges and water. Therefore, we also present the performance of our proposed method considering the ice edges of 16 January 2018 as depicted in Figure 9. For this purpose, we overlay the ice polygons (Norwegian Meteorological Institute [61]) from the Danmarkshavn region over the geo-referenced classified image from our method. Overestimation means predicting a larger sea ice area than the manually labelled cover area. Underestimation means predicting a smaller sea ice area than the manually labelled cover area. As can be seen, our proposed method performs effectively to separate ice edges from the water, although there remains some minor overestimation of the sea ice extent in some areas which is preferable to underestimating. However misclassification still occurs in interior areas of the ice pack where there is low backscatter from both cross- and co-polarization such as for areas of level, undeformed landfast ice close to the Greenland coast. An assessment of the accuracy of the ice edge, based on the Integrated Ice Edge Error (IIEE) metric [62], was performed on this example against a selection of other data sources. In Table 6 it can be seen that the contribution to the error from classifying ice as water (under-representing the ice) is consistent with all the products (4646 to 6632 km$^2$) that are compared, as these have fairly good agreement on the presence of landfast ice. There is also a similar level of error against products with accurate ice edges (1522 to 3766 km$^2$) such as the manually analyzed polygons introduced earlier [61], the Norwegian Ice Chart from the Norwegian Meteorological Institute (https://cryo.met.no/en/latest-ice-charts, accessed on 1 March 2021) which is the routine operational analysis produced by an ice analyst, and the sea ice concentration (SIC) produced by the University of Bremen from Advanced Microwave Scanning Radiometer 2 (AMSR2) data [63]. Products based on low resolution passive microwave radiometry, for example the EUMETSAT Satellite Application Facility on Ocean and Sea Ice (OSI SAF) SIC that uses Special Sensor Microwave Imager/Sounder (SSMIS) [64], are less capable of resolving the ice edge, and here there is a far greater contribution to the IIEE (10,797 km$^2$) because the SAR classification correctly identifies sea ice.

**Table 6.** Area differences and IIEE scores for the 16 January 2018 VGG-16 results, (Figure 9) against 4 different sea ice data products: manual analysis [61], Norwegian Ice Chart, OSI SAF SIC [64], and University of Bremen AMSR2 SIC [63].

| Products | Overestimation km$^2$ | Underestimation km$^2$ | IIEE km$^2$ |
|---|---|---|---|
| Manual Analysis [61] | 3766 | 4646 | 8412 |
| Norwegian Ice Chart | 1522 | 6632 | 8155 |
| OSI SAF SIC [64] | 10,797 | 5482 | 16,279 |
| Bremen AMSR2 SIC [63] | 2637 | 5966 | 8604 |

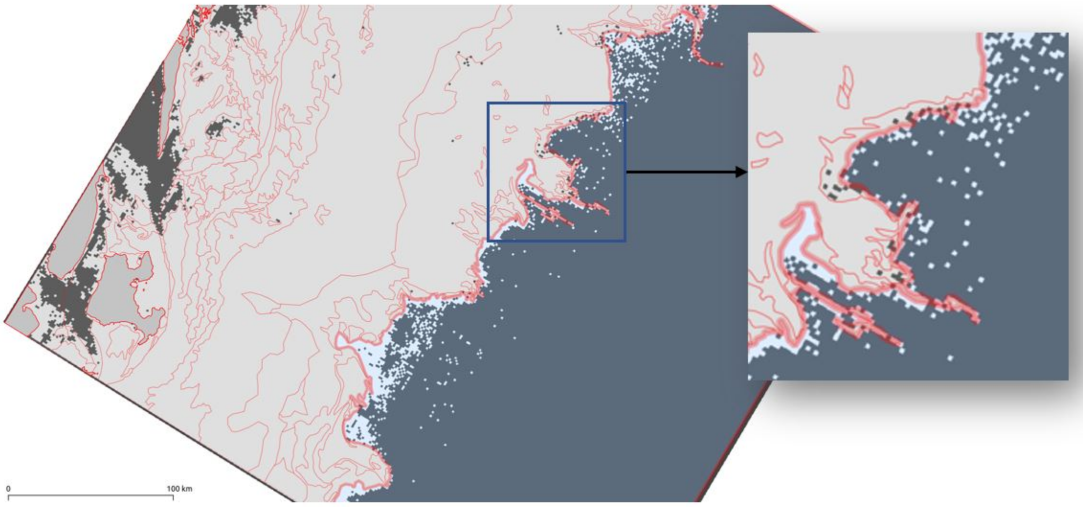


**Figure 9.** Ice edges. We overlay the manually analyzed polygons from the Danmarkshavn region over the classified images from our method to show the effectiveness of our proposed method considering ice edges. The polygons highlighted in the light red color represent the manual analysis, the light grey color represents ice, the dark grey color represent water, and the white color represents overestimated ice from our method.

We have also extended our experimental analysis to multi-class sea ice type classification considering five images from the Danmarkshavn region. The results are depicted in Figure 10. In this classification experiment, we used the modified VGG-16 model trained from scratch with the dataset from north of Svalbard as shown Table 1.

We would like to emphasize that our dataset is scarce and unbalanced, with an unequal number samples from the ice types. This is affecting the classification performance, and the results presented in Figure 10 are slightly biased toward ice types where we have more samples than others. The effect of the imbalance data can be seen in Figure 10, where brash/pancake ice is detected in the right-hand side of the right-most image, which apparently is a dense ice area. In general, brash/pancake ice is located at the edges towards open water. Despite this problem, the results indicate that the VGG-16 trained from scratch shows promising performance in distinguishing different ice types as well as binary ice versus water classification.

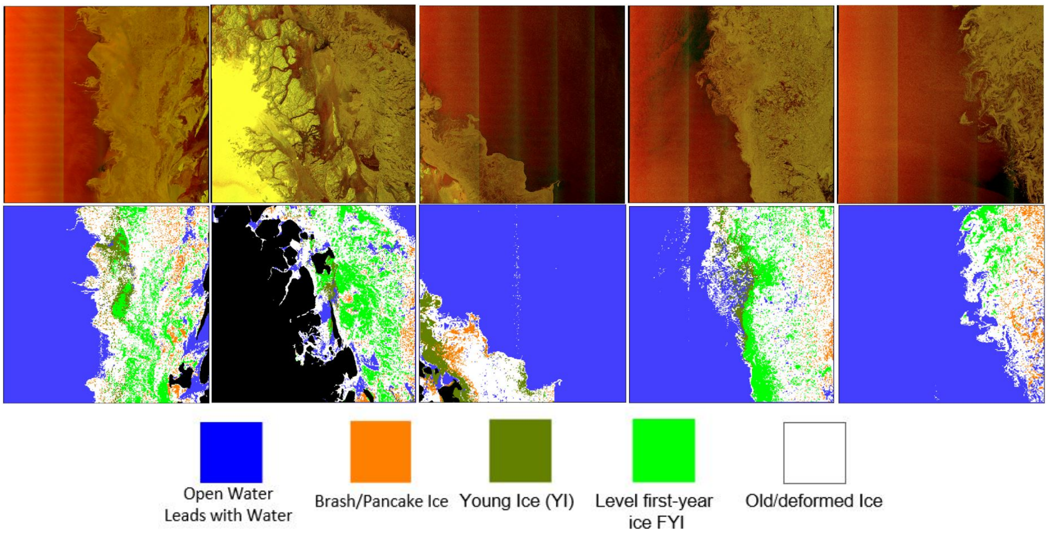


**Figure 10.** Multi-class ice types classification using 32 × 32 patches using the network trained from scratch by considering multi-class ice types in Table 1 from north of Svalbard and tested on the Danmarkshavn region.

We also present the inference result obtained considering only the HH channel. In Figure 11, the left column shows the input SAR image and the right column shows the inference results. As can be seen, the inference result lacks coherency to distinguish sea ice from water. Therefore, both the HV channel and incident angle contribute to the process of properly training the model.

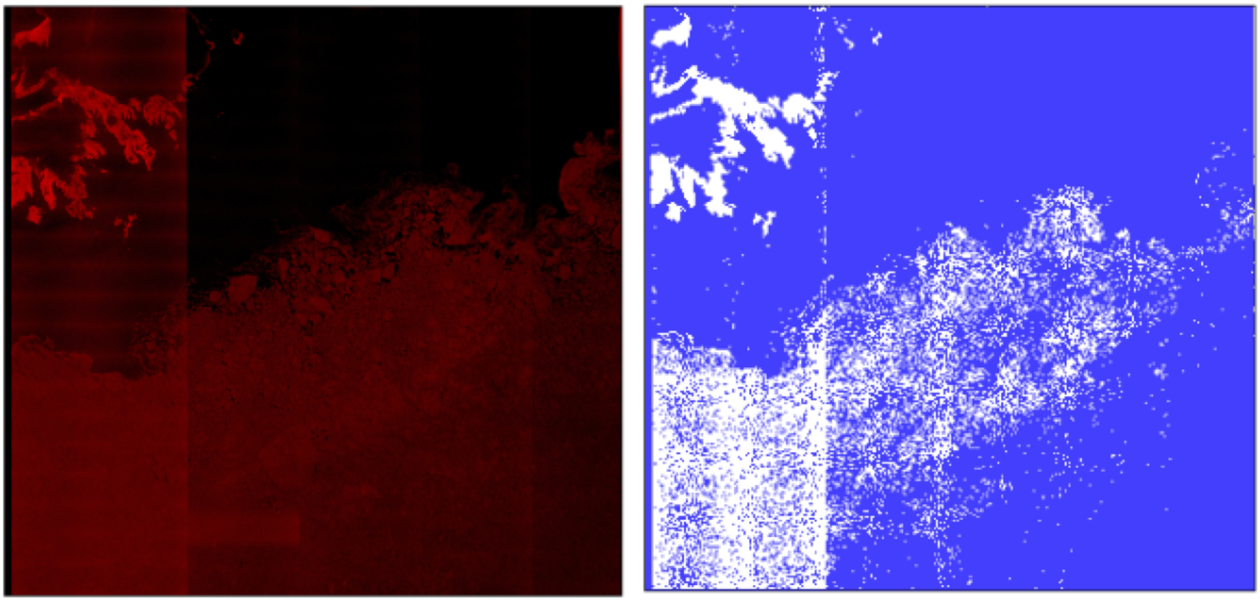


**Figure 11.** The input HH SAR image is shown on the left side and the inference result considering only the HH channel is shown on the right side. The color of the input image is different from the ones reported earlier because in this case we have only HH channel. As can be seen, the result lacks coherency to distinguish sea ice from water.

## 5. Conclusions

In this work, we explored the potential of different CNN models for sea ice classification. We tested and assessed the results both qualitatively and quantitatively. The results showed that these complex architectures (such as those based on the VGG network) typically obtain promising classification results. Moreover, we evaluated the value of data augmentation, and found that even if the quantitative performance improvement was only minor, the data extension technique seemingly can prevent over-fitting caused by a scarce training dataset. We also assessed the robustness of the trained CNN models when applied to SAR scenes collected at different spatial locations and times. Even though our analysis is limited to only a few scenes, our findings are positive and show that the models have good potential. The computational processing to obtain the inference result for a single high resolution SAR image requires a few minutes on a typical desktop computer. We also found that the additive system noise in the SAR imagery is a challenging problem to obtaining refined sea ice maps. Both the computational requirements and the additive system noise are important issues for the operational use of SAR data for sea ice classification.

We also trained our models to perform multi-class classification. In this preliminary study, we had a scarce and unbalanced dataset, which obviously affected the output, but the analysis still showed promise. This motivates us to carry out our research in this direction. In our investigation we performed patch-wise classification which degrades the spatial resolution. Future work will address a pixel-wise setup. However, the pixel-wise set-up will be driven by more computational overhead. Therefore, our future work will also focus on transforming the current architecture to process the input data quickly. For this purpose, we will replace the fully connected layers by convolution layers based on the work of Sermanet et al. [65]. To reduce the impact of noise on sea ice classification, we would include the nominal noise profiles as a feature directly into the model. Finally, we emphasize that the scarcity of reliable and balanced sea ice training and validation datasets is a severe problem for these complex CNN architectures and needs full attention from the sea ice community. In future work, we will develop semi-supervised learning methods to partly remedy this issue.

**Author Contributions:** Conceptualization, formal analysis, validation, S.K.; writing and formal analysis, H.U.; data curation and review, T.K.; writing and editing, validation, visualization, N.H.; review and editing, supervision, T.E.; review and editing, supervision, A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded in part by Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA) and the Research Council of Norway (RCN Grant no. 237906), the European Union's Horizon 2020 research and innovation programme ExtremeEarth project, grant agreement no. 825258 (http://earthanalytics.eu/, accessed on 1 March 2021) and by the Fram Center under the Automised Large-scale Sea Ice Mapping (ALSIM) "Polhavet" flagship project.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** https://dataverse.no/dataset.xhtml?persistentId=doi:10.18710/QAYI4O (accessed on 1 March 2021).

**Acknowledgments:** This work is funded in part by Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA) and the Research Council of Norway (RCN Grant no. 237906), the European Union's Horizon 2020 research and innovation programme ExtremeEarth project, grant agreement no. 825258 (http://earthanalytics.eu/, accessed on 1 January 2020) and by the Fram Center under the Automised Large-scale Sea Ice Mapping (ALSIM) "Polhavet" flagship project.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

We provide the abbreviations used in the paper in this part

| | |
|---|---|
| SAR | Synthetic aperture radar |
| LSTM | Long short term memory |
| DNNs | Deep neural networks |
| FCN | Fully convolutional networks |
| PCA | Principal component analysis |

| | |
|---|---|
| S1 | Sentinel-1 |
| HH | Horizontal-horizontal polarization |
| RLU | Rectified linear unit |
| ResNet | Residual convolutional neural network |
| DL | Deep learning |
| CNN | Convolutional neural network |
| SVM | Support vector machine |
| GLCM | Gray-level co-occurrence matrix |
| EO | Earth observation |
| HV | Horizontal-vertical polarization |
| ESA | European Space Agency |

## References

1. Bobylev, L.P.; Miles, M.W. Sea Ice in the Arctic Paleoenvironments. In *Sea Ice in the Arctic*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 9–56.
2. Serreze, M.; Barrett, A.; Stroeve, J.; Kindig, D.; Holland, M. The emergence of surface-based Arctic amplification. *Cryosphere* **2009**, *3*, 11. [CrossRef]
3. Vihma, T. Effects of Arctic sea ice decline on weather and climate: A review. *Surv. Geophys.* **2014**, *35*, 1175–1214. [CrossRef]
4. Najafi, M.R.; Zwiers, F.W.; Gillett, N.P. Attribution of Arctic temperature change to greenhouse-gas and aerosol influences. *Nat. Clim. Chang.* **2015**, *5*, 246. [CrossRef]
5. Stroeve, J.C.; Serreze, M.C.; Holland, M.M.; Kay, J.E.; Malanik, J.; Barrett, A.P. The Arctic's rapidly shrinking sea ice cover: A research synthesis. *Clim. Chang.* **2012**, *110*, 1005–1027. [CrossRef]
6. Haykin, S.; Lewis, E.O.; Raney, R.K.; Rossiter, J.R. *Remote Sensing of Sea Ice and Icebergs*; John Wiley & Sons: Hoboken, NJ, USA, 1994; Volume 13.
7. Ren, Y.; Li, X.; Yang, X.; Xu, H. Development of a Dual-Attention U-Net Model for Sea Ice and Open Water Classification on SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2021**.[CrossRef]
8. Han, Y.; Liu, Y.; Hong, Z.; Zhang, Y.; Yang, S.; Wang, J. Sea Ice Image Classification Based on Heterogeneous Data Fusion and Deep Learning. *Remote Sens.* **2021**, *13*, 592. [CrossRef]
9. Song, W.; Li, M.; Gao, W.; Huang, D.; Ma, Z.; Liotta, A.; Perra, C. Automatic Sea-Ice Classification of SAR Images Based on Spatial and Temporal Features Learning. *IEEE Trans. Geosci. Remote Sens.* **2021**. [CrossRef]
10. Awange, J.L.; Kiema, J.B.K. Microwave remote sensing. In *Environmental Geoinformatics*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 133–144.
11. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
12. Yan, X.; Cui, B.; Xu, Y.; Shi, P.; Wang, Z. A method of information protection for collaborative deep learning under gan model attack. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**. [CrossRef]
13. Otter, D.W.; Medina, J.R.; Kalita, J.K. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 604–624. [CrossRef]
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Jin, K.H.; McCann, M.T.; Froustey, E.; Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **2017**, *26*, 4509–4522. [CrossRef] [PubMed]
16. Mustaqeem; Kwon, S. A CNN-assisted enhanced audio signal processing for speech emotion recognition. *Sensors* **2020**, *20*, 183.
17. Liu, J.; Scott, K.A.; Gawish, A.; Fieguth, P. Automatic detection of the ice edge in SAR imagery using curvelet transform and active contour. *Remote Sens.* **2016**, *8*, 480. [CrossRef]
18. Lindell, D.B.; Long, D.G. Multiyear Arctic sea ice classification using OSCAT and QuikSCAT. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 167–175. [CrossRef]
19. Shen, X.; Zhang, J.; Zhang, X.; Meng, J.; Ke, C. Sea ice classification using Cryosat-2 altimeter data by optimal classifier–feature assembly. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1948–1952. [CrossRef]
20. Zakhvatkina, N.; Smirnov, V.; Bychkova, I. Sea ice classification based on neural networks method using Sentinel-1 data. *Int. Multidiscip. Sci. GeoConf. SGEM* **2019**, *19*, 617–623.
21. Zakhvatkina, N.Y.; Demchev, D.; Sandven, S.; Volkov, V.A.; Komarov, A.S. SAR Sea Ice Type Classification and Drift Retrieval in the Arctic. In *Sea Ice in the Arctic*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 247–299.
22. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
23. Eppler, D.T.; Farmer, L.D.; Lohanick, A.W.; Hoover, M. Classification of sea ice types with single-band (33.6 GHz) airborne passive microwave imagery. *J. Geophys. Res. Ocean.* **1986**, *91*, 10661–10695. [CrossRef]
24. Moen, M.A.; Doulgeris, A.P.; Anfinsen, S.N.; Renner, A.H.; Hughes, N.; Gerland, S.; Eltoft, T. Comparison of feature based segmentation of full polarimetric SAR satellite sea ice images with manually drawn ice charts. *Cryosphere* **2013**, *7*, 1693–1705. [CrossRef]

25. Fors, A.S.; Brekke, C.; Doulgeris, A.P.; Eltoft, T.; Renner, A.H.; Gerland, S. Late-summer sea ice segmentation with multi-polarisation SAR features in C and X band. *Cryosphere* **2016**, *10*, 401–415. [CrossRef]
26. Yu, Z.; Wang, T.; Zhang, X.; Zhang, J.; Ren, P. Locality preserving fusion of multi-source images for sea-ice classification. *Acta Oceanol. Sin.* **2019**, *38*, 129–136. [CrossRef]
27. Cristea, A.; van Houtte, J.; Doulgeris, A.P. Integrating incidence angle dependencies into the clustering-based segmentation of SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2925–2939. [CrossRef]
28. Orlando, J.R.; Mann, R.; Haykin, S. Classification of sea-ice images using a dual-polarized radar. *IEEE J. Ocean. Eng.* **1990**, *15*, 228–237. [CrossRef]
29. Alhumaidi, S.M.; Jones, L.; Park, J.D.; Ferguson, S.M. A neural network algorithm for sea ice edge classification. *IEEE Trans. Geosci. Remote Sens.* **1997**, *35*, 817–826. [CrossRef]
30. Bogdanov, A.V.; Sandven, S.; Johannessen, O.M.; Alexandrov, V.Y.; Bobylev, L.P. Multisensor approach to automated classification of sea ice image data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1648–1664. [CrossRef]
31. Leigh, S.; Wang, Z.; Clausi, D.A. Automated ice–water classification using dual polarization SAR satellite imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 5529–5539. [CrossRef]
32. Li, Y.; Gao, F.; Dong, J.; Wang, S. A Novel Sea Ice Classification Method from Hyperspectral Image Based on Bagging PCA Hashing. In Proceedings of the 2018 Fifth International Workshop on Earth Observation and Remote Sensing Applications (EORSA), Xi'an, China, 18–20 June 2018; pp. 1–4.
33. Park, J.W.; Korosov, A.A.; Babiker, M.; Won, J.S.; Hansen, M.W.; Kim, H.C. Classification of Sea Ice Types in Sentinel-1 SAR images. *Cryosphere Discuss.* **2019**, 1–23. [CrossRef]
34. Zhang, Y.; Zhu, T.; Spreen, G.; Melsheimer, C.; Zhang, S.; Li, F. Sea ice-water classification on dual-polarized Sentinel-1 imagery during melting season. In Proceedings of the 21st EGU General Assembly, EGU2019, Vienna, Austria, 7–12 April 2019; Volume 21.
35. Nogueira, K.; Miranda, W.O.; Dos Santos, J.A. Improving spatial feature representation from aerial scenes by using convolutional networks. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 289–296.
36. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962.
37. Wang, L.; Wong, A.; Scott, K.A.; Clausi, D.A.; Xu, L.; Shafiee, M.J.; Li, F. Sea ice concentration estimation from satellite SAR imagery using convolutional neural network and stochastic fully connected conditional random field. In Proceedings of the CVPR 2015 Earthvision Workshop, Boston, MA, USA, 11–12 June 2015.
38. Wang, L.; Scott, K.A.; Xu, L.; Clausi, D.A. Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4524–4533. [CrossRef]
39. Wang, L.; Scott, K.; Clausi, D. Sea ice concentration estimation during freeze-up from SAR imagery using a convolutional neural network. *Remote Sens.* **2017**, *9*, 408. [CrossRef]
40. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
42. Nogueira, K.; Penatti, O.A.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556. [CrossRef]
43. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land use classification in remote sensing images by convolutional neural networks. *arXiv* **2015**, arXiv:1508.00092.
44. Kruk, R.; Fuller, M.C.; Komarov, A.S.; Isleifson, D.; Jeffrey, I. Proof of Concept for Sea Ice Stage of Development Classification Using Deep Learning. *Remote Sens.* **2020**, *12*, 2486. [CrossRef]
45. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
46. Han, Y.; Gao, Y.; Zhang, Y.; Wang, J.; Yang, S. Hyperspectral Sea Ice Image Classification Based on the Spectral-Spatial-Joint Feature with Deep Learning. *Remote Sens.* **2019**, *11*, 2170. [CrossRef]
47. Gao, Y.; Gao, F.; Dong, J.; Wang, S. Transferred deep learning for sea ice change detection from synthetic-aperture radar images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1655–1659. [CrossRef]
48. Petrou, Z.I.; Tian, Y. Prediction of Sea Ice Motion With Convolutional Long Short-Term Memory Networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6865–6876. [CrossRef]
49. Mustaqeem; Sajjad, M.; Kwon, S. Clustering-based speech emotion recognition by incorporating learned features and deep BiLSTM. *IEEE Access* **2020**, *8*, 79861–79875. [CrossRef]
50. WMO Sea-Ice Nomenclature, Volumes I, II and III. 2014. Available online: https://library.wmo.int/doc_num.php?explnum_id=4651 (accessed on 1 March 2021).
51. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.

52. Wang, M.; Lu, S.; Zhu, D.; Lin, J.; Wang, Z. A high-speed and low-complexity architecture for softmax function in deep learning. In Proceedings of the 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, China, 26–30 October 2018; pp. 223–226.
53. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
54. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and flexible image augmentations. *Information* **2020**, *11*, 125. [CrossRef]
55. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA. Available online: http://www.deeplearningbook.org (accessed on 1 March 2021).
56. Lohse, J.; Doulgeris, A.P.; Dierking, W. Mapping sea-ice types from Sentinel-1 considering the surface-type dependent effect of incidence angle. *Ann. Glaciol.* **2020**, 1–11. [CrossRef]
57. Piantanida, R.; Miranda, N.; Hadjduch, G.*Thermal Denoising of Products Generated by the S-1 IPF*; S-1 Mission Performance Centre. 2017. Available online: https://sentinel.esa.int/documents/247904/2142675/Thermal-Denoising-of-Products-Generated-by-Sentinel-1-IPF (accessed on 1 March 2021)
58. Joint WMO-IOC Technical Commission for Oceanography. Marine Meteorology. *SIGRID-3 : A Vector Archive Format for Sea Ice Charts: Developed by the International Ice Charting Working Group's Ad Hoc Format Team for the WMO Global Digital Sea Ice Data Bank Project*; WMO & IOC: Geneva, Switzerland, 2004.
59. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
60. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
61. Hughes, N. ExtremeEarth Polar Use Case Training Data 2020. Available online: https://zenodo.org/record/3695276#.X-ytf2j0mUn (accessed on 1 March 2021).
62. Goessling, H.F.; Tietsche, S.; Day, J.J.; Hawkins, E.; Jung, T. Predictability of the Arctic sea ice edge. *Geophys. Res. Lett.* **2016**, *43*, 1642–1650. [CrossRef]
63. Melsheimer, C.; Spreen, G. AMSR2 ASI Sea Ice Concentration Data, Arctic, Version 5.4 (NetCDF) (July 2012–December 2019). 2019. Available online: https://doi.pangaea.de/10.1594/PANGAEA.898399 (accessed on 1 March 2021). [CrossRef]
64. Lavergne, T.; Sørensen, A.M.; Kern, S.; Tonboe, R.; Notz, D.; Aaboe, S.; Bell, L.; Dybkjær, G.; Eastwood, S.; Gabarro, C.; et al. Version 2 of the EUMETSAT OSI SAF and ESA CCI sea-ice concentration climate data records. *Cryosphere* **2019**, *13*, 49–78. [CrossRef]
65. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.

# 7

# Paper 2: Deep Semi-supervised Teacher–Student Model Based on Label Propagation for Sea Ice Classification

# Deep Semisupervised Teacher–Student Model Based on Label Propagation for Sea Ice Classification

Salman Khaleghian, Habib Ullah, Thomas Kræmer, Torbjørn Eltoft, *Member, IEEE*, and Andrea Marinoni, *Senior Member, IEEE*

***Abstract*—In this article, we propose a novelteacher–student-based label propagation deep semisupervised learning (TSLP-SSL) method for sea ice classification based on Sentinel-1 synthetic aperture radar data. For sea ice classification, labeling the data precisely is very time consuming and requires expert knowledge. Our method efficiently learns sea ice characteristics from a limited number of labeled samples and a relatively large number of unlabeled samples. Therefore, our method addresses the key challenge of using a limited number of precisely labeled samples to achieve generalization capability by discovering the underlying sea ice characteristics also from unlabeled data. We perform experimental analysis considering a standard dataset consisting of properly labeled sea ice data spanning over different time slots of the year. Both qualitative and quantitative results obtained on this dataset show that our proposed TSLP-SSL method outperforms deep supervised and semisupervised reference methods.**

***Index Terms*—Deep learning, earth observation, scarce training data, sea ice classification, semisupervised learning (SSL).**

## I. Introduction

ARCTIC sea ice keeps the northern polar regions cool and thereby helps to moderate the global climate. It is a key component of the Arctic environment [1] that substantially affects the polar physical environment and its ecosystems. The Arctic has faced severe environmental impacts over the past few decades. These changes have transformed its environment, ecology, and meteorology and caused unsteady variations in the weather and sea ice conditions, which pose new challenges to maritime industries, including but not limited to aquaculture, natural energy resources, and travel exploration operating in the high north areas [2], [3]. Therefore, proper monitoring of the sea ice conditions and how it changes with time is important [4], [5].

For high-resolution sea ice analysis, researchers and ice centers around the world are using synthetic aperture radar (SAR) data [6], [7]. These data are not restricted by weather conditions and polar darkness [8]. An important part of sea ice analysis includes sea ice classification. Sea ice classification based on SAR data [9] is carried out by classical statistical classification methods, traditional machine learning (TML) methods, and deep-learning-based methods (DLMs). Statistical and TML methods rely on handcrafted features, which may not properly encapsulate the challenging sea ice characteristics [10]. Therefore, their generalization capabilities and their abilities to find efficient features that can be considered to various geographic areas and time frames are limited [10]. DLMs, when properly trained on large training datasets, have shown excellent generalization capabilities in many research fields, including several remote sensing applications such as food security monitoring [11], hybrid data-driven Earth observation modeling [12], and flood mapping from high-resolution optical data [13]. We consider these achievements in the aforementioned fields and believe that deep neural networks (DNNs) may also show performance improvement in automatic sea ice classification [14], [15]. However, scarce training data is the most challenging issue in sea ice data analysis. This problem is particularly challenging in the Arctic, where gathering of precise true observations is expensive, time driven, and sometimes not feasible [16]. For sea ice classification, archived ice charts are available rendering huge labeled data. Nonetheless, these charts are very coarsely labeled and do not have the quality and details needed to train a DLM effectively [17].

To extract accurate information from large-scale datasets, when limited amount of labeled data are available, semisupervised learning (SSL) has been introduced in the technical literature [18]. These methods aim to combine labeled data with unlabeled records. In the past few years, semisupervised models have presented performance improvement in various fields of remote sensing research, such as despeckling of SAR images [19], change detection in heterogeneous remote sensing images [20], and hyperspectral image classification [21]. Considering these successes, we anticipate that deep SSL methodologies could also be favorable in sea ice classification and potentially lead to significant improvements by overcoming the specific challenge of few labeled samples. In fact, a deep SSL technique is halfway

Manuscript received July 1, 2021; revised September 11, 2021 and September 22, 2021; accepted September 23, 2021. Date of publication October 14, 2021; date of current version November 2, 2021. This work was supported in part by the Centre for Integrated Remote Sensing and Forecasting for Arctic Operations and the Research Council of Norway under Grant 237906, in part by the European Union's Horizon 2020 Research and Innovation Program ExtremeEarth Project under Grant 825258, and in part by the Fram Center under the Automised Large-scale Sea Ice Mapping "Polhavet" Flagship Project. *(Corresponding author: Salman Khaleghian.)*

Salman Khaleghian, Thomas Kræmer, Torbjørn Eltoft, and Andrea Marinoni are with the Faculty of Science and Technology, University of Tromsø—The Arctic University of Norway, 9019 Tromsø, Norway (e-mail: salman.khaleghian@uit.no; thomas.kramer@uit.no; torbjorn.eltoft@uit.no; andrea.marinoni@uit.no).

Habib Ullah is with the Faculty of Science and Technology, Norwegian University of Life Sciences, 1430 Ås, Norway (e-mail: habib.ullah@nmbu.no).

Digital Object Identifier 10.1109/JSTARS.2021.3119485

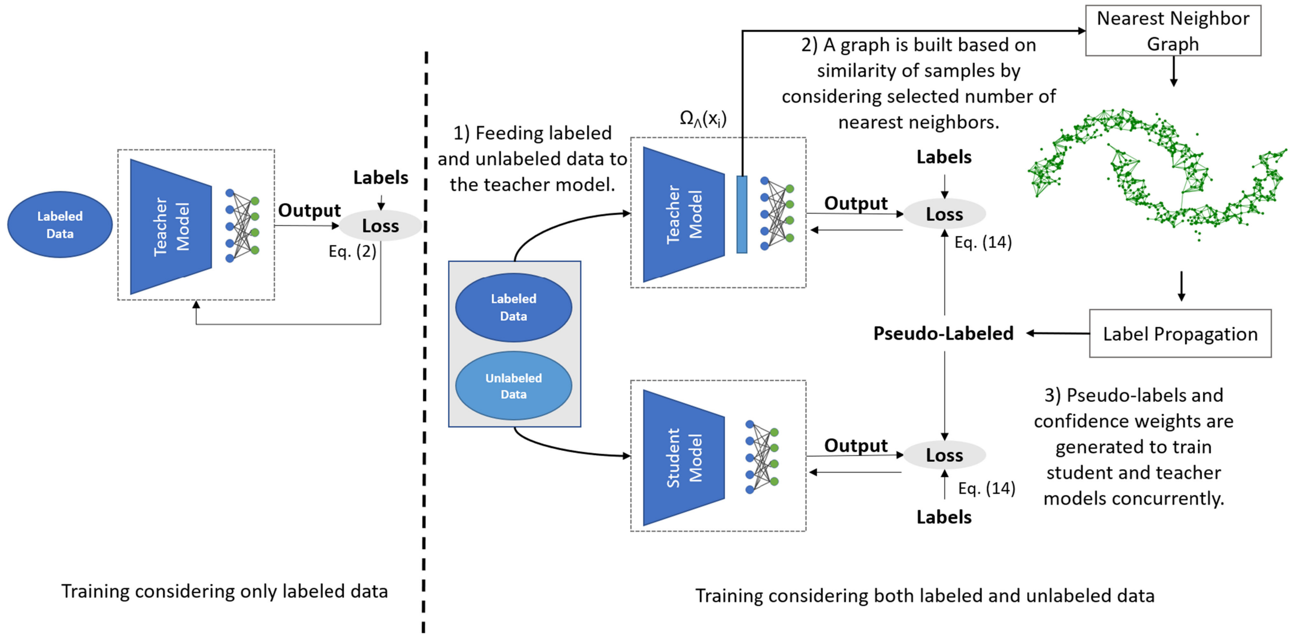

Labeled Data
Teacher Model
Output
Labels
Loss
Eq. (2)
Training considering only labeled data
1) Feeding labeled and unlabeled data to the teacher model.
Labeled Data
Unlabeled Data
$\Omega_\Lambda(x_i)$
2) A graph is built based on similarity of samples by considering selected number of nearest neighbors.
Nearest Neighbor Graph
Teacher Model
Output
Labels
Loss
Eq. (14)
Pseudo-Labeled
Label Propagation
3) Pseudo-labels and confidence weights are generated to train student and teacher models concurrently.
Student Model
Output
Loss
Eq. (14)
Labels
Training considering both labeled and unlabeled data

Fig. 1. TSLP-SSL method. We have two models, namely, teacher and student models. The teacher model is trained on labeled data during the first stage, and then, both models are trained on labeled and unlabeled data during the second stage of the training.

between supervised and unsupervised learning. This technique exploits multiple layers to progressively extract higher level features from the raw input data considering both labeled and unlabeled data.

We propose a *teacher–student*-based label propagation deep semisupervised learning (TSLP-SSL) method. Our architecture consists of two models, namely, a teacher model and a student model. The teacher model is trained in a two-step procedure. Initially, we trained the teacher model in a supervised fashion utilizing only the labeled data. We then feed both the labeled and unlabeled samples to the trained teacher model and consider the feature space embedding to engender pseudo-labels for the unlabeled data through a label propagation procedure [22]–[24]. The original and the pseudo-labels are in the next step used to train the student model, which is subsequently used during the inference stage. The purpose of using the student model is to avoid the problem of the teacher model being biased toward the labeled data, which is like in case of a small training set. Our proposed method, hence, effectively exploits a relatively large amount of unlabeled data to improve the final classification performance. The training methodology is depicted in Fig. 1 and is more thoroughly described in Section III. The summary of our contributions is as follows.

1) We propose a novel TSLP-SSL method. One of the major attractions of our proposed method is its capability to deal with a small number of labeled samples. This is a favorable property in the case of sea ice classification using SAR data, where the availability of a large amount of reliable labeled data is scarce.
2) We consider sea ice datasets to train and analyze the generalization capabilities of our proposed method. We compare our method with a supervised method and three state-of-the-art semisupervised methods. Our results show that our proposed method performs better than all the reference methods, especially in cases with a small number of labeled samples.
3) Additionally, we present a comprehensive literature review covering both the probabilistic learning method and the DLM.

The rest of this article is organized as follows. Related work is described in Section II. We present our proposed deep models and training approaches in Section III. Section IV depicts the experimental analysis considering a set of SAR images. Finally, Section V concludes this article and presents future work.

## II. Related Works

In general, sea ice classification can be divided into two major classes: TML/probabilistic methods and DLMs [25]. The approaches in the latter class fall into two subclasses, namely, supervised deep learning and semisupervised deep learning methods. The literature is very limited in the case of semisupervised DLMs since methods in this subcategory are quite recent and still under development.

### *A. Probabilistic Methods for Sea Ice Classification*

The literature on TML/probabilistic methods is very rich, and we will restrict ourselves to only including a few recent publications. Statistical algorithms often combine probabilistic models and classical classification methods with texture or polarimetric features to perform sea-ice-type maps. An extensive survey is given in [26].

Some specific studies in this category are highlighted below. Examples of machine learning algorithms include the use of standard multilayer perceptrons, as in [14], support vector machines, as in [7], or decision tree methods [15], as in [15]. Statistical and shallow machine learning methods often rely on having extracted the input features in a preoperation prior to the classification. Karvonen [27] and Dinessen [28] used probabilistic and statistical features for estimating sea ice concentration from SAR imagery. Johansson *et al.* [29] used statistical entropy and horizontal–vertical (HV) polarization computations to isolate sea ice from open water and thicker sea ice. Furthermore, Fors *et al.* [30] investigated the potential of $C$- and $X$-band multipolarization SAR features for sea ice segmentation during late summer. Dabboor *et al.* [31] analyzed a set of compact polarimetric parameters for classifying newly formed ice and multiyear ice. Hong and Yang [32] used the statistical coefficient, incidence angle, environment temperature, and speed of wind to improve the sea ice and water classification. Johansson *et al.* [33] used a statistical mixture model to isolate open water from sea ice. Their method is based on the semiautomatic segmentation technique. They applied the algorithm to explore the sea ice characteristics in Svalbard. Aldenhoff *et al.* [34] demonstrated that $C$-band SAR can reliably generate the layout of the ice boundary, whereas the $L$-band shows effectiveness considering thin ice and water regions.

### B. DLMs for Sea Ice Classification

Deep-learning-based approaches have been widely exploited for addressing the challenge of sea ice classification. Malmgren-Hansen *et al.* [17] applied a convolutional neural network (CNN) model to predict Arctic sea ice by fusing data from two different satellites. They found that the CNNs are showing good performance for multisensor data integration. It is worth noting that they used archived ice chart data for both training and validation. However, these data are coarsely labeled, hence leading to undesired effects in the training of the CNN model. Wang *et al.* [10], [35], [36] exploited CNNs for ice concentration estimation. Tom *et al.* [37] proposed an ice monitoring model based on Sentinel-1 data with a deep learning approach. Boulze *et al.* [38] introduced a CNN for detecting different kinds of sea ice [39] using SAR data. They trained the CNN considering the archived ice chart data. They performed comparison with a random forest classifier using texture features.

SSL methods are proposed for classification when only scarce training data or a limited number of training samples are available. The idea of SSL relies on the assumption that unlabeled samples provide essential information and clues on how the data are distributed. Therefore, a DLM can be trained by considering this distribution. In this sense, different approaches such as teacher–student models [40], graph-based methods [41], pseudo-labeling [42], consistency regularization [43], and generative models (i.e., generative adversarial networks—GANs) [44] have been introduced. Shin [40] proposed a multiteacher single-student method to solve the visual attribute prediction problem. His method learnt task-specific domain experts called teacher networks and a student network by forcing a model to imitate the distributions learned by domain experts. Xie *et al.* [45] proposed a noisy student method for generating pseudo-labels to train a model in an iterative way. The output of the trained model based on the labeled samples is exploited to produce pseudo-labels for the unlabeled samples, which are subsequently used to train another model. They used the teacher–student model to train a larger student model by incorporating noise, considering data augmentation (DA), dropout, and stochastic depth. Tarvainen and Valpola [46] proposed a mean teacher method that averages model weights instead of label predictions. Their method improves test accuracy and enables training with fewer labeled samples. Salimans *et al.* [47] trained the semisupervised generative adversarial network (semi-GAN) as a generative model. Kingma *et al.* [48] exploited a variational autoencoder in the form of a semisupervised model. In their method, a classifier is trained on top a latent representation to predict the labels. Iscen *et al.* [24] proposed a transductive label propagation model for deep SSL. This model is trained in an iterative two-step procedure. In the first phase, a CNN is trained using the labeled part of the dataset in a supervised manner. In the second phase, based on a manifold assumption in the feature space of the CNN, pseudo-labels are produced for the unlabeled data through a label propagation procedure using a nearest neighbor graph. The pseudo-labels are considered to extend the set of labeled samples in the second stage to train the CNN model. Berthelot *et al.* [49] used an augmentation technique to introduce an SSL approach. They assumed that the distribution of a classifier should remain the same considering unlabeled data. They used average prediction to produce pseudo-labels for the unlabeled samples.

### C. SSL Methods for Sea Ice Classification

The aforesaid cases show that the development of SSL methods is a hot topic in the data analysis community. However, it is also true that the application of SSL architectures to sea ice classification is very limited. For example, Han *et al.* [50] investigated an approach for sea ice classification based on active learning (AL) and SSL. They acquired the most informative data examples considering AL. They exploited these informative examples in training the SSL method. Staccone [51] introduced an SSL approach based on GANs for sea ice classification. In this work, both labeled and unlabeled data were considered to achieve more accurate results by exploiting the knowledge from both data sources. Li *et al.* [52] presented an SSL method for ice and water classification based on self-training. Their method combined a contextual model and the self-training approach into a unified framework.

Our proposed method falls into the subcategory of SSL methods. We propose a teacher–student model considering the feature space using the label propagation method, which is summarized in the following section.

## III. Teacher–Student-Based Label Propagation Method

As mentioned above, labeled sea ice samples are difficult to acquire, making the training of sea ice classification architectures a difficult task. Therefore, we explore a novel TSLP-SSL method for this application. We adequately utilize a limited number of labeled samples and a comparatively much large number of unlabeled samples to train a deep CNN architecture for extracting sea ice information. Our proposed TSLP-SSL method consists of a teacher model and a student model, which are cooperatively trained in an iterative way during two training stages. Our method is different from the teacher–student models presented in [45] and [46] in two major aspects. First, in our case, features generated by the trained teacher model are extracted before the final classification layer and used in the label propagation process to produce pseudo-labels for the unlabeled samples using a $k$-nearest neighbor approach. Hence, label propagation is performed in feature space, and not in output label space. Second, the pseudo-labels from the teacher model are exploited, together with the original labels, to train the student model in order to find an optimal decision boundary during a second iterative training stage. Our proposed method is also different from the deep SSL model in [24] in the way it aims to avoid the model to be biased toward the labeled data. In fact, the method in [24] is based on a single model, which is trained on only the labeled data, making it susceptible to be biased toward these data samples. The biasing problem may be even more significant in the sea ice classification task, considering the small amount of labeled data and noting the fact that texture features are important for discriminating between different ice types.

In our proposed method, both models are represented by a CNN constructed of a 13-layer architecture [24]. During the first training stage, the teacher model is trained on the labeled data only. During the second stage, the teacher model generates pseudo-labels for the unlabeled data. These pseudo-labels, combined with the labeled samples, are used to train the student model. The motivation for considering an additional student model is to handle the problem of the teacher model being biased toward the labeled data, as discussed above [53]. To further elaborate on this issue, the teacher model formulates a decision boundary considering a small set of labeled data. However, this decision boundary may not be the best boundary when also considering the unlabeled data during the second stage, especially if the teacher model gets overfitted to the labeled data because of the limited number of samples [54]. The idea is that the student model should discover a more appropriate decision boundary, as illustrated in Fig. 2. Fig. 2 displays a simplified case, in which the triangles represent samples from one arbitrary class and the circles show samples from another class. Hence, the red and blue symbols represent labeled data from the two classes, respectively, and the black symbols represent unlabeled data from both classes. Since the teacher model is trained using labeled data only, the decision boundary shown as a blue solid line in Fig. 2 could be a solution. A better decision boundary is discovered by repeatedly training the student model from scratch with both pseudo-labeled and labeled data. In this way, the student model would end up with the decision boundary defined by the green-dashed line, which properly separates both the labeled and unlabeled data from both the two classes. It is worth noting that this example shows the advantage of using label propagation based on nearest neighbors instead of using the network output as pseudo-labels.

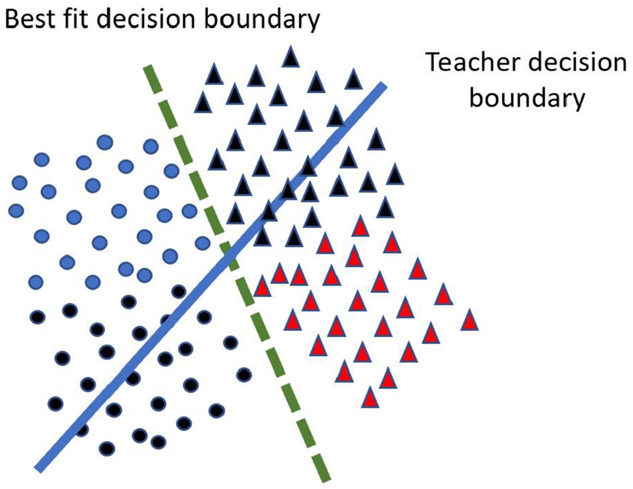


Fig. 2. Complexity of tuning the teacher decision boundary to also take into account the unlabeled data. We show two-class labeled data with red triangles and blue circles. The black markers represent the unlabeled data.

During the second stage of our training, the teacher model generates predictions for the entire dataset. The feature space embedding is subsequently used to construct a nearest neighbor graph and an adjacency matrix, from which we assign pseudo-labels to the unlabeled samples in a transductive label propagation procedure [24].

### A. Formulation for the Learning Process

To clearly provide the details of the process of label propagation for our teacher model, we present the affiliated notations in this section. In this, we will largely follow the outline in [24]. We consider a set of $n$ samples denoted by $X := (x_1, \ldots, x_s, x_{s+1}, \ldots, x_n)$ with $x_i \in X$, where $s$ samples $x_i$ for $i \in S := \{1, \ldots, s\}$, represented by $X_S$, are labeled according to $Y_S := (y_1, \ldots, y_s)$. Each element in $Y_S$ is $y_i \in G$, where $G := \{1, \ldots, g\}$ is a discrete label set of $g$ classes. The rest of the $e := n - s$ samples $x_i$ for $i \in E := \{s+1, \ldots, n\}$, represented by $X_E$, are unlabeled. We consider all samples in $X$ and labels in $Y_S$ to train the CNN to assign class labels to the previously unseen samples. The CNN takes an input sample $x_i$ from $X$ and builds a vector of class probabilities $f_\Lambda(x_i)$, $f_\Lambda : X \rightarrow \mathbb{R}^g$, where $\Lambda$ represents the hyperparameters of our deep model. In this process, the feature extraction stage is represented by the function $\Omega_\Lambda : X \rightarrow \mathbb{R}^d$, which maps the input data to a $d$-dimensional feature vector, where the $i$th sample is represented by $d_i := \Omega_\Lambda(x_i)$. In the next stage, a vector of class probabilities is built by the softmax on top of the fully connected layer considering $\Omega_\Lambda$. The prediction of the CNN for the $i$th sample is the class of the highest probability, i.e.,

$$\hat{y_i} = \text{argmax}_j f_\Lambda(x_i)_j \tag{1}$$

where $j$ is the $j$th dimension of the vector. In supervised learning, the loss function in (2) is minimized to train the CNN

$$\xi_{\text{sup}}(X_S, Y_S; \Lambda) = \sum_{i=1}^{s} \varepsilon_{\text{sup}}(f_\Lambda(x_i), y_i). \tag{2}$$

Equation (2) applies only to the labeled samples, i.e., $x_i \in X_S$. In fact, (2) shows one term of the loss function in SSL. In classification problems, the *cross-entropy* loss function is generally used for $\varepsilon_{\text{sup}}$, which for a given sample $x_i$ is defined as

$$\varepsilon_{\text{sup}}(f_\Lambda(x_i), y_i) = -\sum_{k=1}^{g} y'_k \log\left(f_\Lambda(x_i)\right)_k \tag{3}$$

where $y'_k$ is the $k$th component of the one-hot encoding of $y_i \in G$. Pseudo-labeling finds a pseudo-label $\hat{y}_i$ for each sample $x_i$ for $i \in E$. The pseudo-labels for unlabeled samples in $X_E$ are represented by $\hat{Y}_E = \{\hat{y}_{s+1}, \ldots, \hat{y}_n\}$, and they form an additional loss term formulated as

$$\xi_{\text{pseu}}(X_E, \hat{Y}_E; \Lambda) = \sum_{i=s+1}^{n} \varepsilon_{\text{pseu}}(f_\Lambda(x_i), \hat{y}_i). \tag{4}$$

### B. Pseudo-Label Generation and Learning Process

In our method, the CNN is represented by the parameters $\Lambda$, and we formulate the descriptor set as $D := (d_1, \ldots, d_s, d_{s+1}, \ldots, d_n)$, where $d_i := \Omega_\Lambda(x_i)$. We build a sparse affinity matrix $\Delta \in \mathbb{R}^{n\times n}$, where its elements are represented by

$$\nu_{ij} = \begin{cases} [d_i^T, d_j]_+^\gamma, & \text{if} \quad i \neq j \wedge d_i \in N_k(d_j) \\ 0, \text{otherwise} \end{cases} \tag{5}$$

where $N_k$ represents the set of $k$-nearest neighbors in $X$, and $\gamma$ is a hyperparameter. It is worth noticing that building the sparse affinity matrix is computationally efficient even if we have a very large number of samples. We then build a symmetric adjacency matrix $\Theta = \Delta + \Delta^T$ such that $\Theta \in \mathbb{R}^{n\times n}$. The diagonal of the matrix $\Theta$ consists of zeroes. The rest of the elements of $\Theta$ are nonnegative pairwise similarities between $d_i$ and $d_j$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$. We formulate the symmetrically normalized counterpart of $\Theta$ as

$$\Xi = \Gamma^{-\frac{1}{2}}\Theta\Gamma^{-\frac{1}{2}} \tag{6}$$

where $\Gamma = (\Theta 1_n)$ is the degree matrix and $1_n$ is an $n$-dimensional vector with all elements set to 1. We formulate a label matrix $Y$ of size $n \times g$ consisting of the elements

$$Y_{ij} = \begin{cases} 1, & \text{if} \quad i \in S \wedge y_i = j \\ 0, \text{otherwise} \end{cases}. \tag{7}$$

The rows of the matrix $Y$ represent one-hot encoded labels for the labeled samples. Subsequently, the diffusion amounts to formulating an $n \times g$ matrix $\psi$ such that

$$\psi = (I - \alpha\Xi)^{-1}Y \tag{8}$$

where $\alpha \in [0, 1)$ is a parameter. The elements of $\psi$ are represented by $\delta_{ij}$. In fact, calculating matrix $\psi$, according to (8), is impractical for large $n$ because the inverse matrix $(I - \alpha\Xi)^{-1}$ is not sparse. Therefore, we use the conjugate gradient method to solve the linear system

$$(I - \alpha\Xi)\psi = Y. \tag{9}$$

Equation (9) is fast and valid since the matrix $(I - \alpha\Xi)$ is positive definite. We find the pseudo-labels $\hat{Y}_E = \{\hat{y}_{s+1}, \ldots, \hat{y}_n\}$ for unlabeled samples as

$$\hat{y}_i = \text{argmax}_j \delta_{ij} \tag{10}$$

where $\delta_{ij}$ is the $(i, j)$th element of matrix $\psi$. It is worth noting that finding pseudo-labels from matrix $\psi$ in this way has some unwanted causes. For example, we assign pseudo-labels to all unlabeled samples; however, we are clearly not confident about the same certainty for all generated pseudo-labels. Moreover, pseudo-labels may not represent the same number of samples for each class, which will affect the performance of the learning process. To handle the former problem, we affiliate a weight representing the certainty of the prediction to each pseudo-label. For this purpose, we consider the entropy $\Upsilon$ to compute the level of uncertainty and provide a weight $\omega_i$ to sample $x_i$ formulated as

$$\omega_i = 1 - \frac{\Upsilon(\hat{\delta}_i)}{\log(g)} \tag{11}$$

where $\Upsilon : \mathbb{R}^g \to \mathbb{R}$ is the entropy function, and the weight $\omega_i$ is normalized in $[0, 1]$ because $\log(g)$ is the maximum possible entropy in $\mathbb{R}^g$ [when all datapoints are equally distributed to the clusters, the maximum entropy for $g$ classes is $H = -\sum_{c=1}^{g} 1/g\log(1/g) = \log(g)$]. $\hat{\delta}_i$ is a $g$-dimensional vector of the $i$th rowwise normalized counterpart of $\delta_i$ with components formulated as

$$\hat{\delta}_{ij} = \frac{\delta_{ij}}{\sum_k \delta_{ik}}. \tag{12}$$

To cope with the issue of the situation when we have different number of samples for each class, we provide weight $\upsilon_j$ to class $j$ that is inversely related to class size, formulated as

$$\upsilon_j = (|S_j| + |E_j|)^{-1} \tag{13}$$

where $|S_j|$ is the number of labeled samples and $|E_j|$ is the number of pseudo-labeled samples in class $j$. To this end, we formulated per-sample and per-class weights. We relate the weighted loss to the labeled and pseudo-labeled samples as follows:

$$\begin{aligned} \xi_w(X, Y_S, \hat{Y}_E; \Lambda) = &\sum_{i=1}^{s} \upsilon_{y_i}\varepsilon_{\text{sup}}(f_\Lambda(x_i), y_i) \\ &+ \sum_{i=s+1}^{n} \omega_i \upsilon_{\hat{y}_i}\varepsilon_{\text{pseu}}(f_\Lambda(x_i), \hat{y}_i). \end{aligned} \tag{14}$$

In fact, (14) is the sum of weighted versions of $\xi_{\text{sup}}$ and $\xi_{\text{pseu}}$ in (2) and (4), respectively. Iscen *et al.* [24] used one CNN model to produce the pseudo-labels and then used these labels to train the same model. On the contrary of this approach, we are using two CNN models in the form of a teacher model and a student model. The teacher model generates the pseudo-labels, which are combined with the labeled samples to train the student model. Therefore, the trained student model is not biased toward the labeled data. To this end, the student and teacher models are trained in parallel, according to (14), in which $\hat{y}_i$ in the student model comes from the teacher model.

In summary, considering the nearest neighbor graph definition in the form of affinity matrix, label propagation, sample

TABLE I
DIFFERENT WATER AND ICE CLASSES

| WMO code | Classes |
|---|---|
| 02 | Open Water/ Leads with Water |
| 01–02 | Brash/Pancake Ice |
| 83 | Young Ice (YI) |
| 86–89 | Level first-year ice FYI |
| 95 | Old/deformed Ice |

and class weights, and label and pseudo-label loss terms, our semisupervised method follows a repetitive procedure. Initially, we randomly initialize all the parameters. We then train the teacher model using the $s$ labeled samples in $X_S$, considering the supervised loss term. We use the trained teacher model to extract descriptors $D$ for the complete training set $X$. We then find the $k$-nearest neighbors of all samples to build the adjacency matrix $\Theta$ and carry out label propagation by computing (9). We then assign pseudo-labels to the unlabeled samples in $X_E$ by considering (10). Subsequently, we train both the teacher and student models for one epoch on the complete training set $X$ using the weighted loss function in (14). This process is repeated for $T'$ epochs.

## IV. Experimental Analysis

### A. SAR-Based Sea Ice Dataset

We have trained our proposed method considering 31 Sentinel-1 images. The images are acquired from the North of Svalbard with 40 m × 40 m pixel resolution. They are preprocessed using the ESA SNAP software by applying thermal noise removal, calibration using the $\sigma_0$ lookup table, and multilooking using a $3 \times 3$ boxcar filter. After converting the intensity images to dB values, they are clipped and scaled linearly in the range [0, 1] considering individual channels. The range in dB for horizontal–horizontal (HH) polarization and HV polarization are [min: $-30$, max: 0] and [min: $-35$, max: $-5$], respectively.

To create a suitable dataset for sea ice classification, we used labeled polygons generated from 31 Sentinel-1 EW scenes from the North of Svalbard. These polygons were carefully labeled manually according to coregistered optical images with as small as possible time gaps. We used these images for training our proposed method. More details can be found in [39]. The dataset consists of five classes, as shown in Table I.

Nonetheless, to perform sea ice classification and create a proper dataset [55] for deep learning, we extracted patches with size equal to $32 \times 32$ pixels, corresponding to a spatial resolution of 1280 m$^2$, from inside the polygons, with a stride of 10 pixels. This dataset can be accessed from the link [55]. It is worth mentioning that we analyzed the effect of different patch sizes in a previous work [9]. We found that the validation results got better by increasing the patch size. However, this improvement comes at the cost of a lower spatial resolution as larger patches cover wider areas of the surface. For instance, a larger patch will be classified as water if the majority of the pixels represent water. This would become a significant issue at ice edges as classification based on larger patches would lead to coarser or nonsmooth edges. Therefore, there is a tradeoff between accuracy and resolution. To compensate for this, in our proposed work, we consider a patch size equal to $32 \times 32$ pixels. We extracted two channel patches consisting of HH and HV intensities. It is also worth mentioning that we also analyzed the effect of different channel composition (HH, HV, and incidence angle) in our previous work [9]. We found that adding the HV channel to the HH gives large improvement. However, the improvement resulting from also adding the incidence angle is quite small. In the current work, we do not include the incidence angle as this also enables more proper comparison with other SSL methods [48], [49]. These reference SSL methods largely apply different DA techniques, and the inclusion of the incidence angle is not feasible because of the DA techniques. Therefore, the patches in our work consist of only HH and HV intensities to maintain consistency. In Table I, we provide ice type codes, following the definitions of the World Meteorological Organization [56] and a brief description of each class. We consider binary sea ice classification. The first class, namely, the *water class*, consists of open water and leads with water, and the various ice types are grouped together as the *ice class*. The total number of patches for water is 9317, and for ice, it is 5433. We provided the dataset online [55]. For now, we are interested in analyzing the performance of DNNs for binary classification. Our consideration based on our experience with sea ice classification is that if DNNs can perform well in the binary classification case, they may also classify multiple sea ice types properly.

For validation, we consider some other Sentinel-1 scenes provided by the Norwegian Meteorological Institute [57] from the Danmarkshavn area on the Northeastern coast of Greenland and extract 1516 water patches and 1324 ice patches, mostly from challenging areas. In the first experiment, we consider the training dataset from the North of Svalbard and split it into labeled $X_S$ and unlabeled $X_E$ samples. In the next experiment, to show the capability of the proposed method in classifying real unlabeled data, we consider 5000 random patches picked from the Norwegian Meteorological Institute dataset as the unlabeled dataset $X_E$ and use all samples in the training set as the labeled dataset $X_S$. We insert a different number of labeled datasets for each class, i.e., 15, 30, 60, 100, 500, and 1000. For the inference results, we apply SAR images from the Norwegian Meteorological Institute dataset [57], which were collected in 2018.

### B. Our Model Configurations

We exploit the same network models for the teacher and student models. Similar to [24], we use the network architecture defined in [46] and shown in Table II. We trained the teacher model for 100 epochs in the first training step. In the second step, we trained the teacher model for 200 epochs based on the label propagation to produce pseudo-labels. These labels are then exploited to train the student model concurrently. The learning rate for the teacher model is 0.0008 for the first step and 0.0001 for the second step. The learning rate for the student model during the second step is 0.002. For DA, we used only rotation in both steps to keep the same physical meaning of all

TABLE II
BASE CNN ARCHITECTURE

| Layer | Hyperparameters |
|---|---|
| Input | $32 \times 32$ patches |
| Convolutional | 128 filters, $3 \times 3$ |
| Convolutional | 128 filters, $3 \times 3$ |
| Convolutional | 128 filters, $3 \times 3$ |
| Pooling | Max-pooling, $3 \times 3$ |
| Dropout | $p = 0.5$ |
| Convolutional | 256 filters, $3 \times 3$ |
| Convolutional | 256 filters ,$3 \times 3$ |
| Convolutional | 256 filters, $3 \times 3$ |
| Pooling | Max-pooling, $3 \times 3$ |
| Dropout | $p = 0.5$ |
| Convolutional | 256 filters, $3 \times 3$ |
| Convolutional | 256 filters ,$3 \times 3$ |
| Convolutional | 256 filters, $3 \times 3$ |
| Pooling | Average pool ($6 \times 6$ to $1 \times 1$) |
| Softmax | Fully connected 128 to 2 |

TABLE III
VALIDATION ACCURACY FOR DIFFERENT AMOUNT OF LABELED DATA AND UNLABELED DATA FROM THE TRAINING DATASET

| | 15 | 30 | 40 | 60 | 100 | 500 | 1000 |
|---|---|---|---|---|---|---|---|
| Fully supervised | 39.60 | 52 | 55.67 | 70.35 | 88.72 | 91.50 | 92.06 |
| semi-GANs [48] | - | - | 71.96 | 88.14 | 89.23 | 90.5 | 90.04 |
| MixMatch [50] | - | - | - | 86.28 | 85.88 | 88.19 | 91.02 |
| LP-SSL [24] | 55.62 | 75.34 | 75.23 | 89.97 | 90.42 | **91.73** | 91.24 |
| TSLP-SSL | **88.03** | **86.96** | **90.87** | **90.47** | **91.21** | 91.07 | **91.94** |

the channels of the SAR data and considered the same values for the hyperparameters as used in the previous studies [24], [58] in all experiments. We run the experiments on a single NVIDIA Quadro RTX 5000 with 16-GB memory. The code is available.[1]

### C. Results and Discussion

We trained our models with a distinct number of labeled data to assess the performance of our proposed method in comparison with four reference methods. For this purpose, we consider both a supervised CNN model and three semisupervised methods, namely, semi-GANs [47], MixMatch [49], and label propagation model (LP-SSL) [24]. In the supervised CNN model, we consider the same CNN architecture that we use for both our teacher and student models. We present the validation results in Table III in terms of accuracy for both our proposed TSLP-SSL method and the reference methods. In the first experiment, we use our training data and split it into labeled, i.e., $X_S$, $Y_S$, and unlabeled datasets, $(X_E$. For the validation, we use the validation data that were mentioned previously (see Section IV-A). As can be seen in Table III, our proposed method outperforms the fully supervised CNN architecture considering 15, 30, 40, 60, and 100 labeled samples. Similarly, our method also outperforms the semisupervised methods semi-GANs [47], MixMatch [49], and LP-SSL [24] considering different number of labeled datasets except in case of 500 labeled samples. For comprehensive analysis, we also consider other performance metrics, namely, average precision, average recall, and average F1-score, for both the classes: water and ice. We present the results in Table IV. As can be seen, we also outperform in most cases considering both the supervised and semisupervised methods. In fact, our method learns more information from the unlabeled data, especially when a very limited number of samples are available. In fact, the student model in our approach has the potential to remedy the problem of overfitting of the teacher model when only few samples are available, and it presents comparable validation accuracy when considering 500 and 1000 labeled datasets. However, when the number of labeled datasets increases, the amount of information extracted from the unlabeled data does not significantly improve the results. It is worth noticing that the *good samples* of the labeled data can significantly impact the results in the second step. This can be seen when comparing the results of using 15 and 30 labeled samples in Tables III and IV.

In fact, our proposed method can learn from the unlabeled data and, thus, improves its performance. It even achieves better validation accuracy than the supervised and LP-SSL models considering 15, 30, 40, 60, and 100 labeled samples. In order to explain the behavior of our method considering 500 and 1000 labeled samples, we compute the accuracy of the pseudo-labels from the teacher model during the second step of the training process. This can be done since the ground-truth labels of the unlabeled data can be extracted from the training dataset. We consider the comparison of our proposed method with the fully supervised CNN architecture. When both the methods are trained on 500 and 1000 labeled datasets, the accuracy on the pseudo-labels reaches more than 99%, but at the same time, the validation accuracy does not increase, as shown in Table III. This means that there is no more information in the unlabeled data to further improve the validation accuracy considering this particular dataset. We investigate this by training the supervised model with all the data in the training dataset, and it reached a validation accuracy of 91.57%.

We also investigated the inference results on a single-image SAR scene from Danmarkshavn considering 30, 60, and 100 labeled datasets in our proposed TSLP-SSL model. The results of this experiment are reported in Fig. 3, where the first row shows results using the supervised model and the second row shows results using our proposed method. Blue color indicates the water and white color indicates the ice class. As can be seen, our method presents improvement compared to the supervised model, especially in the noisy areas.

### D. Feature Separability of Our Proposed Method

Furthermore, we illustrate the capability of the label propagation step that we use to generate the pseudo-labels for training the student model. In fact, label propagation is characterized by consolidated feature separability, which helps generate meaningful pseudo-labels for training the student model. To explain this visually, we extract the feature vector output from the last convolution layer. The dimension of the feature vectors is 128. We transform the feature vectors into three components based on the principal component analysis (PCA), considering both labeled and unlabeled data, to visually understand the feature space. These components are shown in Fig. 4. Fig. 4(a) and (c) shows the feature space when training the teacher model in the

[1] https://github.com/sakh251/TSLP-SSL

TABLE IV
AVERAGE OF PRECISION, RECALL, AND F1-SCORE FOR DIFFERENT AMOUNT OF LABELED DATA AND UNLABELED DATA FROM THE TRAINING DATASET

| | 15 | | | 30 | | | 40 | | | 60 | | | 100 | | | 500 | | | 1000 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pre. | Rec. | F1 | Pre. | Rec. | F1 | Pre. | Rec. | F1 | Pre. | Rec. | F1 | Pre. | Rec. | F1 | Pre. | Rec. | F1 | Pre. | Rec. | F1 |
| Fully supervised | .5623 | .5981 | .3645 | .6064 | .7100 | .4948 | .5991 | .7016 | .5123 | .6259 | .7452 | .6163 | .7812 | .6977 | .7283 | .8860 | .8036 | .8336 | .8955 | .8154 | .8480 |
| semi-GANs [48] | - | - | - | - | - | - | .4278 | .4295 | .4286 | .7951 | .6709 | .7086 | .7914 | .7192 | .7474 | .8957 | .7602 | .8059 | .8506 | .7825 | .8103 |
| MixMatch [50] | - | - | - | - | - | - | - | - | - | **.9160** | .5263 | .5041 | .9137 | .5108 | .4740 | .8804 | .6848 | .7319 | .9089 | .7665 | .8143 |
| LP-SSL [24] | .5963 | .6922 | .4921 | .6392 | **.7474** | .6493 | .6611 | **.8038** | .6674 | .7971 | **.7738** | .7847 | .8041 | .7949 | .7990 | **.9154** | .7866 | .8324 | **.9074** | .7751 | .8211 |
| TSLP-SSL | **.7591** | **.7927** | **.7741** | **.7314** | .7345 | **.7329** | **.8299** | .7640 | **.7914** | .8599 | .7546 | **.7925** | **.8452** | .7609 | **.7990** | .8674 | **.8070** | **.8326** | .9007 | **.8062** | **.8432** |

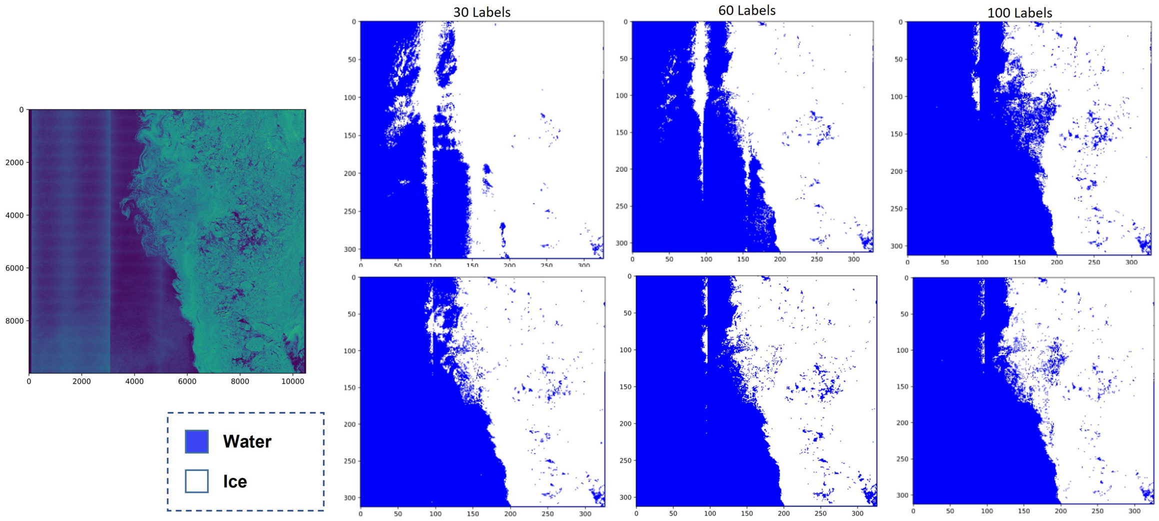


Fig. 3. Inference results. We present qualitative results of a single input image. The first row depicts the results considering supervised deep learning, and the second row depicts the results using our proposed TSLP-SSL model.

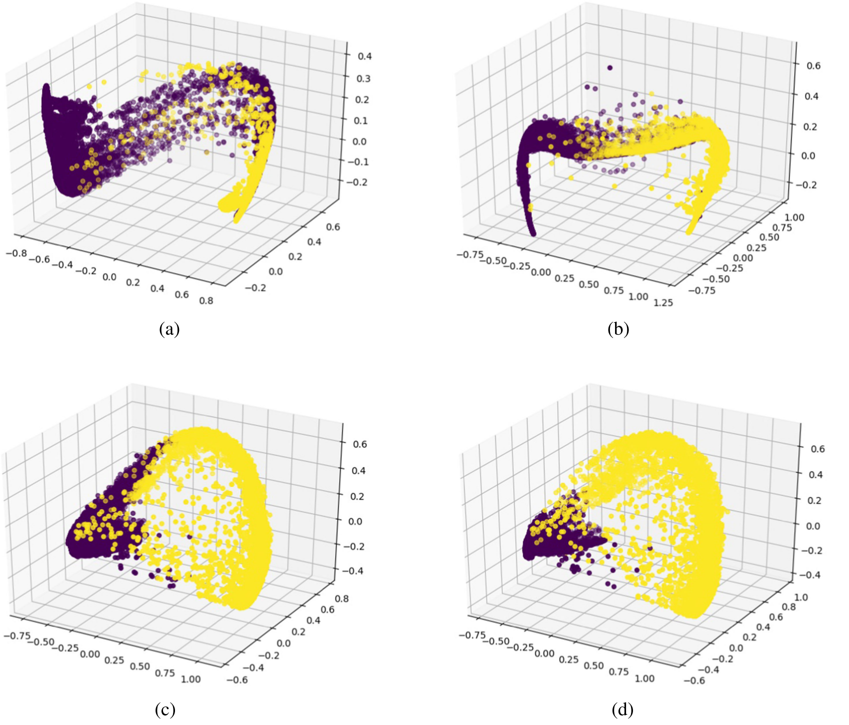

(a)
(b)
(c)
(d)

Fig. 4. Three PCA components' visualization of extracted features (flattened vector after convolution layers with 128 values) from labeled and unlabeled data. The yellow color represents water and the purple color represents ice. (a) and (c) show the supervised feature space from first step with 60 and 1000 labeled data, respectively. (b) and (d) show the best feature space of second step with 60 and 1000 labeled data, respectively.

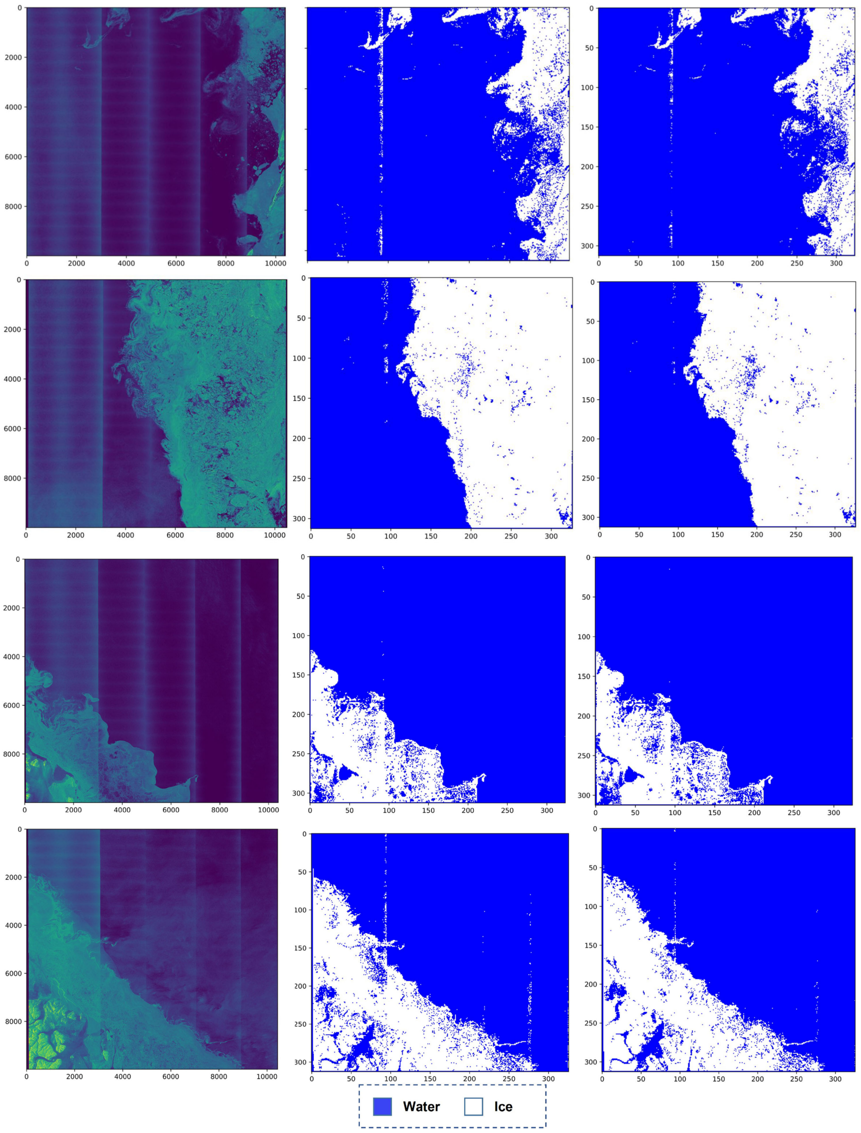


Fig. 5. Inference results. The first column shows input images, the second column shows the results obtained with supervised deep learning, and the third row shows results obtained with our TSLP-SSL model, which is trained by also taking into account unlabeled data from other images.

first step considering 60 and 1000 labeled samples, respectively. Fig. 4(b) and (d) shows the feature space representation after label propagation is applied in the second step. The yellow circles represent water and the purple circles represent the ice class. As can be seen, label propagation leads to more separable classes in the feature space, especially when 1000 labeled samples are considered. Therefore, through label propagation, the unlabeled data help to build a more class-separable feature space and generate more meaningful and informative pseudo-labels to train the student model.

TABLE V

Validation Accuracy, Average Precision, Average Recall, and Average F1-Score Considering Additional Real Unlabeled Data

| | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| Fully supervised | 92.06 | 0.8955 | 0.8154 | 0.8480 |
| semi-GANs [48] | 89.22 | 0.8730 | 0.7264 | 0.7716 |
| MixMatch [50] | 89.55 | 0.8788 | 0.7345 | 0.7800 |
| LP-SSL [24] | 91.19 | 0.9243 | 0.7626 | 0.8144 |
| TSLP-SSL | **92.93** | **0.9291** | **0.8182** | **0.8606** |

### *E. Extended Unlabeled Data*

To elaborate a bit more on the capability of our proposed method, we conduct another experiment. We evaluate the validation accuracy of the proposed method by considering 1000 data samples from the training dataset as labeled data (i.e., considering it as an element of $X_S$) and adding unlabeled data not contained in the training dataset. For this purpose, we extract 5000 random patches from the Danmarkshavn data and add to the training process in the second step $X_E$. We present the performance of all the methods in Table V in terms of accuracy, average precision, average recall, and average F1-scores. As can be seen, our method performs better than the fully supervised CNN method and three semisupervised methods: semi-GANs [47], MixMatch [49], and LP-SSL [24]. These results demonstrate that our proposed method can extract and use relevant information from real unlabeled data and learn new information from unseen and unlabeled data. This is a useful and powerful capability that can be beneficial in sea ice classification, where the amount of available training data is limited.

We also present inference results using four different images from the Danmarkshavn data considering the student model trained on 1000 labeled datasets and extended with unlabeled data. In Fig. 5, the left column depicts the original SAR images, the middle column presents the inference results obtained with the supervised learning model, and the last column shows the results obtained with our proposed TSLP-SSL method. Water is highlighted in blue color and ice is highlighted in white color. These inference results again show the capability of our proposed semisupervised method in using the information of unlabeled data.

## V. Conclusion

In this article, we proposed a teacher–student-based label propagation method for sea ice classification. The teacher model and the student model were trained in an iterative way during the training stage. The teacher model produced features that were extracted before the final classification layer. These features were used during the label propagation process. Considering the unlabeled data, the labels were propagated to produce pseudo-labels. Subsequently, the pseudo-labels from the teacher models were fed to the student model during the training to find an unbiased decision boundary. Our method outperformed the supervised CNN and the semisupervised LP-SSL models. We presented both qualitative and quantitative results for our proposed method and the reference methods. Our proposed method outperformed both the reference methods. Our proposed method considered a very limited number of labeled samples starting from 15 samples and unlabeled samples to train the models efficiently. In fact, our proposed method was characterized by the ability to learn useful information from both labeled and unlabeled data. Our method reduced the dependence on labeled samples, which is very time consuming and costly to collect for sea ice analysis. Therefore, this property of our method makes it a good fit for the community of sea ice analysis, where limited labeled data are available. We have also shown that by adding more unlabeled samples, the performance of the inference results has improved. Considering the semisupervised aspect, our method can be extended to other problem areas, where a very limited number of labeled samples are available since we coped with the biasing and dependence issues related to the labeled samples.

The dataset we collected consists of different ice types. However, the number of samples for each ice type is limited. Considering the promising performance of our proposed method for binary sea ice classification, in our future work, we would adopt and extend our method to ice type classification.

## Acknowledgment

This work is funded in part by Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA) and the Research Council of Norway (RCN Grant no. 237906), the European Union's Horizon 2020 research and innovation programme ExtremeEarth project, grant agreement no. 825258 (http://earthanalytics.eu/) and by the Fram Center under the Automised Large-scale Sea Ice Mapping (ALSIM) "Polhavet" flagship project.

## References

[1] L. P. Bobylev and M. W. Miles, "Sea ice in the Arctic paleoenvironments," in *Sea Ice in the Arctic*. Berlin, Germany: Springer, 2020, pp. 9–56.

[2] T. Vihma, "Effects of Arctic Sea ice decline on weather and climate: A review," *Surv. Geophys.*, vol. 35, no. 5, pp. 1175–1214, 2014.

[3] M. R. Najafi, F. W. Zwiers, and N. P. Gillett, "Attribution of Arctic temperature change to greenhouse-gas and aerosol influences," *Nat. Climate Change*, vol. 5, no. 3, pp. 246–249, 2015.

[4] J. C. Stroeve, M. C. Serreze, M. M. Holland, J. E. Kay, J. Malanik, and A. P. Barrett, "The Arctic's rapidly shrinking sea ice cover: A research synthesis," *Climatic Change*, vol. 110, no. 3, pp. 1005–1027, Feb. 2012.

[5] S. Haykin, E. O. Lewis, R. K. Raney, and J. R. Rossiter, *Remote Sensing of Sea Ice and Icebergs*, vol. 13. Hoboken, NJ, USA: Wiley, 1994.

[6] A. Cristea, J. van Houtte, and A. P. Doulgeris, "Integrating incidence angle dependencies into the clustering-based segmentation of SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2925–2939, 2020.

[7] M. Ghanbari, D. A. Clausi, L. Xu, and M. Jiang, "Contextual classification of sea-ice types using compact polarimetric SAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7476–7491, Oct. 2019.

[8] J. L. Awange and J. B. K. Kiema, "Microwave remote sensing," in *Environmental Geoinformatics*. Berlin, Germany: Springer, 2013, pp. 133–144.

[9] S. Khaleghian, H. Ullah, T. Kræmer, N. Hughes, T. Eltoft, and A. Marinoni, "Sea ice classification of SAR imagery based on convolution neural networks," *Remote Sens.*, vol. 13, no. 9, 2021, Art. no. 1734.

[10] L. Wang, K. Scott, and D. Clausi, "Sea ice concentration estimation during freeze-up from SAR imagery using a convolutional neural network," *Remote Sens.*, vol. 9, no. 5, 2017, Art. no. 408.

[11] W. Wen, J. Timmermans, Q. Chen, and P. M. van Bodegom, "A review of remote sensing challenges for food security with respect to salinity and drought threats," *Remote Sens.*, vol. 13, no. 1, 2021, Art. no. 6.

[12] D. H. Svendsen, M. Piles, J. Muñoz-Marí, D. Luengo, L. Martino, and G. Camps-Valls, "Integrating domain knowledge in data-driven Earth observation with process convolutions," *IEEE Trans. Geosci. Remote Sens.*, 2021.

[13] L. Hashemi-Beni and A. A. Gebrehiwot, "Flood extent mapping: An integrated method using deep learning and region growing using UAV optical data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2127–2135, Jan. 2021.

[14] N. Asadi, K. A. Scott, A. S. Komarov, M. Buehner, and D. A. Clausi, "Evaluation of a neural network with uncertainty for detection of ice and water in SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 247–259, Jan. 2021.

[15] J. Lohse, A. P. Doulgeris, and W. Dierking, "An optimal decision-tree design strategy and its application to sea ice classification from SAR imagery," *Remote Sens.*, vol. 11, no. 13, 2019, Art. no. 1574.

[16] J. Lohse, A. Doulgeris, and W. Dierking, "Mapping sea-ice types from Sentinel-1 considering surface-type dependent effect of incidence angle," *Ann. Glaciol.*, vol. 61, no. 83, pp. 260–270, 2020.

[17] D. Malmgren-Hansen *et al.*, "A convolutional neural network architecture for Sentinel-1 and AMSR2 data fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 1890–1902, Mar. 2021.

[18] G.-J. Qi and J. Luo, "Small data challenges in Big Data Era: A survey of recent progress on unsupervised and semi-supervised methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2020.3031898.

[19] E. Dalsasso, L. Denis, and F. Tupin, "SAR2SAR: A semi-supervised despeckling algorithm for SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4321–4329, 2021.

[20] X. Jiang, G. Li, X.-P. Zhang, and Y. He, "A semisupervised Siamese network for efficient change detection in heterogeneous remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3061686.

[21] Y. Ding, X. Zhao, Z. Zhang, W. Cai, N. Yang, and Y. Zhan, "Semi-supervised locality preserving dense graph neural network with ARMA filters and context-aware learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3100578.

[22] M. Douze, A. Szlam, B. Hariharan, and H. Jégou, "Low-shot learning with large-scale diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3349–3358.

[23] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 321–328.

[24] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5065–5074.

[25] N. Zakhvatkina, V. Smirnov, and I. Bychkova, "Sea ice classification based on neural networks method using Sentinel-1 data," *Int. Multidisciplinary Sci. GeoConf.*, vol. 19, no. 2.2, pp. 617–623, 2019.

[26] N. Zakhvatkina, V. Smirnov, and I. Bychkova, "Satellite SAR data-based sea ice classification: An overview," *Geosciences*, vol. 9, no. 4, 2019, Art. no. 152.

[27] J. Karvonen, "Baltic sea ice concentration estimation using SENTINEL-1 SAR and AMSR2 microwave radiometer data," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2871–2883, May 2017.

[28] F. Dinessen, "Operational multisensor sea ice concentration algorithm utilizing SENTINEL-1 and AMSR2 data," in *Proc. 19th EGU General Assembly*, 2017, Art. no. 19037.

[29] A. M. Johansson, C. Brekke, G. Spreen, and J. A. King, "X-, C-, and L-band SAR signatures of newly formed sea ice in Arctic leads during winter and spring," *Remote Sens. Environ.*, vol. 204, pp. 162–180, 2018.

[30] A. S. Fors, C. Brekke, A. P. Doulgeris, T. Eltoft, A. H. Renner, and S. Gerland, "Late-summer sea ice segmentation with multi-polarisation SAR features in C and X band," *Cryosphere*, vol. 10, no. 1, pp. 401–415, 2016.

[31] M. Dabboor, B. Montpetit, and S. Howell, "Assessment of the high resolution SAR mode of the RADARSAT constellation mission for first year ice and multiyear ice characterization," *Remote Sens.*, vol. 10, no. 4, 2018, Art. no. 594.

[32] D.-B. Hong and C.-S. Yang, "Automatic discrimination approach of sea ice in the arctic ocean using SENTINEL-1 extra wide swath dual-polarized SAR data," *Int. J. Remote Sens.*, vol. 39, no. 13, pp. 4469–4483, 2018.

[33] A. M. Johansson *et al.*, "Consistent ice and open water classification combining historical synthetic aperture radar satellite images from ERS-1/2, ENVISAT ASAR, RADARSAT-2 and sentinel-1A/B," *Ann. Glaciology*, vol. 61, no. 82, pp. 40–50, 2020.

[34] W. Aldenhoff, C. Heuzé, and L. E. Eriksson, "Comparison of ice/water classification in Fram Strait from C-and L-band SAR imagery," *Ann. Glaciol.*, vol. 59, no. 76pt2, pp. 112–123, 2018.

[35] L. Wang, K. A. Scott, L. Xu, and D. A. Clausi, "Sea ice concentration estimation during melt from dual-pol SAR scenes using deep convolutional neural networks: A case study," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4524–4533, Aug. 2016.

[36] Y. Gao, F. Gao, J. Dong, and S. Wang, "Transferred deep learning for sea ice change detection from synthetic-aperture radar images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 10, pp. 1655–1659, Oct. 2019.

[37] M. Tom, R. Aguilar, P. Imhof, S. Leinss, E. Baltsavias, and K. Schindler, "Lake ice detection from SENTINEL-1 SAR with deep learning," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 409–416, 2020.

[38] H. Boulze, A. Korosov, and J. Brajard, "Classification of sea ice types in SENTINEL-1 SAR data using convolutional neural networks," *Remote Sens.*, vol. 12, no. 13, 2020, Art. no. 2165.

[39] J. Lohse, A. P. Doulgeris, and W. Dierking, "Mapping sea-ice types from SENTINEL-1 considering the surface-type dependent effect of incidence angle," *Ann. Glaciol.*, vol. 61, no. 83, pp. 260–270, 2020.

[40] M. Shin, "Semi-supervised learning with a teacher-student network for generalized attribute prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 509–525.

[41] Q. She, J. Zou, M. Meng, Y. Fan, and Z. Luo, "Balanced graph-based regularized semi-supervised extreme learning machine for EEG classification," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 4, pp. 903–916, 2021.

[42] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[43] K. Yu, H. Ma, T. R. Lin, and X. Li, "A consistency regularization based semi-supervised learning approach for intelligent fault diagnosis of rolling bearing," *Measurement*, vol. 165, 2020, Art. no. 107987.

[44] J. Gordon and J. M. Hernández-Lobato, "Combining deep generative and discriminative models for Bayesian semi-supervised learning," *Pattern Recognit.*, vol. 100, 2020, Art. no. 107156.

[45] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 687–10 698.

[46] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[47] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[48] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[49] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060.

[50] Y. Han *et al.*, "A cooperative framework based on active and semi-supervised learning for sea ice classification using EO-1 hyperion data," *Trans. Japan Soc. Aeronaut. Space Sci.*, vol. 62, no. 6, pp. 318–330, 2019.

[51] F. Staccone, "Deep learning for sea-ice classification on synthetic aperture radar (SAR) images in Earth observation: Classification using semi-supervised generative adversarial networks on partially labeled data," master's thesis, School Elect. Eng. Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2020.

[52] F. Li, D. A. Clausi, L. Wang, and L. Xu, "A semi-supervised approach for ice-water classification using dual-polarization SAR satellite imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 28–35.

[53] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowl.-Based Syst.*, vol. 215, 2021, Art. no. 106771.

[54] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4334–4343.

[55] S. Khaleghian, J. P. Lohse, and T. Kræmer, "Synthetic-aperture radar (SAR) based ice types/ice edge dataset for deep learning analysis," 2020. [Online]. Available: https://doi.org/10.18710/QAYI4O

[56] J. Falkingham and V. Smolyanitsky, "Electronic chart systems ice objects catalogue," Version 5.1. draft for approval. Feb. 2012. [Online]. Available: http://hdl.handle. net/11329/403

[57] N. Hughes, "Extremeearth polar use case training data," 2020. [Online]. Available: https://zenodo.org/record/3695276#.X-ytf2j0mUn

[58] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2077–2086.

**Salman Khaleghian** received the bachelor's degree in applied mathematics from Shahed University, Tehran, Iran, in 2006, and the M.S. degree in computer software engineering from Science and Research branch of Azad University, Tehran, Iran, in 2010. He is currently working toward the Ph.D. degree in scalable computing for Earth observation with the Center for Integrated Remote Sensing and Forecasting for Arctic Operations, Faculty of Science and Technology, University of Tromsø—The Arctic University of Norway, Tromsø, Norway, and the SIRIUS Lab, Department of Informatics, University of Oslo, Oslo, Norway.

His research interests include machine learning, deep learning, scalable deep learning, and computer vision.

**Habib Ullah** received the M.S. degree in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2009, and the Ph.D. degree in information and communication technology from the University of Trento, Trento, Italy, in 2015.

From 2015 to 2016, he was an Assistant Professor of Electrical Engineering with COMSATS University, Islamabad, Pakistan. From 2016 to 2020, he was an Assistant Professor with the College of Computer Science and Engineering, University of Ha'il, Ha'il, Saudi Arabia. In 2020, he was a Postdoctoral Researcher with the University of Tromsø—The Arctic University of Norway, Tromsø, Norway. He is currently an Associate Professor with the Norwegian University of Life Sciences, Ås, Norway. His research interests include computer vision and machine learning.

**Thomas Kræmer** received the M.Sc. degree in data analysis and sensor technology in 2011 from the University of Tromsø—the Arctic University of Norway, Tromsø, Norway, where he is currently working toward the Ph.D. degree.

Since 2016, he has been the Head Engineer with the Earth Observation Laboratory, University of Tromsø—The Arctic University of Norway. His research interests include algorithms for automated analysis of synthetic aperture radar images for sea ice applications.

**Torbjørn Eltoft** (Member, IEEE) received the M.Sc. and Ph.D. degrees in physics from University of Tromsø, Norway, in 1981 and 1984, respectively.

In 1988, he joined the Faculty of Science and Technology, University of Tromsø (UiT)—The Arctic University of Norway, Tromsø, Norway, where he is currently a Professor of Remote Sensing with the Department of Physics and Technology. He is also the Director of the Centre for Integrated Remote Sensing and Forecasting for Arctic Operations. He has a significant publication record. His research interests include signal and image analysis, statistical modeling, and machine learning with applications in synthetic aperture radar and ocean color remote sensing.

Dr. Eltoft was the co-recipient of the 2000 Outstanding Paper Award in Neural Networks awarded by the IEEE Neural Networks Council and an honorable mention for the 2003 *Pattern Recognition* Journal Best Paper Award. He was the recipient of the UiT Award for Research and Development in 2017. He was an Associate Editor for *Pattern Recognition* from 2005 to 2011 and a Guest Editor for *Remote Sensing* on the Special Issue for the PolInSAR 2017 Conference.

**Andrea Marinoni** (Senior Member, IEEE) received the B.S., M.Sc. (*cum laude*), and Ph.D. degrees in electronic engineering from the University of Pavia, Pavia, Italy, in 2005, 2007, and 2011, respectively.

He is currently an Associate Professor with the Earth Observation Group, Centre for Integrated Remote Sensing and Forecasting for Arctic Operations, Department of Physics and Technology, University of Tromsø—The Arctic University of Norway, Tromsø, Norway, and a Visiting Academic Fellow with the Department of Engineering, University of Cambridge, Cambridge, U.K. From 2015 to 2017, he was a Visiting Researcher with the Earth and Planetary Image Facility, Ben-Gurion University of the Negev, Be'er Sheva, Israel; the School of Geography and Planning, Sun Yat-Sen University, Guangzhou, China; the School of Computer Science, Fudan University, Shanghai, China; the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China; and the Instituto de Telecomunicações, Instituto Superior Tecnico, Universidade de Lisboa, Lisbon, Portugal. From 2013 to 2018, he was a Research Fellow with the Telecommunications and Remote Sensing Laboratory, Department of Electrical, Computer and Biomedical Engineering, University of Pavia. In 2009, he was a Visiting Researcher with the Communications Systems Laboratory, Department of Electrical Engineering, University of California, Los Angeles, Los Angeles, CA, USA. In 2020 and 2021, he was a Visiting Professor with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia. His research interests include efficient information extraction from multimodal remote sensing, nonlinear signal processing applied to large-scale heterogeneous records, Earth observation interpretation and Big Data mining, and analysis and management for human–environment interaction assessment.

Dr. Marinoni was the recipient of the two-year "Applied Research Grant," sponsored by the Region of Lombardy, Italy, and STMicroelectronics N.V., in 2011; the INROAD Grant, sponsored by the University of Pavia and Fondazione Cariplo, Italy, for supporting excellence in design of ERC proposal in 2017; the "Progetto professionalitá Ivano Becchi" grant funded by the Fondazione Banco del Monte di Lombardia, Italy, and sponsored by University of Pavia and NASA Jet Propulsion Laboratory, Pasadena, CA, for supporting the development of advanced methods of air pollution analysis by remote sensing data investigation, in 2018; and the Åsgard Research Program and Åsgard Recherche+ Program grants funded by the Institut Français de Norvège, Oslo, Norway, in 2019 and 2020, respectively, for supporting the development of scientific collaborations between French and Norwegian Research Institutes. He is the Founder and the Chair of the IEEE Geoscience and Remote Sensing Society (GRSS) Norway Chapter. He is also an Ambassador of the IEEE Region 8 Humanitarian activities, and a research contact point for the Norwegian Artificial Intelligence Research Consortium. He is a topical Associate Editor of machine learning for IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He was a Guest Editor for three special issues on multimodal remote sensing and sustainable development of IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. He is the Leader of the GR4S Committee of the IEEE GRSS, coordinating the organization of schools and workshops sponsored by the IEEE GRSS worldwide.

# 8

# Paper 3: AFSD- Adaptive Feature Space Distillation for Distributed Deep Learning

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 July 2022, accepted 29 July 2022, date of publication 8 August 2022, date of current version 16 August 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3197646*

RESEARCH ARTICLE

# AFSD: Adaptive Feature Space Distillation for Distributed Deep Learning

**SALMAN KHALEGHIAN[1], HABIB ULLAH[2], EINAR BROCH JOHNSEN[3], ANDERS ANDERSEN[1], AND ANDREA MARINONI[1], (Senior Member, IEEE)**

[1]Faculty of Science and Technology, UiT The Arctic University of Norway, 9019 Tromsø, Norway
[2]Faculty of Science and Technology, Norwegian University of Life Sciences (NMBU), 1430 Ås, Norway
[3]Department of Informatics, University of Oslo, 0315 Oslo, Norway

Corresponding author: Salman Khaleghian (salman.khaleghian@uit.no)

This work was supported in part by the Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA) and the Research Council of Norway (RCN) under Grant 237906, in part by the European Union's Horizon 2020 Research and Innovation Programme Extreme Earth Project under Agreement 825258, and in part by the Fram Center under the Automised Large-Scale Sea Ice Mapping (ALSIM) through the ''Polhavet'' Flagship Project.

**ABSTRACT** We propose a novel and adaptive feature space distillation method (AFSD) to reduce the communication overhead among distributed computers. The proposed method improves the Codistillation process by supporting longer update interval rates. AFSD performs knowledge distillates across the models infrequently and provides flexibility to the models in terms of exploring diverse variations in the training process. We perform knowledge distillation in terms of sharing the feature space instead of output only. Therefore, we also propose a new loss function for the Codistillation technique in AFSD. Using the feature space leads to more efficient knowledge transfer between models with a longer update interval rates. In our method, the models can achieve the same accuracy as Allreduce and Codistillation with fewer epochs.

**INDEX TERMS** Distributed deep learning, convolutional neural networks, knowledge distillation, codistillation.

## I. INTRODUCTION

To efficiently process big data, new deep learning based systems have been proposed. These systems significantly improve the overall performance when considering the big data. Furthermore, these systems scale up the training and inference process of deep learning techniques. To address the need for computational resources, the training process could be distributed across multiple computers connected by a network [1], [2]. Distributed deep learning is the most widely used approach to speed up the neural network training by leveraging the computational resources of multiple devices (e.g., multiple GPUs) [1]. These devices are used to accelerate training by distributing data (data-parallel) across the devices. Each device holds a copy of the model being trained, and the copies are kept synchronized throughout the training process. In one step of a typical implementation, every device computes a gradient using different data samples. The gradients are then averaged across all devices (e.g., via an Allreduce operation). Subsequently, each device locally performs an optimization step using the average gradient [1], [3], [4]. The model parameters on every device are initialized to the same value to keep them synchronized. Increasing the number of devices brings more computational power. However, it also brings a significant communication overhead to synchronize the model parameters across all devices at every step [5], [6].

In the literature, different methods are introduced to reduce the communication overhead. For example, quantizing or compressing gradients before synchronizing them [7], synchronizing periodically rather than on every update [8], and only synchronizing among subsets of devices [9]. These methods reduce the communication overhead per update but they also impact the quality of a model or training time. In this regards, one-step Codistillation [10] method is proposed based on online distillation. The distillation process involves training a single model to match the ensemble output rather than the real data labels [11]. Codistillation makes use of distillation in an online manner to accelerate training by

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

transferring the improved performance of the ensemble to each model [12]. In the codistillation technique, the update interval for synchronizing the stored models is important, because it can affect the communication overhead between the machines. However, by having a longer update interval, the efficiency of knowledge distillation is crucial for knowledge sharing between models. Otherwise, the training time would increase which is against the essence of the Codistillation process.

In this paper, we propose *adaptive feature space distillation* (AFSD). In our proposed method, the models perform knowledge distillation by sharing features instead of output as in regular Codistillation [10]. We achieve this by means of a new loss function for the Codistillation technique, characterized by reduced communication overhead. Our method can achieve the same accuracy with longer update intervals by considering fewer epochs. Our method performs knowledge distillation across the models infrequently, which provides flexibility to the models in terms of learning diverse variations in the data. In short, the main contribution of this paper are:

- A novel and adaptive feature space distillation method characterized by reduced communication overhead.
- A new loss function for knowledge distillation which shares features instead of output in the Codistillation technique.
- We outperform the state-of-the-art methods, Allreduce and Codistillation, by getting the same performance with fewer epochs.

The paper is organized as follows: related work is discussed in Section II and background material in Section III. We present the AFSD method in Section IV. Section V validates the proposed method experimentally. Section VI discuss the conclusions and future work.

## II. RELATED WORK

We can categorize distributed deep learning techniques from two perspectives [1], namely *concurrency in networks* and *concurrency in training*. The first category can be further divided into the two sub-categories: *model parallelism* and *data parallelism*.

### A. CONCURRENCY IN NETWORKS

In this category, we compute the output of the layers or the whole network in concurrent mode for the forward evaluation and backpropagation phases. Model parallelism divides the work according to the neurons in each layer. Different parts of the Deep Neural Network (DNN) are computed on different processors in different machines [1]. For example, Huang *et al.* [13] proposed an approach for training huge DNNs that can not be stored in one GPU. With data parallelism, several replicas of a neural network model are created during training, each on a different worker (processor). The workers process different mini-batches locally at each step using an optimizer. For example, the replicas of the model are synchronized (i.e., either by average gradients or parameters) at every step by communicating either with a centralized parameter server [14], [15] or decentralized using Allreduce [16], [17], [18]. By relaxing the synchronization restrictions and creating an inconsistent model, training workers can read parameters and update gradients asynchronously [19]. Data communication in distributed deep learning can be reduced using methods such as quantization [20], [21], [22] or sparsification [23], [24], [25], [26], [27].

### B. CONCURRENCY IN TRAINING

In this category, concurrency is used in the training stage. Multiple instances of training processes run independently on different machines. Concurrency is also used for ensemble learning. Distributed training of ensembles is a completely parallel process, requiring no communication between the workers [28]. Ensemble learning requires more memory and computational power in the training and inference phases. Therefore, knowledge distillation has been used in a two-step training to transfer knowledge of an ensemble with several networks to a single network [12], [29], [30]. To handle the problem of two-step training, Zhang *et al.* [31] investigated how an ensemble of students can learn collaboratively and teach each other throughout the training process. Kim *et al.* [32] introduced a fusion learning method that trains a robust classifier by integrating feature maps. Park and Kwak [33] used feature-level ensembles for knowledge distillation by transferring the ensemble knowledge between multiple teacher networks. Although these methods can be trained in parallel, their main problem is accuracy when the number of epochs is not taken into account. Codistillation [10] taken advantage of ensemble learning and mutual learning to speed up the training. Codistillation uses a distillation-like loss that penalizes predictions made by one model on a batch of training samples for deviating from the predictions made by other models on the same batch.

Our proposed method falls in the category *concurrency in training*. It includes distilled knowledge between models by directly tuning their feature space.

## III. BACKGROUND

Distributed ensemble learning (DEL) addresses the problem of communication overhead by training multiple instances of models (weights) independently on the same dataset. The overall prediction is the average of the predictions of all the models. DEL requires no communication between the computers [1]. However, ensemble learning increases the cost during the validation stage since the predictions from multiple machines are averaged. Ensemble learning also causes latency [1]. The distillation approach [12] addresses this problem by a two-step processes. In the first step, ensemble learning is performed over several machines, so-called teachers. In the second step, a student model is trained to mimic the teacher models. The student model is then used during the test stage. It reduces the cost of ensemble learning by adding another phase to the training process. Using more machines, distillation increases the training time and

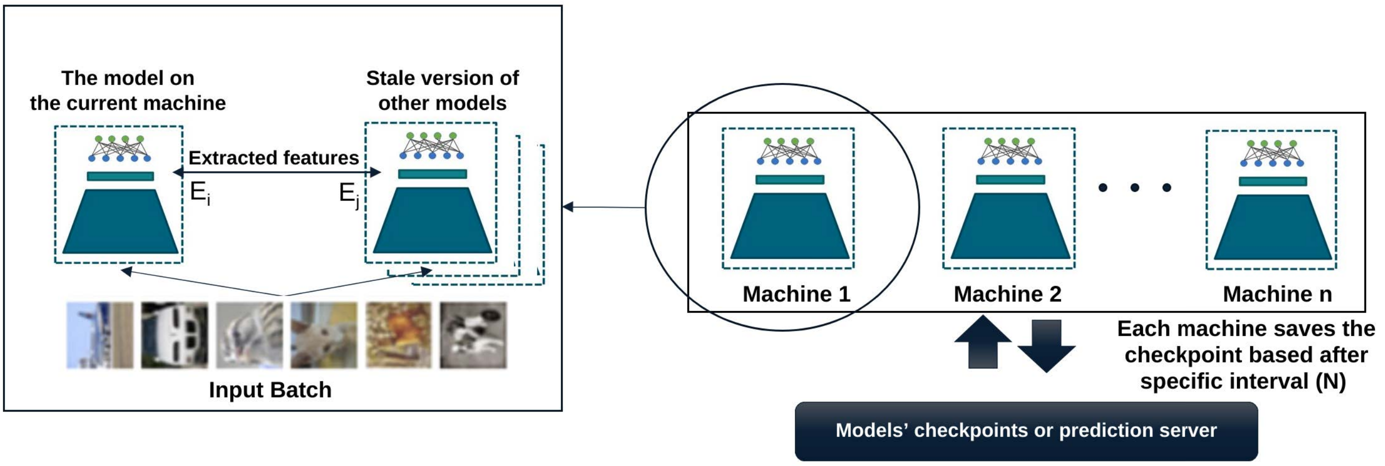

The model on the current machine
Stale version of other models
Extracted features
E_i
E_j
Input Batch
Machine 1
Machine 2
Machine n
Each machine saves the checkpoint based after specific interval (N)
Models' checkpoints or prediction server

**FIGURE 1.** Architecture of our proposed method. The checkpoints of the models are stored on a shared storage or a prediction server after the specific interval (N). Each model is forced to produce the same feature space as the others models for the same inputs batch.

complexity in return for a quality improvement close to a larger teacher ensemble model [10].

However, the ensemble method with distillation remains time consuming. In contrast, one-step Codistillation [10] is based on online distillation and trains $N$ copies of a model in parallel. It starts distillation early in the training process. In the Codistillation technique, the length of the update interval can affect the communication overhead among the machines. For example, the longer the update interval, the lower is the communication overhead. In the ideal case, the communication among different models should be reduced with longer update intervals. Moreover, the update interval affects the diversity between the trained models. The distillation process in Codistillation reduces diversity by forcing the models to predict the same outputs for the same inputs.

## IV. AFSD: ADAPTIVE FEATURE SPACE DISTILLATION

AFSD, the method proposed in this paper, exploits the feature space of each model to explore more variations in the training data instead of using the outputs of the networks to share knowledge between the models. In fact, we tune the models to generate similar feature spaces for transferring knowledge between the models. We consider feature spaces to be similar when the distance between extracted features is the same for the same inputs using different models. To perform knowledge distillation more efficiently with fewer epochs, we manipulate and tune the feature space by considering a distillation term. Our method is based on the Codistillation technique [10] to share knowledge between models rather than synchronizing models to have the same weights. We train $n$ copies of a model in parallel and start distillation early in the training process by adding a new distillation loss term to the loss function. In fact, we have a set of students who simultaneously learn during the training process to handle the classification together (Figure 1). Each model saves the checkpoint of its weights on a shared storage after each update interval, so each model considers the other models as teachers in a distillation-like setup. However, each model uses the stale version of stored models on shared storage (or a prediction server) and performs additional forward passes with the same input batch.

### A. FORMULATION OF THE LEARNING PROCESS

We consider an input batch $X = \{x_1, \ldots x_n\}$ where $x_k$ is an input sample and $n$ is the size of the input batch and labels $Y = \{y_1, y_2, \ldots, y_n\}$ according to the input batch $X$. The output of model $i$ is defined by $f_{\theta i}(x_k)$ where $x_k \in X$ and $\theta_i$ is the parameter of model $i$. The extracted features from model $i$ are represented by $F_{\theta i}(x_k) = a_{ik}$ where $a_{ik}$ are extracted features by considering the $x_k$ sample by the model $i$. The number of models is represented by $m$.

We propose a loss function including a distillation penalty to force the models to produce the same feature space (extracted features before fully connected layers (FC)). For this purpose, the penalty term forces models to have the same distance between extracted features when considering the same inputs. In other words, if the distance between two features for two samples is shorter or longer using one model, the distance between the features for the same two samples extracted from the other models should be similar. For this purpose, we formulate a $n \times n$ distance matrix representing the distance between an individual feature and all other features. We formulate the distance matrix $E_i$ of the model $i$ as

$$E_i = \begin{pmatrix} D(a_{i1}, a_{i1}) & D(a_{i1}, a_{i2}) & \ldots & D(a_{i1}, a_{in}) \\ D(a_{i2}, a_{i1}) & D(a_{i2}, a_{i2}) & \ldots & D(a_{i2}, a_{in}) \\ \ldots & \ldots & \ldots & \ldots \\ D(a_{in}, a_{i1}) & D(a_{in}, a_{i2}) & \ldots & D(a_{in}, a_{in}) \end{pmatrix} \tag{1}$$

where $a_{ik}$ and $a_{ij}$ are extracted features considering samples $x_k$ and $x_j$ using model $i$. Let $D$ be the distance metric and $n$ the input batch size. We formulate the loss function for the model $i$ as

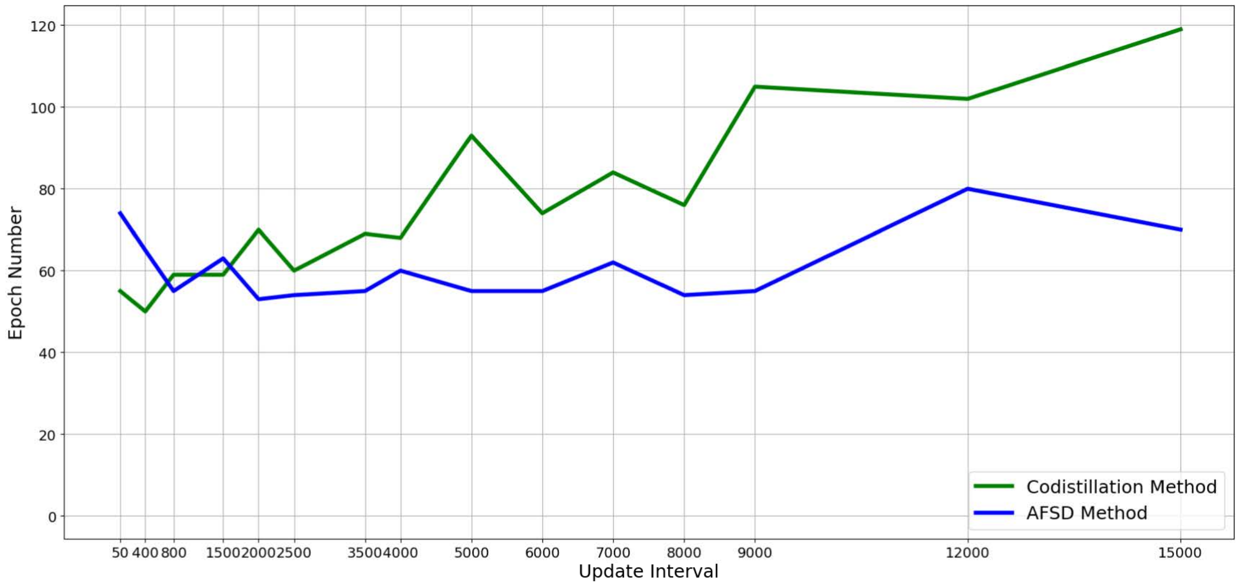


**FIGURE 2.** Comparison between earliest epoch number that achieves the same accuracy as Allreduce considering Codistillation and our proposed method.

follows:

$$loss = L(Y, f_{\theta i}(X)) + \alpha \frac{1}{m-1} \sum g(E_i, E_j)$$
$$i \neq j \quad and \quad j \in 1, 2, 3, \ldots m \qquad (2)$$
$$g(E_i, E_j) = \frac{1}{n} e^T |E_i - E_j| e \qquad (3)$$

where $m$ is the number of models on different machines, X is the input batch, Y is the input labels, $\alpha$ is the penalty coefficient and $L$ represents the loss between prediction and the labels. The function $g$ indicates the average distance between elements of $E_i$ and $E_j$ considering batch size $n$, where $e$ is the column vector whose entries are all 1's and T is the transpose operator. The first term is the cross entropy loss and the second term is the distillation loss. Based on this loss function, we show our proposed method in Algorithm 1.

**Algorithm 1** AFSD Algorithm

```
Initialization of network parameters ($\theta_i, i \in 1, 2, 3, \ldots, m$)
Initialization of learning rate $\mu$ and penalty coefficients $\alpha$ for each model
Repeat for the number of epochs:
Do in parallel for $i \in 1, 2, 3, \ldots, m$ :
Get next batch (X,Y) by size n
Update $\theta$:
    $\theta_i^{k+1} = \theta_i^k + \mu \nabla_{\theta i}(L(Y, f_{\theta i}^k(X)) + \alpha \frac{1}{m-1} \sum g(E_i, E_j))$
    i $\neq$ j   and   j $\in$ 1,2,3, ...m
```

## V. EXPERIMENTAL ANALYSIS

In our experiments, we want to evaluate our method and show that it can address the aforementioned problems. The evaluation is done in terms of four research questions (RQs). RQ1: How does AFSD cope with the impact of the update interval without affecting the performance? RQ2: Does AFSD achieve the same performance with fewer epochs considering a longer update interval? RQ3: How does the new distillation loss term based on the feature space affect the training process and the outputs of the models? RQ4: How does AFSD work when different network architectures are trained? We address these research questions in the subsections V-A, V-B, V-C, respectively.

*Experimental Setup and Design:* In our experiments, we use the standard CIFAR10 dataset [34]. For comparison, we consider the Allreduce [16] and the Codistillation [10] techniques. We consider the Allreduce technique as a baseline for comparison. For the Allreduce technique, we used hyperparameters and the gradual warmup strategy for changing the learning rate [16]. For evaluating our model, we also consider different architectures, namely ResNet20, ResNet32, and VGG16. In case of ResNet20, we trained the network using the Allreduce technique and we achieved a top-1 validation accuracy of 91.71% after 114 epochs. In the Allreduce technique, the input batch is divided

**TABLE 1.** Validation accuracy and earliest epoch that achieves the same accuracy as Allreduce considering Codistillation [10] and AFSD with update interval between 40 and 3500.

| | 50 | | 400 | | 800 | | 1500 | | 2000 | | 2500 | | 3500 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy |
| Codistillation [10] | 55 | 91.78 | 50 | 91.71 | 59 | 91.71 | 59 | 91.79 | 70 | 91.71 | 60 | 91.81 | 69 | 91.73 |
| AFSD | 74 | 91.77 | 65 | 91.74 | 55 | 91.74 | 63 | 91.72 | 53 | 91.74 | 54 | 91.75 | 55 | 91.74 |

**TABLE 2.** Validation accuracy and earliest epoch that achieves the same accuracy as Allreduce considering Codistillation [10] and AFSD with update interval between 4000 and 15000.

| | 4000 | | 5000 | | 6000 | | 7000 | | 8000 | | 9000 | | 12000 | | 15000 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy | Epoch | Accuracy |
| Codistillation [10] | 68 | 91.72 | 93 | 91.76 | 74 | 91.72 | 84 | 91.72 | 76 | 91.72 | 105 | 91.72 | 102 | 91.71 | 119 | 91.76 |
| AFSD | 60 | 91.74 | 55 | 91.96 | 55 | 91.72 | 62 | 91.71 | 54 | 91.79 | 55 | 91.71 | 80 | 91.76 | 70 | 91.73 |

between different GPUs. However, we use the same batch input for training the models based on AFSD. Therefore, we feed the data twice as compared to the Allreduce technique and we expect that AFSD driven by linear scalability should achieve the same accuracy with half the number of epochs. We set the batch size such that each GPU receives 128 samples in each batch in all three methods.

In our experiments, we use two servers with three Nvidia GPUs namely Quadro RTX 5000 with 16GB memory on each. The servers are connected through a point-to-point 10Gb network. We use NFS shared storage to save and restore models checkpoints.

### *A. UPDATE INTERVAL (RQ1 AND RQ2)*

We compare our proposed method with the Codistillation method [10] considering different update intervals. We consider update intervals from 50 steps to 15000 steps. To show the capability of our proposed method, we compare the epoch number on which each method achieves the desired accuracy based on the Allreduce method. The comparison between the Codistillation method and our proposed method is shown in Figure 2. Table 1 shows validation accuracy and earliest epoch that achieves the same accuracy as Allreduce considering Codistillation [10] and AFSD with update interval 40 to 3500 and Table 2 shows same validation accuracy for update interval from 4000 to 15000. In this experiment, we record the epoch number when a specific method reaches the same accuracy as Allreduce. It can be seen, when the update interval is longer, our method achieves the desired accuracy with much fewer epochs. Additionally, our proposed method is driven by linear scalability and tolerates longer update intervals considering 12000 steps. In fact, when using 9000 steps as the update interval, we update the saved checkpoint after 23 epochs (the batch size is 128, and we have 50000 samples in the training dataset). This shows that our method can achieve the same accuracy and scalability with very little communication overhead. However, when we update the models more often with shorter update intervals, we reduce the diversity of the models. When we use more powerful distillation, information sharing does not have the intended benefits. The comparison between Allreduce, Codistillation, and our proposed method using ResNet20 [36] is shown in Figure 3. We consider update intervals equal to 5000 and 9000 steps. We use the early stopping strategy when Codistillation and AFSD achieve 91.71% or more accuracy. We reduce the learning rate on epochs 45 and 55 with a factor 0.1 when Codistillation and AFSD are used. We also consider the first four epochs as the warm-up epochs applicable to the update interval. Otherwise, we let the networks continue the training independently, based on the specified update intervals. As we can see in Figure3, our method can achieve the same accuracy with fewer epochs compared to Allreduce and Codistillation. Since we are using much longer update intervals, the networks are trained more independently in a different direction. Therefore, we can see a rise in training loss when we transfer knowledge based on distillation loss terms. However, the loss reduces through the training process.

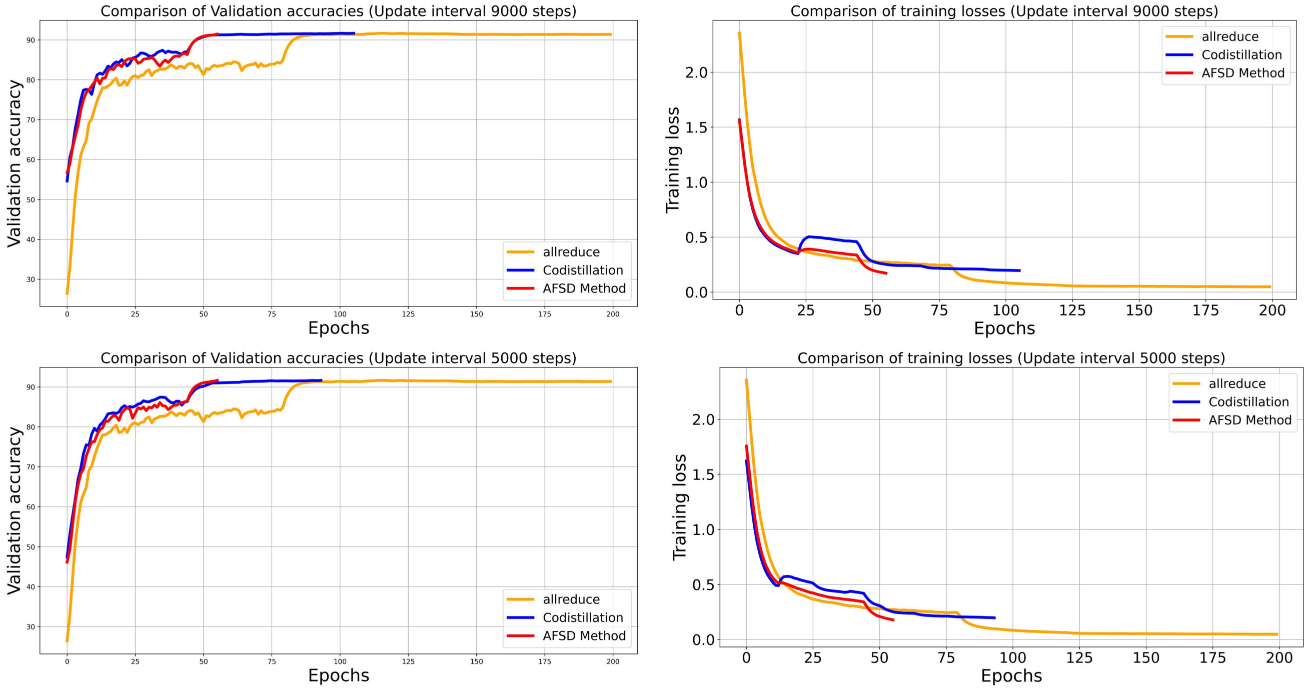


**FIGURE 3.** Validation accuracy and training loss using Codistillation [10], our proposed method, and Allreduce [35]. We use the early stopping strategy when the Codistillation achieves the same accuracy of Allreduce considering 9000 steps (a, b) and 5000 steps (c, d) update intervals.

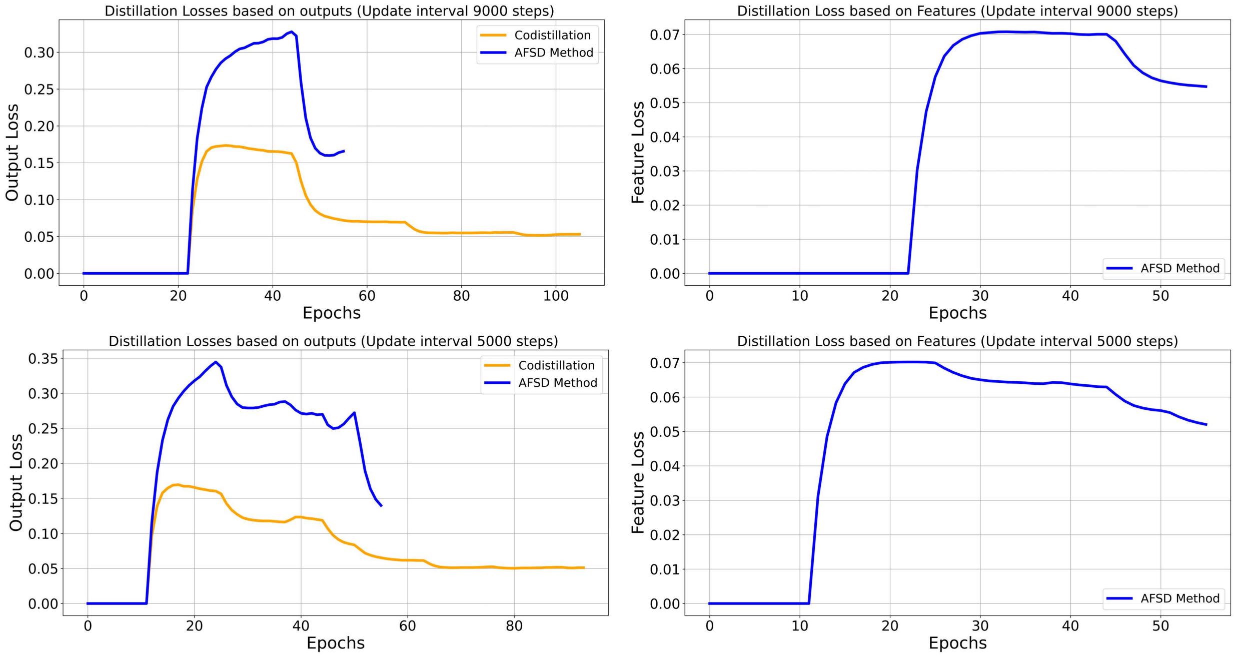


**FIGURE 4.** Distillation losses based on the outputs and features considering Codistillation and our proposed method. We do not include distillation loss based on the outputs in our method, but we measure it to compare with the Codistillation approach.

### B. DISTILLATION LOSS (RQ3)

The loss function is based on two terms: cross-entropy loss and distillation loss. The Codistillation technique uses distillation loss based on the output of the networks. In contrast, our method encodes distillation loss based on the feature space. We want to illustrate the difference between the two loss terms by showing how these behave during the training process for AFSD. Figure4 shows the distillation loss considering the Codistillation method and AFSD. It should be noted, in our method, we do not use output-based distillation, but we measure it during the training. As we can see, in Figures4 (a) and4 (c), distillation based on outputs fluctuates and even increases with AFSD. The networks are robust for classifying the extracted features even with different outputs since we just force them to generate the same features. This is because the neural network models can represent the same function in different ways with different parameter values [37]. However, Figures4 (b) and4 (d) show that the loss between extracted features is reduced through the training, and we can transfer knowledge between the models. Even small changes in the features can lead to more effective distillation, and the networks can achieve the same accuracy with fewer epochs.

### C. NETWORK ARCHITECTURES (RQ4)

In order to evaluate the generalization capability of AFSD regardless of the use of a specific network architecture, we consider other architectures in this section. Figure5 shows the validation accuracy of the ResNet32 network using Codistillation and AFSD for update intervals equal to 5000 and 9000 steps. For this experiment, we consider 92.41% as the top-1 accuracy of Allreduce. The Allreduce operation achieves this accuracy after 123 epochs. We use the learning rate schedule for this network to reduce the learning rate by a factor of 0.1 on the epoch number equal to 50 and 60. As we can see, our method achieves this accuracy with fewer epochs compared to the Codistillation method.

We also explore the VGG-16 [38] model and a 13-layer CNN [39] architecture to consider architectures not belonging to the ResNet families. However, considering both Codistillation and AFSD, using these architectures, we would not get the same accuracy with fewer epochs then needed for the Allreduce technique. Therefore it seems these methods can be more effective with the ResNet family of architectures.

### D. THREATS TO VALIDITY

In our experiments, we used three GPUs on each server. In each server we considered the Allreduce algorithm to train a model in a synchronized manner on these three GPUs. Increasing the number of GPUs on each server could affect the results since it would increase the number of epochs to get the appropriate accuracy. Since this would be same for both Codistillation and AFSD, we consider them significant parameters. Experiments with more GPUs on each server in a two-way setup can be considered.

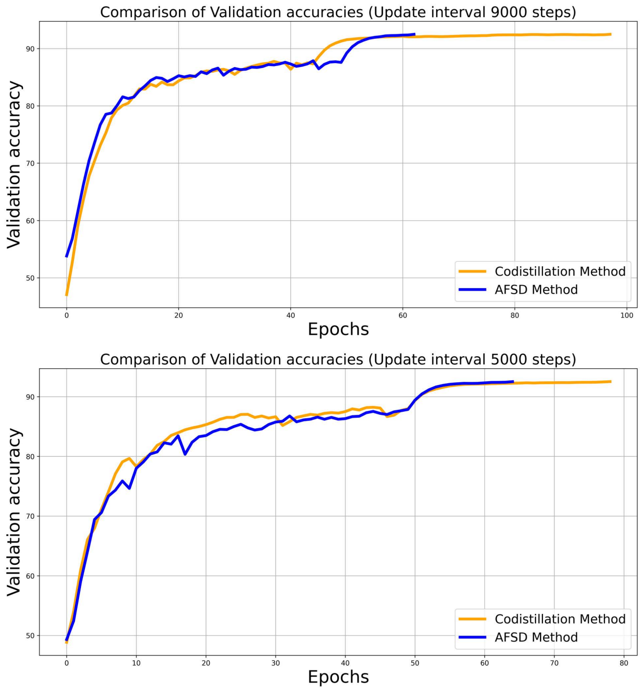


**FIGURE 5. Validation accuracy of the ResNet32 network using Codistillation and our proposed method for the update intervals equal to 5000 and 9000.**

We considered the ResNet20, ResNet30, VGG16 and a 13-layer CNN [39] architectures in our experiments. Deeper architectures with more parameters could exhibit different behaviors. Deeper architectures usually learn features at various levels of abstractions. Therefore, considering only high-level features at the end of the network would not be sufficient to share knowledge. We can also observe this issue when a pure convolution network like VGG16 is used. Hence deeper architectures with more parameters and more intermediate features can be considered for future experiments.

We consider random initialization as a diversity enforcement regularization. However, it can violate a specific situation when the initialized weights or training directions are aligned together. Hence we assume that this diversity can be accomplished by weight randomization.

## VI. CONCLUSION

In this paper, we propose AFSD, a new method for large scale distributed deep learning. The main novelty of AFSD is knowledge sharing based on the feature space of parallel models in the Codistillation setup. Our method supports much longer update intervals using a new knowledge distillation loss function. By prolonging the update interval, the models become more diverse and contribute more to the training process. Additionally, the communication overhead is significantly reduced. We show that with only two updates through the training process, the models can achieve linear scalability using the feature space for sharing the information.

In future work, we will consider our approach of feature space sharing for scalable semi-supervised learning. It has been shown that generating pseudo-labels for unlabeled data using feature spaces increases the performance of deep semi-supervised learning [40], [41]. An extension of our method to semi-supervised learning, could be useful to address the issue of scarce training data, which is critical in many areas where a small number of labeled data and a large amount of unlabeled data are available.

## REFERENCES

[1] T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–43, Jul. 2020.

[2] A. C. Zhou, B. Shen, Y. Xiao, S. Ibrahim, and B. He, "Cost-aware partitioning for efficient large graph processing in geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1707–1723, Jul. 2020.

[3] R. Mayer and H.-A. Jacobsen, "Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–37, Jan. 2021.

[4] M. A. Soyturk, P. Akhtar, E. Tezcan, and D. Unat, "Monitoring collective communication among GPUs," 2021, *arXiv:2110.10401*.

[5] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2016, *arXiv:1604.00981*.

[6] F. Zhang, Z. Chen, C. Zhang, A. C. Zhou, J. Zhai, and X. Du, "An efficient parallel secure machine learning framework on GPUs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 9, pp. 2262–2276, Sep. 2021.

[7] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6c340f25839e6acdc73414517203f5f0-Paper.pdf

[8] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19. [Online]. Available: https://openreview.net/forum?id=S1g2JnRcFX

[9] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 344–353.

[10] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," 2018, *arXiv:1804.03235*.

[11] Z. Allen-Zhu and Y. Li, "Towards understanding ensemble, knowledge distillation and self-distillation in deep learning," 2020, *arXiv:2012.09816*.

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS), Deep Learn. Workshop*, 2014. [Online]. Available: https://openreview.net/forum?id=S1g2JnRcFX

[13] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 103–112.

[14] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. NIPS*, vol. 2, 2014, pp. 1–4.

[15] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, "GeePS: Scalable deep learning on distributed GPUs with a GPU-specialized parameter server," in *Proc. 11th Eur. Conf. Comput. Syst.*, Apr. 2016, pp. 1–16.

[16] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," *CoRR*, vol. abs/1706.02677, pp. 1–12, Jun. 2017.

[17] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," 2017, *arXiv:1705.09056*.

[18] A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, "S-Caffe: Co-designing MPI runtimes and caffe for scalable deep learning on modern GPU clusters," in *Proc. 22nd ACM SIGPLAN Symp. Princ. Pract. Parallel Program.*, 2017, pp. 193–205.

[19] F. Niu, B. Recht, C. Ré, and S. J. Wright, "HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent," 2011, *arXiv:1106.5730*.

[20] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1737–1746.

[21] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds., San Juan, PR, USA, May 2016, pp. 1–14.

[22] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, 2017.

[23] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 440–445. [Online]. Available: https://aclanthology.org/D17-1045

[24] C.-Y. Chen, J. Choi, D. Brand, A. Agrawal, W. Zhang, and K. Gopalakrishnan, "Adacomp: Adaptive residual gradient compression for data-parallel distributed training," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–9.

[25] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, Apr./May 2018, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SkhQHMW0W

[26] C. Renggli, S. Ashkboos, M. Aghagolzadeh, D. Alistarh, and T. Hoefler, "SparCML: High-performance sparse communication for machine learning," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2019, pp. 1–15.

[27] P. D'Ambra and S. Filippone, "A parallel generalized relaxation method for high-performance image segmentation on GPUs," *J. Comput. Appl. Math.*, vol. 293, pp. 35–44, Feb. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037704271500254X

[28] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why m heads are better than one: Training a diverse ensemble of deep networks," 2015, *arXiv:1511.06314*.

[29] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=S1gmrxHFvB

[30] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *Proc. Interspeech*, Aug. 2017, pp. 3697–3701.

[31] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.

[32] J. Kim, M. Hyun, I. Chung, and N. Kwak, "Feature fusion for online mutual knowledge distillation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4619–4625.

[33] S. Park and N. Kwak, "Feature-level ensemble knowledge distillation for aggregating knowledge from multiple networks," in *Proc. ECAI*. Amsterdam, The Netherlands: IOS Press, 2020, pp. 1411–1418.

[34] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, USA, Tech. Rep., 2009.

[35] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, May 2015, pp. 1–14.

[39] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[40] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5070–5079.

[41] S. Khaleghian, H. Ullah, T. Kraemer, T. Eltoft, and A. Marinoni, "Deep semisupervised teacher–student model based on label propagation for sea ice classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10761–10772, 2021.

**SALMAN KHALEGHIAN** received the bachelor's degree in applied mathematics/computer science and the M.S. degree in computer software engineering, in 2006 and 2010, respectively. He is currently pursuing the Ph.D. degree in scalable computing for earth observation with the Center for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA), Faculty of Science and Technology, University of Tromsø (UiT), and the SIRIUS Laboratory, Department of informatics, University of Oslo (UiO). His research interests include machine learning, deep learning, scalable deep learning, and computer vision.

**HABIB ULLAH** received the M.S. degree in electronics and computer engineering from Hanyang University, South Korea, in 2009, and the Ph.D. degree in information and communication technology from the University of Trento, Italy, in 2015. He served as an Assistant Professor in electrical engineering at COMSATS University Islamabad, Pakistan, from 2015 to 2016. He worked as an Assistant Professor at the College of Computer Science and Engineering, University of Ha'il, Saudi Arabia, from 2016 to 2020. He also worked as a Postdoctoral Researcher at the UiT The Arctic University of Norway, in 2020. He is currently working as an Associate Professor with the Norwegian University of Life Sciences (NMBU). His research interests include computer vision and machine learning.

**EINAR BROCH JOHNSEN** is currently a Professor with the Department of Informatics, University of Oslo. He is active in formal methods for distributed and concurrent systems, including object-oriented and actor languages, manycore computing, cloud computing, and digital twins. He is one of the main developers of the ABS Modeling Language.

He is the Strategy Director of SIRIUS, a Center for Research-Driven Innovation on Scalable Data Access, with eight years funding, from the Research Council of Norway. He has been prominently involved in many national and European research projects; in particular, he was the Coordinator of the EU FP7 Project Envisage on formal methods for cloud computing, from 2013 to 2016, and the Scientific Coordinator of the EU H2020 Project HyVar on hybrid variability systems, from 2015 to 2018. His research interests include programming models and methodology, program specification and modeling, formal methods and associated theory, lightweight analysis, type systems, testing, and deductive verification and formal logic.

He is a member of IFIP WG2.2 "Formal Description of Programming Concepts". He was a Board Member of SINTEF ICT, from 2009 to 2015. He is currently a member of the Scientific Council of the Science Centre, UiO, a Board Member of the Formal Methods Europe, and a Steering Committee Member of the conference series on Fundamental Approaches to Software Engineering (FASE), Integrated Formal Methods (iFM), and Formal Techniques for Networked and Distributed Systems (FORTE). He was the General Chair of FM 2015 and DisCoTec 2008, and the PC Chair of FASE 2022, SEFM 2018, TAP 2017, ESOCC 2016, iFM 2013, and FMOODS 2007. He is an Editorial Board Member of the Journals *Formal Aspects of Computing* and *Journal of Logical and Algebraic Methods in Programming.*

**ANDERS ANDERSEN** is the Head of Department at the Department of Computer Science, UiT The Arctic University of Norway. He is Leading a national workgroup for the strengthening of ICT-security in technology studies in Norway, from 2019 to 2020. The workgroup is appointed by the Norwegian Association of Higher Education Institutions (UHR) on behalf of the Ministry of Education and Research.

The research of Anders Andersen covers four main domains. The first domain is security, where the focus is adaptable security, secure storage and sharing of data, security related to mobile systems, NFC and secure elements (e.g. SIM), and analysis of sensitive data (e.g. person-sensitive health data). The second domain is support for mobility and context, where configuration and reconfiguration of systems based on the current context are made possible with adaptable architectures. This domain includes integration of a wide range of services and information sources for the development of personalized and context sensitive solutions. The third domain is support for multimedia applications, including support for continuous media. He has used formal specifications directly for quality of service (QoS) management in running systems and he has developed an explicit binding architecture for multimedia communication. The fourth domain is adaptive system architectures, where he has developed programming abstractions for adaption control and techniques to observe and analyse system behavior.

His research interests include security, mobile services, personalisation, complex distributed applications, and privacy aware analysis of sensitive data.

**ANDREA MARINONI** (Senior Member, IEEE) received the B.S., M.Sc., *(cum laude)* and Ph.D. degrees in electronic engineering from the University of Pavia, Pavia, Italy, in 2005, 2007, and 2011, respectively.

From 2013 to 2018, he has been a Research Fellow at the Telecommunications and Remote Sensing Laboratory, Department of Electrical, Computer, and Biomedical Engineering, University of Pavia. In 2009, he has been a Visiting Researcher at the Communications Systems Laboratory, Department of Electrical Engineering, University of California–Los Angeles (UCLA), Los Angeles, CA, USA. In 2011, he has been the recipient of the two-year ‘‘Applied Research Grant’’, sponsored by the Region of Lombardy, Italy, and STMicroelectronics N.V. In 2017, he has been the recipient of the INROAD grant, sponsored by University of Pavia and Fondazione Cariplo, Italy, for supporting excellence in design of ERC proposal. In 2018, he has been the recipient of the ‘‘Progetto Professionalità Ivano Becchi’’ grant funded by Fondazione Banco del Monte di Lombardia, Italy, and sponsored by the University of Pavia and NASA Jet Propulsion Laboratory, Pasadena, CA, for supporting the development of advanced methods of air pollution analysis by remote sensing data investigation. He has been the recipient of Åsgard Research Programme and Åsgard Recherche+Programme grants funded by the Institut Français de Norvège, Oslo, Norway, in 2019 and 2020, respectively, for supporting the development of scientific collaborations between French and Norwegian research institutes. From 2015 to 2017, he has been a Visiting Researcher at the Earth and Planetary Image Facility, Ben-Gurion University of the Negev, Be'er Sheva, Israel; School of Geography and Planning, Sun Yat-sen University, Guangzhou, China; School of Computer Science, Fudan University, Shanghai, China.; Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China.; Instituto de Telecomunicações, Instituto Superior Tecnico, Universidade de Lisboa, Lisbon, Portugal. In 2020 and 2021, he has been a Visiting Professor at the Department of Electrical, Computer, and Biomedical Engineering, University of Pavia. He is currently an Associate Professor with the Earth Observation Group, Centre for Integrated Remote Sensing and Forecasting for Arctic Operations (CIRFA), Department of Physics and Technology, UiT the Arctic University of Norway, Tromsø, Norway, and a Visiting Academic Fellow with the Department of Engineering, University of Cambridge, U.K. His main research interests include efficient information extraction from multimodal remote sensing, nonlinear signal processing applied to large scale heterogeneous records, earth observation interpretation and big data mining, and analysis and management for human–environment interaction assessment.

He is the Founder and the Current Chair of the IEEE GRSS Norway Chapter. He is also an Ambassador for IEEE Region 8 Humanitarian Activities, and a Research Contact Point for the Norwegian Artificial Intelligence Research Consortium (NORA–nora.ai). He serves as a Topical Associate Editor of machine learning for IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He has been the Guest Editor of three special issues on multimodal remote sensing and sustainable development for IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. He is also the leader of the GR4S committee within IEEE GRSS, coordinating the organization of schools and workshops sponsored by IEEE GRSS worldwide.

# 9 Conclusions and Future Works

We explored the potential of different CNN models for sea ice classification. The results showed that CNN architectures (such as those based on the VGG network) typically obtain promising classification results. We assessed the robustness of the trained CNN models when applied to SAR scenes collected at different spatial locations and times our findings were positive and show that the models have good potential. However, we found that the additive system noise in the SAR imagery is a challenging problem in obtaining refined sea ice maps. Additionally, we emphasized that the scarcity of reliable and balanced sea ice training and validation datasets is a severe problem for training deep neural network architectures.

We proposed a semi-supervised method that considered very limited labeled samples and relatively large unlabeled samples to train the models to address the scarce training data issue. In fact, our method in semi-supervised learning is characterized by the ability to learn practical information from labeled and unlabeled data. Our method reduced the dependency on labeled samples which is very time-consuming and costly to collect for sea ice analysis. However, as mentioned before, Deep Neural Networks training is computationally intensive, especially in semi-supervised learning where we can increase unlabeled data to involve in the training process easier. Additionally, we found selecting the labeled and unlabeled data can significantly affect the performance of the models. Specifically, unlabeled and labeled data should come from the same classes and the same distribution. Otherwise, the methods could not be able to extract useful information from unlabeled data to improve the model.

We proposed a new method for large-scale distributed deep learning analysis to address computation complexity. Our method is a novel approach for knowledge sharing based on the feature space of parallel models. In our method, the communication overhead is significantly reduced and we showed that with only a few communications through the training process, the models can achieve linear scalability. We considered our proposed distributed deep learning method in a supervised manner. However, this method can be used to extend our semi-supervised method for large-scale semi-supervised analysis.

## 9.1 Future works

In our methods, we considered patch-wise classification which degrades the spatial resolution. To cope with this issue one direction could be considering a pixel-wise set-up. However, the pixel-wise set-up will be driven by more computational overhead. Therefore, transforming the current architecture to process the input data quickly in a pixel-wise setup is necessary. For this purpose, one direction can be accomplished by replacing the FC layers with convolution layers based on the work of Sermanet et al. [137].

We considered a supervised version of our proposed DDL. In future work, our adaptive feature space distillation can be used with the junction of label propagation [135, 73] idea for scalable semi-supervised learning. The feature space from different models based on unlabeled data can bring more information.

# Bibliography

[1] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.

[2] V. Sze, Y.-H. Chen, T.-J. Yang, and J. Emer, "Efficient processing of deep neural networks: A tutorial and survey." [Online]. Available: https://arxiv.org/abs/1703.09039

[3] A. Ng. (2019) Deep learning course - stanford cs229. [Online]. Available: http://cs229.stanford.edu/materials/CS229-DeepLearning.pdf

[4] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[5] e. a. Manolis Koubarakis, Konstantina Bereta, "From copernicus big data to extreme earth analytics." in *22nd International Conference on Extending Database Technology (EDBT 2019). Lisbon, Portugal*, pp. 26–29, year=2019.

[6] M. A. Wulder, J. G. Masek, W. B. Cohen, T. R. Loveland, and C. E. Woodcock, "Opening the archive: How free data has enabled the science and monitoring promise of landsat," *Remote Sensing of Environment*, vol. 122, pp. 2–10, 2012.

[7] ESA. (2019) Esa copernicus program. [Online]. Available: https://www.copernicus.eu/en/about-copernicus

[8] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[10] G. E. Hinton, “Learning multiple layers of representation,” *Trends in cognitive sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[11] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[12] X. Jia, B.-C. Kuo, and M. M. Crawford, “Feature mining for hyperspectral image classification,” *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676–697, 2013.

[13] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, “Advances in hyperspectral image classification: Earth monitoring with statistical learning methods,” *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 45–54, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[15] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, “Deep learning in remote sensing: A comprehensive review and list of resources,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[16] G.-J. Qi and J. Luo, “Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[17] Y. Ouali, C. Hudelot, and M. Tami, “An overview of deep semi-supervised learning,” *arXiv preprint arXiv:2006.05278*, 2020.

[18] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[22] Y. Wang, Q. Wang, S. Shi, X. He, Z. Tang, K. Zhao, and X. Chu, “Benchmarking the performance and energy efficiency of ai accelerators for ai training,” in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 744–751.

[23] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, “Benchmark analysis of representative deep neural network architectures,” *IEEE access*, vol. 6, pp. 64 270–64 277, 2018.

[24] T. Ben-Nun and T. Hoefler, “Demystifying parallel and distributed deep learning: An in-depth concurrency analysis,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–43, 2019.

[25] Z. Tang, S. Shi, X. Chu, W. Wang, and B. Li, “Communication-efficient distributed deep learning: A comprehensive survey,” *arXiv preprint arXiv:2003.06307*, 2020.

[26] H. Yu, S. Yang, and S. Zhu, “Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5693–5700.

[27] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, “Remote sensing big data computing: Challenges and opportunities,” *Future Generation Computer Systems*, vol. 51, pp. 47–60, 2015.

[28] H. Özköse, E. S. Arı, and C. Gencer, “Yesterday, today and tomorrow of big data,” *Procedia-Social and Behavioral Sciences*, vol. 195, pp. 1042–1050, 2015.

[29] NASA. (1999) Earth observatory (1999). [Online]. Available: https://earthobservatory.nasa.gov/features/RemoteSensing/remote_02.php

[30] X. Yao, G. Li, J. Xia, J. Ben, Q. Cao, L. Zhao, Y. Ma, L. Zhang, and D. Zhu, “Enabling the big earth observation data via cloud computing and dggs: Opportunities and challenges,” *Remote Sensing*, vol. 12, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/1/62

[31] J. Xia, C. Yang, and Q. Li, “Building a spatiotemporal index for earth observation big data,” *International journal of applied earth observation*

*and geoinformation*, vol. 73, pp. 245–252, 2018.

[32] X. Hu, J. S. Næss, C. M. Iordan, B. Huang, W. Zhao, and F. Cherubini, "Recent global land cover dynamics and implications for soil erosion and carbon losses from deforestation," *Anthropocene*, vol. 34, p. 100291, 2021.

[33] D. G. Goodin, K. L. Anibas, and M. Bezymennyi, "Mapping land cover and land use from object-based classification: An example from a complex agricultural landscape," *International Journal of Remote Sensing*, vol. 36, no. 18, pp. 4702–4723, 2015.

[34] P. Vicharnakorn, R. P. Shrestha, M. Nagai, A. P. Salam, and S. Kiratiprayoon, "Carbon stock assessment using remote sensing and forest inventory data in savannakhet, lao pdr," *Remote Sensing*, vol. 6, no. 6, pp. 5452–5479, 2014.

[35] R. D. D. Altarez, A. Apan, and T. Maraseni, "Spaceborne satellite remote sensing of tropical montane forests: a review of applications and future trends," *Geocarto International*, pp. 1–29, 2022.

[36] C. Persello, J. D. Wegner, R. Hänsch, D. Tuia, P. Ghamisi, M. Koeva, and G. Camps-Valls, "Deep learning and earth observation to support the sustainable development goals: Current approaches, open challenges, and future opportunities," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 2, pp. 172–200, 2022.

[37] G. J. Schumann, G. R. Brakenridge, A. J. Kettner, R. Kashif, and E. Niebuhr, "Assisting flood disaster response with earth observation data and products: A critical assessment," *Remote Sensing*, vol. 10, no. 8, p. 1230, 2018.

[38] E. Lancheros, A. Camps, H. Park, P. Rodriguez, S. Tonetti, J. Cote, and S. Pierotti, "Selection of the key earth observation sensors and platforms focusing on applications for polar regions in the scope of copernicus system 2020–2030," *Remote Sensing*, vol. 11, no. 2, p. 175, 2019.

[39] L. P. Bobylev and M. W. Miles, "Sea ice in the arctic paleoenvironments," in *Sea Ice in the Arctic*. Springer, 2020, pp. 9–56.

[40] T. Vihma, "Effects of arctic sea ice decline on weather and climate: A review," *Surveys in Geophysics*, vol. 35, no. 5, pp. 1175–1214, 2014.

[41] M. R. Najafi, F. W. Zwiers, and N. P. Gillett, "Attribution of arctic temperature change to greenhouse-gas and aerosol influences," *Nature Climate Change*, vol. 5, no. 3, p. 246, 2015.

[42] J. C. Stroeve, M. C. Serreze, M. M. Holland, J. E. Kay, J. Malanik, and A. P. Barrett, "The arctic's rapidly shrinking sea ice cover: a research synthesis," *Climatic Change*, vol. 110, no. 3, pp. 1005–1027, Feb 2012. [Online]. Available: https://doi.org/10.1007/s10584-011-0101-1

[43] S. Haykin, E. O. Lewis, R. K. Raney, and J. R. Rossiter, *Remote sensing of sea ice and icebergs*. John Wiley & Sons, 1994, vol. 13.

[44] O. J. Hegelund, A. Everett, T. Cheeseman, P. Wagner, N. Hughes, M. Pierechod, K. Southerland, P. Robinson, J. Hutchings, Å. Kiærbech *et al.*, "Extending the ice watch system as a citizen science project for the collection of in-situ sea ice observations," in *EGU General Assembly Conference Abstracts*, 2020, p. 7126.

[45] J. K. Hutchings and M. K. Faber, "Sea-ice morphology change in the canada basin summer: 2006–2015 ship observations compared to observations from the 1960s to the early 1990s," *Frontiers in Earth Science*, vol. 6, p. 123, 2018.

[46] H. Kaartokallio, M. A. Granskog, H. Kuosa, and J. Vainio, "Ice in subarctic seas," *Sea Ice*, pp. 630–644, 2017.

[47] C. Haas, "Airborne electromagnetic sea ice thickness sounding in shallow, brackish water environments of the caspian and baltic seas," in *International Conference on Offshore Mechanics and Arctic Engineering*, vol. 47470, 2006, pp. 717–722.

[48] eos. (18) Types of remote sensing: Technology changing the world.

[49] J. Grahn, "Multi-frequency radar remote sensing of sea ice. modelling and interpretation of polarimetric multi-frequency radar signatures of sea ice," 2018.

[50] D. Murashkin, G. Spreen, M. Huntemann, and W. Dierking, "Method for detection of leads from sentinel-1 sar images," *Annals of Glaciology*, vol. 59, no. 76pt2, p. 124–136, 2018.

[51] T. I. S. of the Norwegian Meteorological Institute (NIS). (2022) Ice charts -. [Online]. Available: https://cryo.met.no/en/latest-ice-charts

[52] J. Lohse, "On automated classification of sea ice types in sar imagery," Ph.D. dissertation, Universitetet i Tromsø, 2021.

[53] L.-K. Soh, C. Tsatsoulis, D. Gineris, and C. Bertoia, "Arktos: An intelli-

gent system for sar sea ice image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 1, pp. 229–248, 2004.

[54] J. A. Karvonen, "Baltic sea ice sar segmentation and classification using modified pulse-coupled neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1566–1574, 2004.

[55] W. Dierking, "Sea ice monitoring by synthetic aperture radar," *Oceanography*, vol. 26, no. 2, pp. 100–111, 2013.

[56] ESA. (2022) Sentinel-1 acquisition modes. [Online]. Available: https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes

[57] J.-S. Lee and E. Pottier, *Polarimetric radar imaging: from basics to applications*. CRC press, 2017.

[58] C. Oliver and S. Quegan, *Understanding synthetic aperture radar images*. SciTech Publishing, 2004.

[59] A. J. Schweiger, "Changes in seasonal cloud cover over the arctic seas from satellite and surface observations," *Geophysical Research Letters*, vol. 31, no. 12, 2004.

[60] N. Zakhvatkina, V. Smirnov, and I. Bychkova, "Satellite sar data-based sea ice classification: An overview," *Geosciences*, vol. 9, no. 4, p. 152, 2019.

[61] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 1, pp. 6–43, 2013.

[62] W. Parker, "Discover the benefits of radar imaging: The top 10 considerations for buying and using synthetic aperture radar imagery," *Earth Imaging Journal*, 2012.

[63] W. Dierking, "Sea ice and icebergs," in *Maritime Surveillance with Synthetic Aperture Radar*. Institution of Engineering and Technology, 2020, pp. 173–225.

[64] C. Elachi and J. J. Van Zyl, *Introduction to the physics and techniques of remote sensing*. John Wiley & Sons, 2021.

[65] J. Lohse, "On automated classication of sea ice types in sar imagery," Ph.D. dissertation, UiT The Arctic university of Norway, 2020.

[66] K. Čotar, K. Oštir, and Ž. Kokalj, "Radar satellite imagery and automatic detection of water bodies," *Geodetski glasnik*, vol. 50, no. 47, pp. 5–15, 2016.

[67] R. Pelich, M. Chini, R. Hostache, P. Matgen, and C. López-Martinez, "Coastline detection based on sentinel-1 time series for ship- and flood-monitoring applications," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 10, pp. 1771–1775, 2021.

[68] N. K. Sinha and M. Shokr, *Sea ice: physics and remote sensing*. John Wiley & Sons, 2015.

[69] T. Armstrong, "World meteorological organization. wmo sea-ice nomenclature. terminology, codes and illustrated glossary. edition 1970. geneva, secretariat of the world meteorological organization, 1970.[ix], 147 p.[including 175 photos] + corrigenda slip.(wmo/omm/bmo, no. 259, tp. 145.)," *Journal of Glaciology*, vol. 11, no. 61, pp. 148–149, 1972.

[70] F. T. Ulaby, F. Kouyate, B. Brisco, and T. L. Williams, "Textural infornation in sar images," *IEEE Transactions on Geoscience and Remote Sensing*, no. 2, pp. 235–245, 1986.

[71] DeepAI.og. (1999) Earth observatory (1999). [Online]. Available: deepAI.org

[72] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, "A quantitative theory of immediate visual recognition," *Progress in brain research*, vol. 165, pp. 33–56, 2007.

[73] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5070–5079.

[74] K. Y. Yip, C. Cheng, and M. Gerstein, "Machine learning and genome annotation: a match meant to be?" *Genome biology*, vol. 14, no. 5, pp. 1–10, 2013.

[75] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[76] O. Chapelle, M. Chi, and A. Zien, "A continuation method for semi-supervised svms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 185–192. [Online]. Available:

https://doi.org/10.1145/1143844.1143868

[77] X. J. Zhu, "Semi-supervised learning literature survey," 2005.

[78] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[79] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[80] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234–2242, 2016.

[81] D. Burago, S. Ivanov, and Y. Kurylev, "A graph discretization of the laplace–beltrami operator," *Journal of Spectral Theory*, vol. 4, no. 4, pp. 675–714, 2015.

[82] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in neural information processing systems*, 2014, pp. 3581–3589.

[83] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," in *International conference on machine learning*. PMLR, 2016, pp. 1445–1453.

[84] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum, "Deep convolutional inverse graphics network," *arXiv preprint arXiv:1503.03167*, 2015.

[85] S. Narayanaswamy, T. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, "Learning disentangled representations with semi-supervised deep generative models," 2018.

[86] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.

[87] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, "Semi-supervised learning with ladder networks," *arXiv preprint arXiv:1507.02672*, 2015.

[88] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[89] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *arXiv preprint arXiv:1905.02249*, 2019.

[90] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[91] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698.

[92] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, 2004, pp. 321–328.

[93] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[94] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[95] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[96] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[97] M. Langer, Z. He, W. Rahayu, and Y. Xue, "Distributed training of deep learning models: A taxonomic perspective," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 12, pp. 2802–2818, 2020.

[98] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, "Large scale distributed deep networks," *Advances in neural information processing systems*, vol. 25, 2012.

[99] R. Mayer, C. Mayer, and L. Laich, "The tensorflow partitioning and scheduling problem: it's the critical path!" in *Proceedings of the 1st Workshop on Distributed Infrastructures for Deep Learning*, 2017, pp. 1–6.

[100] A. Mirhoseini, H. Pham, Q. V. Le, B. Steiner, R. Larsen, Y. Zhou, N. Kumar, M. Norouzi, S. Bengio, and J. Dean, "Device placement optimization with reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2430–2439.

[101] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, 2014, pp. 583–598.

[102] M. Li, L. Zhou, Z. Yang, A. Li, F. Xia, D. G. Andersen, and A. Smola, "Parameter server for distributed machine learning," in *Big learning NIPS workshop*, vol. 6, 2013, p. 2.

[103] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[104] B. C. Ooi, K.-L. Tan, S. Wang, W. Wang, Q. Cai, G. Chen, J. Gao, Z. Luo, A. K. Tung, Y. Wang *et al.*, "Singa: A distributed deep learning platform," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 685–688.

[105] M. Langer, A. Hall, Z. He, and W. Rahayu, "Mpca sgd—a method for distributed training of deep learning models on spark," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2540–2556, 2018.

[106] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *arXiv preprint arXiv:1705.09056*, 2017.

[107] S. Zhang, "Distributed stochastic optimization for deep learning," Ph.D. dissertation, New York University, 2016.

[108] R. Rabenseifner, "Optimization of collective reduction operations," in *International Conference on Computational Science*. Springer, 2004, pp. 1–9.

[109] A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, "S-caffe: Co-

designing mpi runtimes and caffe for scalable deep learning on modern gpu clusters," in *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2017, pp. 193–205.

[110] C.-H. Chu, X. Lu, A. A. Awan, H. Subramoni, J. Hashmi, B. Elton, and D. K. Panda, "Efficient and scalable multi-source streaming broadcast on gpu clusters for deep learning," in *2017 46th International Conference on Parallel Processing (ICPP)*. IEEE, 2017, pp. 161–170.

[111] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *Advances in neural information processing systems*, vol. 32, pp. 103–112, 2019.

[112] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, "Geeps: Scalable deep learning on distributed gpus with a gpu-specialized parameter server," in *Proceedings of the Eleventh European Conference on Computer Systems*, 2016, pp. 1–16.

[113] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: training imagenet in 1 hour," *CoRR*, vol. abs/1706.02677, 2017. [Online]. Available: http://arxiv.org/abs/1706.02677

[114] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why m heads are better than one: Training a diverse ensemble of deep networks," *arXiv preprint arXiv:1511.06314*, 2015.

[115] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Conference on Neural Information Processing Systems (NIPS), Deep Learning Workshop*, 2014. [Online]. Available: https://openreview.net/forum?id=S1g2JnRcFX

[116] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=S1gmrxHFvB

[117] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers." in *Interspeech*, 2017, pp. 3697–3701.

[118] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[119] J. Kim, M. Hyun, I. Chung, and N. Kwak, "Feature fusion for online mutual knowledge distillation," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 4619–4625.

[120] S. Park and N. Kwak, "Feature-level ensemble knowledge distillation for aggregating knowledge from multiple networks," in *ECAI 2020*. IOS Press, 2020, pp. 1411–1418.

[121] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," *arXiv preprint arXiv:1804.03235*, 2018.

[122] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," 2011.

[123] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *International conference on machine learning*. PMLR, 2015, pp. 1737–1746.

[124] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1510.00149

[125] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.

[126] M. Höhfeld and S. E. Fahlman, "Probabilistic rounding in neural network learning with limited precision," *Neurocomputing*, vol. 4, no. 6, pp. 291–299, 1992.

[127] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

[128] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compres-

sion: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2017.

[129] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 440–445. [Online]. Available: https://aclanthology.org/D17-1045

[130] C.-Y. Chen, J. Choi, D. Brand, A. Agrawal, W. Zhang, and K. Gopalakrishnan, "Adacomp: Adaptive residual gradient compression for data-parallel distributed training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[131] C. Renggli, S. Ashkboos, M. Aghagolzadeh, D. Alistarh, and T. Hoefler, "Sparcml: High-performance sparse communication for machine learning," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–15.

[132] X. Wu, H. Xu, B. Li, and Y. Xiong, "Stanza: Layer separation for distributed training in deep learning," *IEEE Transactions on Services Computing*, 2020.

[133] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[134] S. Khaleghian, J. P. Lohse, and T. Kræmer, "Synthetic-Aperture Radar (SAR) based Ice types/Ice edge dataset for deep learning analysis," 2020. [Online]. Available: https://doi.org/10.18710/QAYI4O

[135] S. Khaleghian, H. Ullah, T. Kræmer, T. Eltoft, and A. Marinoni, "Deep semisupervised teacher–student model based on label propagation for sea ice classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 10 761–10 772, 2021.

[136] M. Douze, A. Szlam, B. Hariharan, and H. Jégou, "Low-shot learning with large-scale diffusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3349–3358.

[137] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks. 2nd international conference on learning representations, iclr 2014," in *2nd International Conference on Learning*

*Representations, ICLR 2014*, 2014.