Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Sigrid Marie Eie Ledsaak and Nora Strømberg Brask

Sea Ice Segmentation in Sentinel-1 Imagery using a Convolutional Neural Network

Master's thesis in Engineering and ICT
Supervisor: Hossein Nahavandchi
Co-supervisor: Hongchao Fan

June 2021

NTNU
Norwegian University of Science and Technology

Sigrid Marie Eie Ledsaak and Nora Strømberg Brask

Sea Ice Segmentation in Sentinel-1 Imagery using a Convolutional Neural Network

Master's thesis in Engineering and ICT
Supervisor: Hossein Nahavandchi
Co-supervisor: Hongchao Fan
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

The Arctic sea ice is critical for the global environment and represents a significant challenge for navigation in polar regions. Hence, sea ice monitoring has become a crucial responsibility of the states located within the Arctic circle. Due to the large extent of the Arctic sea, satellite remote sensing is the most acknowledged monitoring approach. By being independent of natural illumination, and able to penetrate cloud cover, the active imaging system, Synthetic Aperture Radar (SAR), is highly qualified for the polar conditions. Through their national ice services, the Arctic states provide manually drawn ice charts based on satellite imagery. Despite the expertise of the sea ice analysts in the national ice services, there will always exist an element of subjectivity in the ice charts. Moreover, manual ice charting is time-consuming, extending the time between acquisition and publication of the information. Due to wind and strong currents, the sea ice is highly dynamic, therefore, the delay can be critical for the navigators passaging through the Arctic sea. By applying an automatic approach for sea ice segmentation in SAR imagery, these ice charts can be produced within a shorter time, and with a lower degree of subjectivity. In this thesis, the convolutional neural network architecture, U-Net, has been modified and implemented in a Python software package. Variations of this network were applied to automatically perform sea ice segmentation in noise corrected SAR scenes. Moreover, a balancing approach has been implemented and evaluated. The best performance was achieved by the network trained on an imbalanced dataset with an R^2-score of 0.87. Finally, a binary image segmentation method was proposed, and it obtained results comparable to the state-of-the-art.

(This page is intentionally left blank)

Sammendrag

Den Arktiske havisen er avgjørende for det globale miljøet og representerer en betydelig utfordring for navigering i polare områder. Derfor har overvåking av havis blitt et viktig ansvar for statene som ligger innenfor polarsirkelen. På grunn av det store omfanget av det Nordishavet er den mest anerkjente overvåkingsmetoden via satellittbasert fjernmåling. Ved å være uavhengig av naturlig belysning og kapabel til å trenge gjennom skydekket, er det aktive radarsystemet, Syntetisk Apertur-Radar (SAR), høyt kvalifisert for de polare forholdene. Gjennom sine nasjonale istjenester tilbyr de arktiske statene manuelt tegnede iskart som er basert på satellittbilder. Til tross for ekspertisen til ishavs analytikerne i de nasjonale istjenestene, vil det alltid eksistere et element av subjektivitet i iskartene. Videre er manuell iskartlegging tidkrevende og forlenger dermed tiden mellom anskaffelse og publisering av informasjonen. På grunn av vind og sterke strømmer er havisen veldig dynamisk. Derfor kan denne forsinkelsen være kritisk for navigatørene som passerer gjennom Nordishavet. Ved å anvende en automatisk tilnærming for segmentering av havis i SAR-bilder kan disse iskartene produseres innen kortere tid og med lavere grad av subjektivitet. I denne oppgaven, er den konvolusjonelle nevrale nettverksarkitekturen, U-Net, modifisert og implementert i en Python-programvarepakke. Variasjoner av dette nettverket ble brukt til å utføre automatisk segmentering av havis i støykorrigerte SAR-scener. Videre er en balanseringsmetode implementert og evaluert. Nettverket som var trent med et ubalansert datasett oppnådde den beste ytelsen med en R^2 -score på 0,87. Til slutt ble en binær bildesegmenteringsmetode foreslått. Resultatene fra denne er sammenlignbare med moderne forskning.

(This page is intentionally left blank)

Acknowledgements

The authors of this thesis would like to express their deepest appreciation to Proffessor Hossein Nahavandchi and Proffessor Dr. Hongchao Fan for valuable guidance throughout the work related to this thesis. Moreover, we would like to extend our sincere thanks to the AI4Arctic project, carried out by the Danish Meterological Institute, the Technical University of Denmark, and Nansen Environmental Remote Sensing Center for providing the ASIP Sea Ice Dataset - version 2. An additional thanks is given to Leif Toudal Pedersen and Anton Korosov for assistance related to the dataset. The authors would also like to thank Chaoquan Zhang for providing access to the server utilized in this thesis. Finally, we gratefully acknowledge the support provided by the founder of Vake, Adrian Tofting, Dr. Roghayeh Shamshiri and Gefei Kong during the implementation of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sea ice within the Arctic circle, commonly referred to as the Arctic Sea ice, is critical for the global environment, and has a large impact on human interactions within these areas. In the context of marine navigation in the polar regions, the Arctic sea ice represents a significant navigational hazard. Ice bergs and floating sea ice can severely damage the vessels or block the originally planned route. In order to passage through the Arctic sea, it is therefore essential for the navigators to be able to make decisions based on up-to-date sea ice information. Hence, monitoring of the Arctic sea ice has played a fundamental role in human navigation within the polar regions since the The Viking Age[1]. Moreover, due to ice melting caused by global warming, there will, in the future, be an increased interest in the shipping routes connecting the Atlantic and the Pacific oceans.

Due to the large dimensions of the Arctic sea, the most preferable monitoring approach is by remote sensing satellites. Remote sensing is the process of acquiring information about an object, area or phenomenon from a distance [1]. During the recent decades, the field of sea ice monitoring by remote sensing has undergone drastic changes. In 2013, Teleti and Luis [2] reviewed published literature to evaluate different remote sensing techniques for observing the physical features of the sea ice. They concluded that microwave sensors are superior to optical- and infrared sensors when monitoring sea ice in polar regions. More specifically, synthetic aperture radar (SAR) systems on board satellites are well suited for sea ice monitoring. SAR is a system of active data collection where a satellite sensor transmits energy before recording the amount of reflected energy after interaction with the earth. This returning energy is often referred to as *backscattering*. The signals in SAR-data are responsive to surface characteristics like structure and moisture. Thereby, it is suitable for monitoring the Earth's surface. Given that SAR is an active imaging system, it is independent of weather conditions and illumination from the sun. This makes it highly qualified for the rough polar conditions. Auxiliary data, such as optical imagery, thermal-infrared and passive microwave radiometer data, is commonly applied to support the SAR-observations when available.

Another aspect of remote sensing monitoring is the process of making the acquired satellite information available to users in an understandable format, shortly after they are obtained. The national ice services in the polar regions provide manually drawn ice charts. Experienced sea ice analysts study the SAR images and the auxiliary data in order to draw conclusions about the ice conditions in a specific area. Even though these analysts are efficient, manual ice charting is time-consuming in which extends the waiting time for the marine navigators. Furthermore, despite the professional skills of the analysts, there will, to a certain degree, always exist an element of subjectivity in the humanly made ice charts. The European Union's Earth Observation Programme, Copernicus, found that 90-95% of the ice edges were the same when comparing two ice charts [3]. Additionally, they found that up to 10% of the points can differ with values up to 20% in the comparison. Thus, it is preferable to apply an automatic approach for producing sea ice charts.

The University of Kansas, in cooperation with NASA, started studying the use of artificial intelligence in terms of expert systems for sea ice classification based on SAR in 1990. The study

[1] https://earthobservatory.nasa.gov/features/SeaIce

was continued under a NASA grant and resulted in various techniques for identifying sea ice features in SAR imagery [4], [5]. One of the first fully-automatic near real-time artificial intelligence sea ice classification systems called ARKTOS was developed over a ten year period by [6]. This system was operational in the U.S. National Ice Center from year 1999. Since then, similar systems based on machine learning with manually extracted features have been developed, including the systems presented in [7], [8] and [9].

Due to the immense increase in daily data production, and the comprehensive software improvements over the last years, there has been a considerable progress in the field of deep learning. This progression has enabled the development of systems for automatic feature extraction. By being exposed to a large set of training data, a deep neural network can learn to make successful predictions which map a set of inputs to a set of outputs. Studies have shown that computers can outperform human experts in the task of image interpretation [10]. Moreover, once the deep networks are trained, they are able to execute repetitive tasks within a shorter time compared to a person [2]. The extent of information in satellite images is enormous, and it may be insurmountable for a person to detect all the details in the acquired data. Hence, the application of deep learning in satellite image interpretation has become popular.

Convolutional neural networks (CNNs), a subclass of deep neural networks, are commonly applied in visual image analysis [11]. By sequentially exploiting convolution, a mathematical operation where two functions are combined into one, CNNs are able to extract features, (e.g. shapes, objects, textures) from the input images. This approach has proven to be efficient and accurate in computer vision tasks such as object detection, image classification and semantic segmentation. In 2019, Ghorbanzadeh et al. compared the performance of machine learning methods and different CNNs in optical satellite data to create landslide maps in [12]. They found that the CNNs performed best overall. However, the performance of the CNNs are highly dependent on the design of the networks, and do not automatically outperform the other approaches. Nevertheless, they emphasized that with the right setup and a large amount of training data, the deep learning approaches has a great potential in the field of remote sensing. Moreover, Boulze et al. [13] recently developed an algorithm for classification of sea ice types using SAR-data and a CNN. They found that the performance of the CNN was better than the performance of a random forest classifier[3], and that the CNN was less sensitive to noise in the SAR data.

Semantic image segmentation is a commonly applied approach for sea ice monitoring. In this approach the goal is to assign a class label to each pixel in the image. Each class label describes certain characteristics in which the pixels assigned to this class possesses. Thereby, the image is divided into segments of classes where each class has a semantic meaning. There is a large variety in existing applications for sea ice segmentation. Firstly, Boulze et al. [13] present an approach for partitioning the image into different *types* of sea ice, while Wang et al. [14] focus on the stage of development. Another common approach is to divide the image into segments based on the estimates of sea ice concentration as presented by Kruk et al. [15].

U-Net [16] is a commonly applied CNN for image segmentation. This network was primarily developed for biomedical image segmentation, but has proven to be useful in sea ice segmentation tasks as well [17]. This fully convolutional neural network consists of one contracting path and one expansive path in which provides a u-shaped architecture. In the contracting path convolution is applied repeatedly to reduce the spatial information and increase the feature information. In the expansive path, the spatial information is increased. This is done by concatenating the spatial information from the contracting path with the feature information. Thereby, the image is segmented into high resolution features.

In 2014 and 2016, respectively, the European Space Agency (ESA)[4] launched the two operational radar satellites; Sentinel-1A and Sentinel-1B. SAR images acquired with Sentinel-1, are widely used for sea ice monitoring in polar regions. This is due to the satellite constellation's short revisiting time (6 days), large area coverage (400km) and independence of cloud cover and light

[2]https://becominghuman.ai/deep-learning-and-its-5-advantages-eaeee1f31c86
[3]https://en.wikipedia.org/wiki/Random$_forest$
[4]http://www.esa.int/

conditions. Among the variations of products provided by the Sentinel-1 constellation, the dual polarised Level-1 Ground Range Distance Extra Wide swath mode (S1 L1 GRD EW) products are commonly applied in sea ice monitoring. These consists of five sub-swaths with a spatial resolution of 93x87 m and a pixel spacing of 40x40 m.

One of the major challenges associated with dual polarizations of S1 L1 EW GRD products is the presence of radiometric distortions. Specifically, the cross-polarized channel is highly affected by thermal additive noise. Hence, ESA provided noise correcting vectors for removal of additive noise in S1 images in 2015[5]. However, according to Sun and Li [18] and Park et al. [19], there was still a significant amount of noise remaining in the cross-polarized images after having applied the noise correction provided by ESA. Therefore, in 2018, Nansen Environmental and Remote Sensing Center (NERSC) developed a new denoising method for removal of additive noise in cross-polarized images[19]. This improved the quality of various SAR intensity-based applications, however, the data still suffered from textural noise. Hence, in their sequel work they addressed this problem by removing the residual local variations within the images [20].

With a shared ambition of improving existing approaches for sea ice monitoring, the Danish Meteorological Institute (DMI), the Technical University of Denmark (DTU) and NERSC carried out a collaborative project, AI4Arctic. This project aimed at exploring the potential of deep learning in earth observation applications within polar regions. Moreover, by publishing their discoveries and data, they facilitate further research in the field of deep learning and sea ice monitoring. The ASIP (Automated Sea Ice Products) Sea Ice Dataset – version 2[6] (hereafter referred to as the ASIP dataset), was made available by AI4Arctic to the public in September 2020. This dataset contains S1 L1 GRD EW SAR imagery acquired from the coast of Greenland during the time interval 2018-2019. These SAR scenes have been corrected for noise according to the method developed by NERSC.

In this thesis, an automatic approach for sea ice segmentation based on the noise corrected SAR scenes provided in the ASIP dataset will be presented. A basic balancing approach was implemented, and three different versions of the dataset including a binary-, a multiclass- and a balanced multiclass dataset were generated. These have been utilized to train and evaluate four separate instances of the U-Net architecture. As a contribution to the field of automatic sea ice segmentation, the effect of the balancing method on the network performance has been examined. Furthermore, a binary version of the dataset was generated to consider the approach's ability to differentiate between open water and sea ice. Finally, there is no other study that exclusively applies the noise corrected S1 SAR images from the ASIP dataset in a U-Net, to the best of the authors' knowledge. This experience can definitely be of reference value for projects or research when using the same or similar data.

The following parts of this thesis are structured in 7 chapters. Chapter 2 includes fundamental theory regarding the Arctic sea ice and ice charting, SAR techniques, deep learning and the network architecture applied in this thesis. Chapter 3 contains information about the study area and the dataset applied in the neural network. The largest chapter in this thesis, Chapter 4, lists all the necessary steps preparing the dataset for the network, model prediction, and finally an evaluation method is presented. In Chapter 5, the experimental results are presented and discussed, followed by a discussion of the decisions made throughout this thesis. Finally, there will be a conclusion of this thesis and a description of future work.

[5]https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/document-library/-/asset_publisher/1dO7RF5fJMbd/content/thermal-denoising-of-products-generated-by-the-sentinel-1-ipf

[6]https://data.dtu.dk/articles/dataset/AI4Arctic_ASIP_Sea_Ice_Dataset_-_version_2/13011134

(This page is intentionally left blank)

Chapter 2

Fundamentals

This section will provide a fundamental explanation of the theoretical aspects related to this project. In section 2.1 the Arctic sea ice is described, followed by a presentation of the ice charting process in section 2.2. Thereafter, fundamental aspects related to SAR systems and Sentinel-1 imagery including radiometric distortions will be discussed in section 2.3. Moreover, section 2.4 will give an introduction to deep learning and the network architecture utilized in this thesis. Finally, the performance measures applied in the evaluation of the deep neural network are presented in section 2.5.

2.1 Arctic Sea Ice

Sea ice is defined as frozen sea water that floats on the ocean surface, and the Arctic sea ice is the sea ice present within the Arctic circle. Generally, the Arctic sea ice is at its largest extent in March, and reaches a minimum in September [1]. Due to the rough weather conditions in the polar regions, the Arctic sea ice is very dynamic. And, in order to passage these areas, it is crucial to navigate based on up-to-date information about the current ice conditions. Monitoring of the Arctic sea ice is therefore highly relevant. The minimum and maximum extent of the Arctic sea ice is visualised in Fig. 1. As can be observed in the leftmost image, there is a possibility for passing the Arctic during summer time.

Another reason for monitoring the Arctic sea ice is that it plays an important role in the global environment. By reflecting 80% of the sunlight back into space[2], it slows down the global warming. With increasing temperatures, the extent of the existing sea ice decreases due to warmer weather. This makes the Arctic sea ice a good indicator of global warming as well as an countermeasure for it.

In the context of this thesis, all types of floating ice will be considered sea ice. I.e. ice bergs, glaciers, and ice shelves, that originally arose on land and is now floating on the ocean, will be referred to as sea ice. Since sea ice can take such different forms, and changes quickly, the challenge of sea ice monitoring is complex.

2.2 Ice Charts

Ice charts are intended to support planning and operational services in regions prone to floating sea ice. In order to plan a safe journey, it is important to be aware of the current ice conditions for the specific area. Today, most ice services are manually labelling their ice charts by studying SAR imagery in addition to other available data, such as optical imagery, thermal-infrared and passive

[1] https://earthobservatory.nasa.gov/features/SeaIce
[2] https://nsidc.org/cryosphere/quickfacts/seaice.html

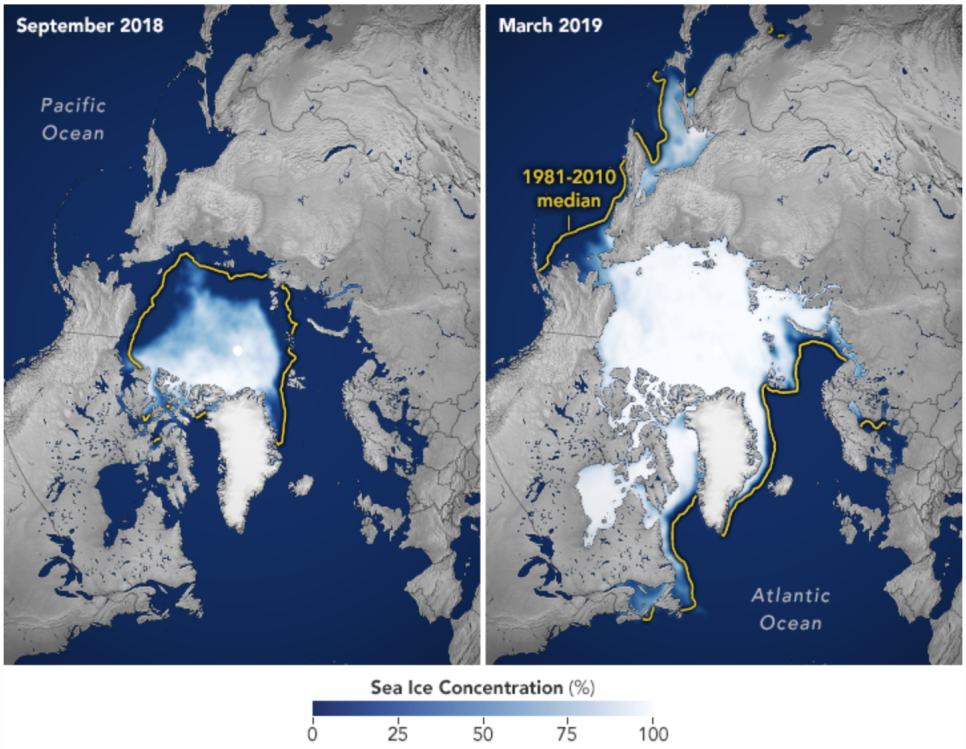


Figure 1: The extent of the Arctic sea ice in September 2018 (left image) and in March 2019(right image)

microwave radiometer data. Geographical coverage and the publication frequency of the ice charts depend on season, ship routes and data availability.

The Dannish Meteorological Institute (DMI)[3] are responsible for monitoring the coastal areas of Greenland, and their ice charts are utilized in this thesis. According to DMI, an ice chart is an interpretation of the ice conditions in a given area at the time of acquisition. It consists of manually drawn polygons in which the ice concentration in each polygon is sufficiently similar. The estimated polygons are based on the subjective opinion of an ice analyst, and there is no definite uncertainty linked to the ice chart. However, the analysts pay extra attention to the areas containing ice edges. This is due to the importance of knowing the ice properties and the marginal zone of the ice when executing operations close to the edges. As a natural consequence, the estimates are less accurate inside the polygons. This might give rise to imperfections of the deep neural network when applying the polygons as ground truth for the image segmentation.

When interpreting an ice chart it is important to understand the convention applied for the specific chart. DMI has applied the Egg Code[4] for characterizing the sea ice in their ice charts. This convention was developed by the World Meteorological Organization (WMO)[5] and is a widespread convention describing ice thickness, stage of development, shape and concentration of ice contained in a polygon. An illustration of the convention can be viewed in Fig. 2.

Which is evident in Fig. 2, C_t represents the total concentration of ice. Moreover, C_a, C_b, C_c and C_d are the partial concentration of each ice type. S_o, S_a, S_b, S_c, S_d and S_e are variable identifiers for the stage of development. And F_a, F_b, F_c, F_d and F_e are representing the form of the ice, such as strips, patches, icebergs or fast ice.

[3]https://www.dmi.dk/

[4]https://www.canada.ca/en/environment-climate-change/services/ice-forecasts-observations/publications/interpreting-charts/chapter-1.html

[5]https://public.wmo.int/en

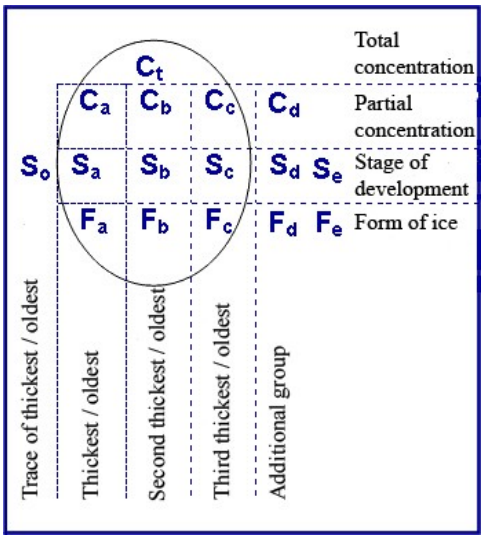


Figure 2: The Egg Code

The only parameter applied in this thesis is the total ice concentration, C_t. This is given in SIGRID-3 code (Sea Ice GeoReferenced Information and Data)[6]. Each code and its explanation is given in table 1, and these will provide a basis for the classes that will be used as labels for the deep neural network.

Table 1: The SIGRID-3 codes and their explaination. The fractions of 10 are representing the concentration of ice. * The category "Bergy water" is used for open sea(water category)in the DMI ice charts. The category "Ice Free" is not used in the DMI ice charts. ** The category "9+/10" is used in the DMI ice charts for sea ice that is fully compacted, but not fast ice (100% ice)

Definition, concentration	Sigrid3 code (CT, CA, CB and CC)
Ice Free	00
Less than 1/10 (open water)	01
Bergy water	02*
1/10	10
2/10	20
3/10	30
4/10	40
5/10	50
6/10	60
7/10	70
8/10	80
9/10	90
9+/10 (95%)	91**
10/10	92

[6]https://library.wmo.int/doc_num.php?explnum_id=9270

2.3 Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar (SAR) is a system of active data collection in which a sensor transmits energy and records the amount of the transmitted energy reflected after interaction with the target. The signals in SAR-data are responsive to surface characteristics like structure and moisture, thus suitable for monitoring the Earth's surface.

Radar satellites are recording the earth obliquely with an off-nadir angle, meaning that there is a non-zero angle between the satellite's nadir position and the SAR beam. Nadir is the point on the earth's surface directly beneath the satellite's location. The radar beam is moving across the swath (the area being covered by the image) with an increasing incidence angle from near to far range. Near range is the portion of the image swath closest to the nadir track of the satellite platform, and far range the portion furthest away. This incidence angle, i.e. the angle between the radar beam and ground surface, will participate in the input to the deep neural network applied in this thesis. In Fig. 3, the incidence angle is illustrated and marked by the letter, A. B refers to the look angle which has the same quantitative angle as the incidence angle. C is the length of the path between the target and the radar, called slant range distance. And D is the ground range distance.

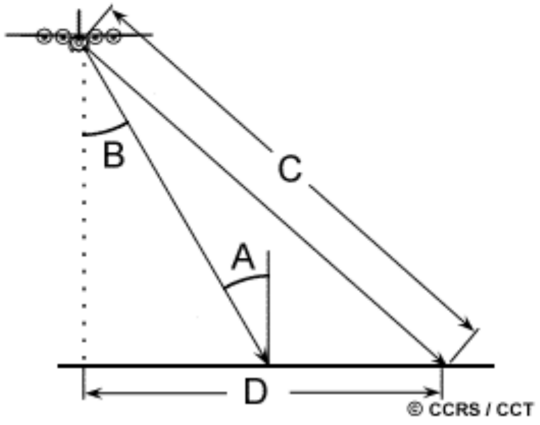


Figure 3: An illustration of a radar satellite's incidence angle (A) , look angle (B) , slant range distance (C) and ground range distance (D)

Source: Canada Centre for Remote Sensing

The spatial resolution of the radar image produced by the sensor is directly affected by the ratio between the wavelength and the length of the antenna. I.e. a longer antenna will give a higher spatial resolution than a short one, for a given wavelength. According to NASA[7], one would need an antenna of about 4250 m to achieve a spatial resolution of 10 m from a C-band radar satellite operating with a wavelength of 5 cm. For a satellite in space, an antenna of this size is unacceptable, hence the synthetic aperture. By combining a sequence of acquisitions from a shorter antenna to simulate a larger antenna, the SAR process is able to provide a higher resolution.

SAR sensors utilize longer wavelengths than optical sensors, in which provides the benefit of independence from weather conditions. While optical satellites like Sentinel-2 collect imagery in the visible, near-infrared and short-wave infrared parts of the electromagnetic spectrum, radar satellites use wavelengths at a centimeter to meter scale. Therefore, they are able to penetrate through clouds and other weather phenomena. The different wavelengths of SAR are often referred to as bands, with letter classifications such as X, C, L, and P. The frequency of these bands are illustrated in Fig. 4.

[7]https://earthdata.nasa.gov/learn/backgrounders/what-is-sar

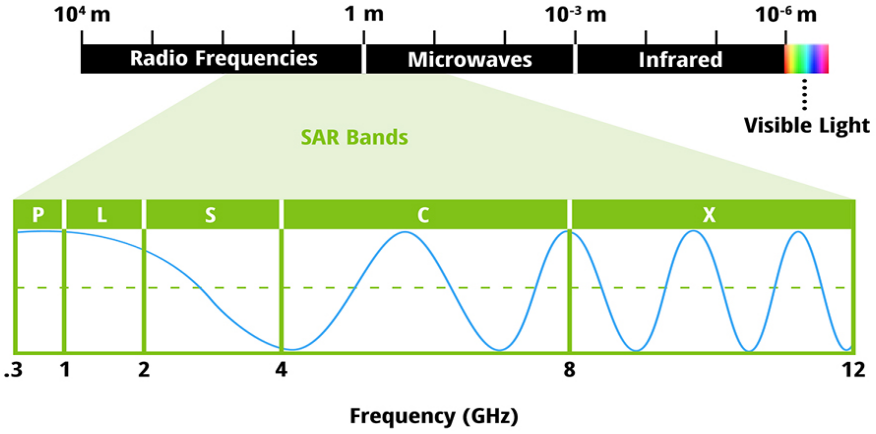


Figure 4: The electromagnetic spectrum with the SAR bands (X, C, L, and P) displayed in green.

Source: NASA

2.3.1 Backscattering and Amplitude

The SAR measurements contain amplitude and phase information. Amplitude of SAR data represents the strength of backscattered signal in terms of surface geometry, surface roughness and dielectric property of the terrain [21]. Moreover, ESA defines backscattering as a measure of the reflective strength of a target. It is the portion of the outgoing radar signal that the target redirects directly back towards the radar antenna[8]. The normalised measure of the returning signal from a target is called the backscattering coefficient, σ_0, and is the parameter that will be used as input for the deep neural network in this project. The formula for the backscattering coefficient, σ^0, is presented in Equation 1.

$$\sigma^0 = \frac{4\pi R^2}{\Delta A} \frac{P_s}{P_i} \tag{1}$$

In equation 1, ΔA represents the area of the illuminated surface, R is the distance from the radar satellite to the target, P_s is the power scattered by the target and P_i represents the initial power of the emitted signal.

Moreover, the distance to a target can be calculated by measuring the recorded phase information provided in the SAR product. However, the SAR products utilized in this thesis does not contain the phase information.

2.3.2 Polarization

Another aspect related to SAR is polarization. The radars have different polarization capabilities depending on the ability to emit and receive waves in different directions; horizontal (H) and vertical (V) (see Fig. 5). There are three different types; single-, dual- and multi-polarization. Single polarization SAR systems emit waves in one direction and can receive in the same, resulting in horizontal-horizontal (HH) or vertical-vertical (VV) imagery. Dual-polarization is a combination of co-polarization (HH or VV) and cross-polarization (HV or VH). In the case of dual-polarization the system is able to emit waves in one direction but receive in both, resulting in either HH and HV or VV and VH imagery. Multi-polarization systems are able to alternately emit waves in each direction while receiving waves in both directions, creating imagery of HH, HV, VV and VH. By

[8]https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/definitions

emitting and receiving signals with different polarization, one is able to capture a large spectre of detailed information about the observed surface.

HH-polarization is to prefer over VV when dealing with maritime monitoring [22]. Ocean clutter, characterized as disturbing radar-echoes from the sea waves, can make the ocean surface appear similar to sea ice in a radar sensor. Given that HH-polarization is better at suppressing the ocean clutter than VV, it is to prefer when discriminating between ice and water. However, even after choosing HH over VV, open water can still appear as bright as sea ice in co-polarized images, whereas in cross-polarized images (HV or VH), the rough sea surface remains darker [22]. Moreover, the cross-polarized signal is less sensitive to variations due to differences in the incidence angle [18]. However, cross-polarization can be limited by the low backscattering from new thin ice. Hence, the combination of HH and HV in dual polarization is an optimal choice for ice/water discrimination [23].

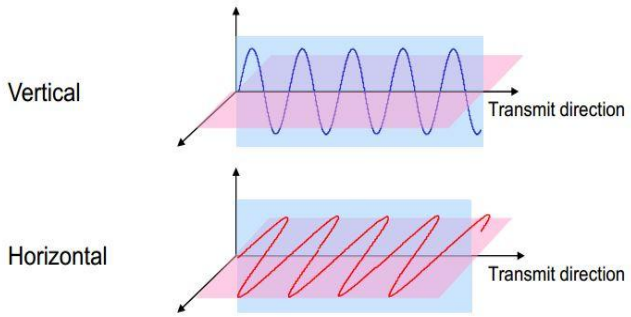


Figure 5: Waves transmitted in vertical and horizontal direction

2.3.3 Multilooking

SAR imagery is often subject to a "salt and pepper"-like noise effect called speckle, causing the image to look coarse-grained. Speckle is a result of random constructive and destructive interference between the multiple backscatters in the signal for each pixel cell. This is due to the fact that every surface type has several backscatter characteristics, enabling the same elements to look different in adjacent pixels. The effect is either constructive or destructive depending on the relative difference in the phases of the backscatters. Speckle can make images difficult to interpret and should therefore be reduced. In multilooking the radar beam is divided into several sub-beams, creating multiple looks of the images. These looks are averaged in order to reduce the amount of speckle, but it comes with a cost of deteriorating the image resolution.

2.3.4 Sentinel Missions and Sentinel-1

Related to the Copernicus Earth Observation program, a joint initiative between the European Commission (EC)[9] and ESA, a new family of satellites, called the Sentinels, was developed. Currently, there are three fully operational twin-satellite constellations in orbit, Sentinel-1, Sentinel-2 and Sentinel-3. These satellites will operate specifically for the purposes of the Copernicus program, including the supply of highly accurate, updated and freely accessible data for the public. Specifically, the Sentinel-1 constellation provides all-weather, night and day radar images. The Sentinel-2 constellation delivers high resolution optical images, and the Sentinel-3 constellation supplies data for land and ocean services. Following is a more detailed description of the two Sentinel-1 satellites which have produced some of the data utilized in this thesis.

In April 2014, ESA launched the first of two polar-orbiting satellites for the first mission of the Copernicus initiative, Sentinel-1A. The second satellite Sentinel-1B, was launched two years later in

[9] https://ec.europa.eu/info/index_en

April 2016, completing the Sentinel-1 mission. The two operational satellites carry a C-band radar instrument, which is an active sensor operating at wavelengths between 3.8 and 7.5 centimeters with dual polarisation capabilities. Furthermore, they are able to provide data with a resolution down to 5m, and a coverage up to 400km, regardless of cloud cover and light conditions. Moreover, each satellite uses 12 days to cover the entire surface of the Earth, making the revisit time of the mission 6 days. These characteristics enable the satellites to fulfill their main objectives, including land monitoring, supporting mapping of natural disasters, sea ice monitoring and climate change monitoring.

2.3.5 Level-1 Ground Range Distance Extra Wide Swath Mode (S1 L1 GRD EW) Products

The Level-1 Ground Range Distance Extra Wide swath mode (S1 L1 GRD EW) products are SAR data that has been multilooked (as described in section 2.3.3) before projected to ground range using the Earth ellipsoid model WGS84. Multilooking is applied on each burst before they are compounded to a single, contiguous, ground range image for each polarization (HH, VV, HV and VH). Only the amplitude information is kept in these products, i.e. phase information is removed. The result has a squared pixel spacing with reduced speckle and coarser resolution.

Among the variations of GRD-products, images provided in the ASIP dataset utilized are acquired with Medium resolution, Extra Wide (EW) swath (Level-1 GRDM EW). These consist of five sub-swaths with a spatial resolution of 93x87 m and a pixel spacing of 40x40 m. EW is a common acquisition mode for maritime monitoring due to its wide area coverage of 400km.

The L1 EW GRD products are acquired using the Terrain Observation with Progressive Scans SAR (TOPSAR) technique. This imaging approach utilizes swaths and bursts in order to obtain a high resolution in addition to a large area coverage. Contained in an EW product are five independently acquired sub-swaths merged together. There exists radiometric variations among the five sub-swaths. I.e. there is an inconsistent amount of additive noise included in the amplitude of all the sub-swaths.

2.3.6 Radiometric Distortions in S1 L1 GRD EW

In the context of SAR imaging, noise is referred to as any unwanted signal competing with the desired signal. There are two primary noise components; system dependent additive noise and signal dependent noise in which can be both additive and multiplicative. Additive noise is extra signal power which is added to the original signal. This noise is easier to cope with than multiplicative noise where the noise is multiplied to the original signal.

The signal-to-noise ratio (SNR or S/N), measures the amount of additive background noise compared to the amount of desired signal, and is given in the unit of Decibel (dB).

$$SNR = \frac{P_{Signal}}{P_{Noise}} \tag{2}$$

As can be observed in Equation 2, there is more signal than noise if the SNR ratio is higher than 1:1 (i.e. higher than 0 dB). Signal dependent noise, such as speckle, arise from imperfections in the SAR system and depend on the strength of the signal itself.

This sub-section will focus on the challenges associated with the usage of S1 GRD EW images acquired utilizing the TOPSAR technique.

Noise Equivalent Sigma Naught/Zero (NESZ)

To detect objects in a SAR image, the intensity of the signal reflected by the target object has to be larger than the thermal noise generated by the SAR system itself. The noise equivalent sigma naught (NESZ) is a commonly used metric for capturing the impact of additive system noise on the quality of the SAR image. According to ESA [10], NESZ is the backscatter coeficcient, σ_0, of the thermal noise in the SAR imagery. All signals with a lower backscatter intensity than NESZ will be indistinguishable from noise. Hence, a smaller NESZ is to prefer.

Thermal Noise in Cross-Polarized SAR Images

In contrast to co-polarized images where thermal noise is hardly noticeable, the intensity of the depolarized signals in cross-polarized images are weaker, and hence more sensitive to thermal noise. Specifically, the problem occurs when utilizing cross-polarized SAR images for imaging polar regions. Since the backscattering coefficient of thin sea ice is very close to the residual noise in open water, open water may be mistaken for being thin ice and vice versa [18].

Due to the recent discovery of the advantages it brings in several SAR-intensity-based applications, the usage of cross-polarization channels has increased. This introduces a demand for precise corrections for thermal noise.

ESA's Vectors for Noise Correction

In 2015, ESA constructed noise vectors designed to support the denoising of GRD-products. These were included in the Sentinel-1 Standard Archive Format for Europe (SAFE)[11], and released with the Sentinel-1 Instrument Processing Facility (IPF) version 2.5[12]. The vectors contain information about the noise floor in each image. Park et al. [19] applied the ESA correction on a HV-polarized SAR image, the improved image quality can be observed in Fig. 6. Even though there is an evident enhancement in the image quality after having applied the ESA-provided vectors, there is still a prominent intensity difference between the two leftmost sub-swaths. Moreover, a banding effect which highlight the boundaries of the noise floor, as discontinuous sharp changes of intensity, is still present [18]. In addition to noise in the range direction, there is also periodical intensity variations in the azimuth direction. This is due to a burst-wise acquisition mode and commonly referred to as scalloping.

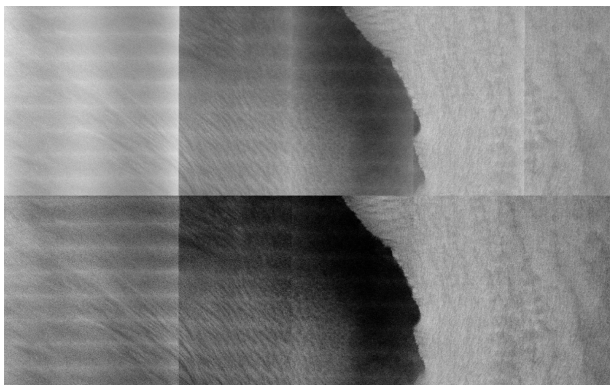


Figure 6: Upper image: Original S1 EW HV-polarized image. Lower image: Corresponding SAR scene corrected with the ESA provided noise vectors. (Scene ID: S1A_EW_GRDM_1SDH_20151218T065121_20151218T065151_ 009093_00D103_EB28).

Source: Park et al. [19]

[10] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/definitions
[11] https://earth.esa.int/SAFE/
[12] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/document-library

TOPSAR Technique and Scalloping

The EW GRD products are acquired by employing the TOPSAR technique, which is a further development of ScanSAR. In order to achieve a large area-coverage without compromising the spatial resolution, the ScanSAR technique acquires EW images using bursts and sub-swaths. With the antenna switching between different sub-swaths in the range direction while the satellite is moving forward, a wide swath coverage is achieved. In each sub-swath, the antenna acquires a certain number of echoed signals in a burst before it switches to another sub-swath to create a new burst. Since the echoed signals are stronger in the edges of the burst than in its center, a variable burst-wise intensity, commonly referred to as scalloping, is introduced. The TOPSAR acquisition technique was proposed with the intention of reducing this scalloping effect. By moving the antenna forward and backward in the azimuth direction for each burst in addition to switching between sub-swaths, the image quality is further improved. Notwithstanding the fact that TOPSAR was developed to reduce scalloping in S1 EW GRD products, the phenomenon is still present. It is most prominent in cross-polarized images over areas with low SNR such as sea surfaces.

2.3.7 Nansen Remote Sensing Center's Correction for Thermal and Textural Noise

After having applied the TOPSAR technique and ESA's noise correcting vectors, there are still noise challenges in both range and azimuth direction. Nansen Remote Sensing Center's (NERSC) approach for addressing the remaining noise problems will be described in this subsection. In 2018, Park et al. presented a three-step approach for addressing the challenges associated with scalloping and noise floor in [19]. Building further on ESA's noise corrections, they applied: 1) azimuth de-scalloping, 2) noise power scaling and interswath power balancing, 3) local residual noise power compensation. This workflow is presented in Fig. 7. Since each sub-swath in an EW image was acquired using different system parameters and processed separately before being merged together, they were separated into a pre-merge state for proper noise elimination.

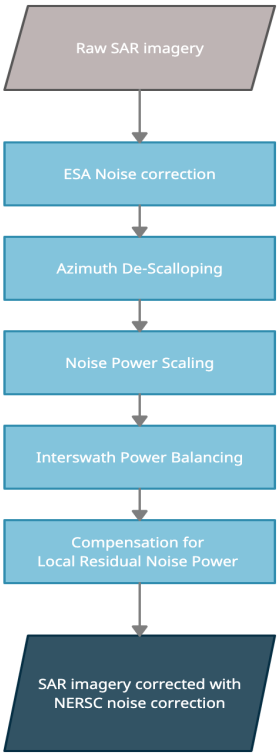


Figure 7: The work flow of the noise correction method applied by NERSC on the SAR images

Due to the fact that azimuth de-scalloping is connected to system parameters, this step is executed first. The center and inter-burst boundaries of each burst in terms of time (referred to as burst time in [19]) in addition to the antenna steering angle is calculated. The antenna steering angle is a function of the focused azimuth burst time, and is zero at the center of the burst. Moreover, the azimuth scalloping, which is a function of the antenna steering angle, is found before the de-scalloping gain for each sub-swath can be computed. The formula for calculating the de-scalloping gain is presented in Equation 3.

$$G_{ds,ssw} = \frac{1}{G_{AAEP}(\Psi(t_{burst}))} \tag{3}$$

I.e. the de-scalloping gain, G_{AAEP}, for each sub-swath is a function of the antenna steering angle, $\Psi(t_{burst})$, in which can be modelled by a two-way azimuth antenna element pattern (AAEP). This de-scalloping gain is then added to the noise field.

Thereafter, the ESA-vectors are modified for obtaining noise power scaling and interswath power balancing. This step is necessary since the noise power in the ESA vectors does not always correspond to the actual noise pattern in the image. In this case, the noise can be under- or overcompensated, and result in amplified noise distortions. Hence, a proper scaling factor was found by minimizing the residual sum of squares of the weighted linear fit of the denoised signal, as given in Equation 4.

$$RSS(k) = \sum_{i=1}^{N} \omega_i(\hat{s}_i(k) - s_i(k)) \tag{4}$$

where

$$s(k) = \sigma^0_{SN} - k * G_{DS} * \sigma^0_N \tag{5}$$

In Equation 5, $s(k)$ is given as the difference between σ^0_{SN} (the raw sigma nought before noise subtraction) and σ^0_N (the NESZ computed from the ESA-provided thermal noise vector), scaled with a factor of k. G_{DS} is the de-scalloping gain computed in the precedent azimuth de-scalloping step. $\hat{s}(k)$ is the linear fit for the denoised signal $s(k)$. ω is the weight factor, N is the number of pixels in the range direction in a sub-swath and i is a running index from 1 to N. For each sub-swath, the scaling factor providing the smallest RSS, and thereby the optimal scaling factor, is chosen.

The scaling factors are applied to the ESA-vectors and it is found that the parts most affected by noise are located close to the edges of the sub-swath. The difference in signal power between the edges of neighbouring sub-swaths are estimated and compensated for in order to create a continuous merged image.

By combining the three aforementioned noise modifications, a noise field was constructed and improved the image quality significantly. When subtracting this noise field from the raw signal, there may arise negative values. These were handled simply by giving them the value zero, in which may cause problems in later processing steps. Hence, the local residual noise power caused by the zero-clipping of negative-valued pixels was compensated for by adopting the local SNR-dependent radiometric correction, as presented in [24].

Even though the thermal noise correction greatly improved the performance in SAR intensity based applications, there was still multiplicative noise in the inter-swath boundaries. Hence, NERSC proposed an additional approach in [20], referred to as textural noise correction. The local variations were re-scaled using a sliding window and applying the noise equivalent standard deviation (NESD) and signal-plus-noise-to-noise ratio (SNNR). Thereby they introduced the coefficients for the NESD model. This approach was added to the thermal noise correction, presented in [19], and applied on the images utilized in this thesis.

2.4 Deep Learning

Deep learning is a subsection of artificial intelligence in which includes machine learning approaches that mimic how the human brain processes information. These approaches are able to produce high-level features from raw data, and are therefore commonly applied in image-, video- and sound processing. For instance, the use of deep learning in object detection allows labeled objects to be produced from a raw image. Deep learning algorithms are generally more complex than traditional machine learning algorithms like Decision Trees[13] or Naive Bayes[14], however, they are able to achieve a better performance when provided a sufficient amount of data. Additionally, traditional machine learning algorithms require the extraction of manually designed features from the input data, while deep learning algorithms can automatically extract the important features. This way, less manual labor and domain expertise is required for deep learning. The described distinction between machine learning and deep learning is visualized in in Fig. 8.

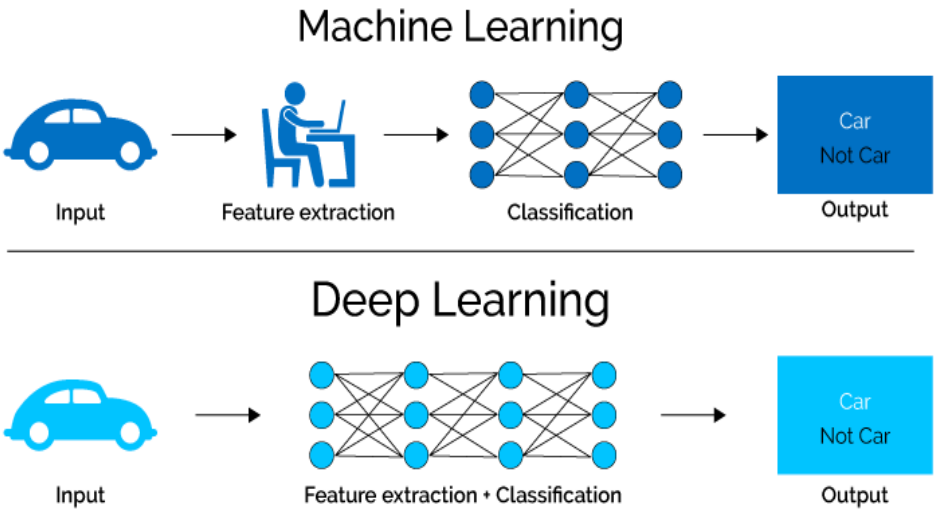


Figure 8: Machine Learning versus Deep Learning

Deep learning has experienced an increased interest in the last decade, mainly due to two contributing factors. Firstly, in today's society incredible amounts of data are produced every second, in fact humans produces 2.5 quintillion ($2.5 * 10^{18}$) bytes of data every day[15]. Additionally, an increasing proportion of this data is processed into datasets and made available to the public. Deep learning is able to achieve impressive results with a large amount of data, hence, the use of deep learning increases with the increased amount of available data. Secondly, computational power is constantly improving and comes at a lower cost today than it did in the past. Deep learning requires large processing powers in order to process the data in a limited time span. More specifically, the production of Graphical Processing Units (GPUs) has enabled faster execution of deep learning methods due to their parallel capabilities.

A deep neural network is similar to a complex function with a large number of parameters. The main objective is to adjust these parameters to obtain an optimal function that minimizes the loss function for the given input data. A neural network is the combination of several consecutive layers with many neurons. The parameters are adjusted during the model training phase. The rest of this section will give a description of these aspects.

[13] https://en.wikipedia.org/wiki/Decision$_tree$
[14] https://en.wikipedia.org/wiki/Naive$_B ayes_c lassifier$
[15] https://techjury.net/blog/how-much-data-is-created-every-day/gref

2.4.1 Layers

Architecturally, there are mainly three types of layers; input-, hidden- and output-layers. These can be viewed in Fig. 9. The input layer is the first layer of the network, and it receives the data as input from the user. The hidden layers are the intermediate layers, which got their name due to the fact that they are not "visible" to the user. I.e. the user does not directly interact with these layers. The output layer computes the result and is the last layer of the neural network. Deep neural network architectures are characterized as having several hidden layers whereas a conventional neural network has one or two. A layer is the highest-level component in a deep learning network and each layer consists of many nodes, called neurons. These neurons are connected to other neurons in both the preceding and following layer.

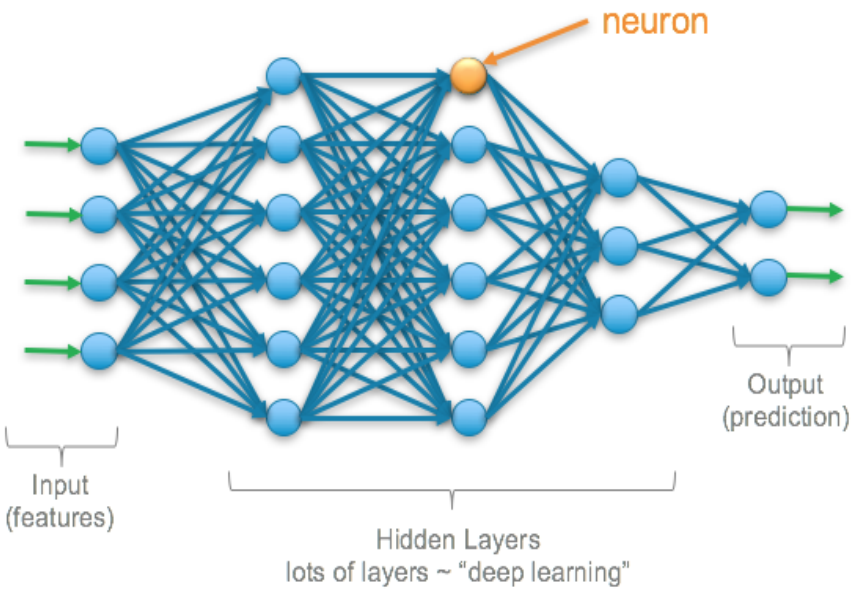


Figure 9: Visualization of a simple deep neural network

2.4.2 Artificial Neurons

Neurons in deep neural networks are mathematical functions which transform the input in order to produce the output. As shown in Fig. 10, an artificial neuron receives an input, x_i, and a weight, w_i, from the preceding layer. The weighted sum of the input and the weights is calculated as shown in the figure. This value is further passed through an activation function to produce the output from the entire neuron. During the training of an artificial neural network, the model adjusts the weights in the neurons of the network to find the combination that minimizes the loss value.

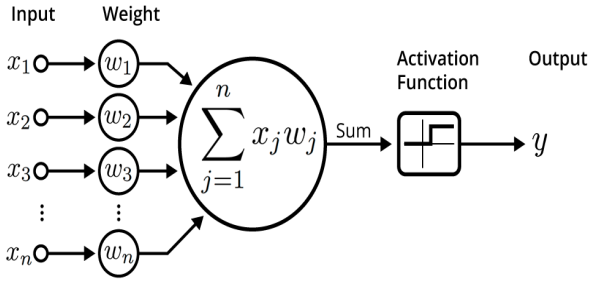


Figure 10: Visualization of an artificial neuron

2.4.3 Activation Functions

The activation function is the part of the neuron that provides the neural network with capabilities to learn a non-linear complex function. This function is applied in the neurons like demonstrated in Fig. 10. As the name suggests, it is the function that decides if the output of the neurons is active or not. Among the different variations of activation functions are Sigmoid, Softmax, ReLU and LeakyReLU, the most common ones. These are presented in 6, 7, 8 and 9, respectively.

$$f(x) = \frac{1}{1 - e^{(-x)}} \tag{6}$$

The Sigmoid activation function, presented in Equation 6, maps real-valued input to the range between 0 and 1. Due to the small range of possible values, it has a tendency to loose a large amount of information as the depth of the neural networks increases.

$$m(x_i) = \frac{e^{(x_i)}}{\sum_j x_j} \tag{7}$$

Similarly to the Sigmoid function, the Softmax activation function, given in Equation 7, maps the input values to the range between 0 and 1. Softmax, as opposed to Sigmoid, ensures that the total sum of the output is equal to 1 making it a probability distribution. In this manner, the Softmax can be applied to multi-class classification problems, whereas Sigmoid is mainly suitable for binary classification.

$$h(x) = \mathbf{max}(0, x) \tag{8}$$

$$l(x) = \mathbf{max}(\alpha x, x) \tag{9}$$

Rectified Linear Units, also known as ReLUs, have since 2017 been the most popular activation functions used in Deep Learning[16]. As can be viewed in Equation 8, the maximum operator is applied on the input value x and 0, resulting in only positive output values. Having a much larger range of possible values, it does not struggle with the same problem as the Sigmoid function. Additionally, due to the simplicity of the operator, it requires less computational power. However, it does not produce negative values in which introduce a problem known as dying ReLUs. In this case, several neurons will be characterized as dead since they are not producing any active output. To address this issue there exists a modified ReLU, called LeakyReLU in which is presented in Equation 9. This activation function "leaks" negative values to the output by reducing them with the factor α. Thereby, the range of the function is extended.

2.4.4 Loss Functions

In deep learning, a model attempts to predict values that are as close to the true target values as possible. The loss function of a deep neural network is a measure of how much the predicted values differ from the target values. In this context, a perfect performance would indicate that the model is able to capture and learn all the details in the dataset, an thereby always make the correct prediction. There are a vast variety of different loss functions in which the most popular ones include Mean Squared Error and Cross-Entropy Loss.

The Mean Squared Error Loss is calculated as the mean squared difference between the true values and the values predicted by the model. The mathematical formula is given in Equation 10, where y_i is the true label for the example, $\hat{y_i}$ is the predicted label and N is the number of

[16] https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

examples.

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}{N} \tag{10}$$

The Mean Squared Error exclusively considers the difference between the correct label and the predicted label. The Cross-Entropy Loss, on the other hand, is able to incorporate the model's confidence in each prediction. Thereby, a model is harder penalized for a wrongly predicted label if it is very confident in that prediction. The Cross-Entropy Loss value decreases as the probability of the predicted value belonging to the true class increases. The probability that a specific example belongs to a certain class is given as a value between 0 and 1. The equation for calculating the cross entropy loss for a binary classification problem is given in equation 11. There are only two classes, $y = 0$ and $y = 1$, in which y is the true label for the record and p is the probability of the predicted class being 1.

$$CE(p, y) = \begin{cases} -\log(p) & \text{if y = 1} \\ -\log(1-p) & \text{otherwise.} \end{cases} \tag{11}$$

For instance, if p is 0.8, the current pixel has a probability of 0.8 for being classified as 1, and 0.2 for being classified as 0. To extend the binary cross entropy loss to handle multi-class problems, one would simply calculate the binary cross entropy loss for each of the classes and sum it.

2.4.5 Optimization Techniques

Optimizers are applied in the deep neural network to solve the optimization problem mentioned in the beginning of this chapter. Their purpose is to find the combination of parameters that minimizes the loss value. Different versions of the Gradient Descent[17] algorithm is commonly applied as optimizers. By adjusting the model parameters in an iterative manner, the Gradient Descent finds a local minimum of the loss function. The direction in which to update the parameters is determined by the opposite direction of the largest gradient in a given point on the loss function. And the size of the update is decided by the learning rate. The opposite direction of the largest gradient at a given point is the direction that decreases the value of the function most rapidly. A challenge related to using gradient based learning is that the gradient in a local minimum is similar to the gradient in a global minimum. Therefore, the optimizer runs the risk of getting stuck in a local minimum, and thereby never reach the global minimum. This challenge is visualized in Fig. 11.

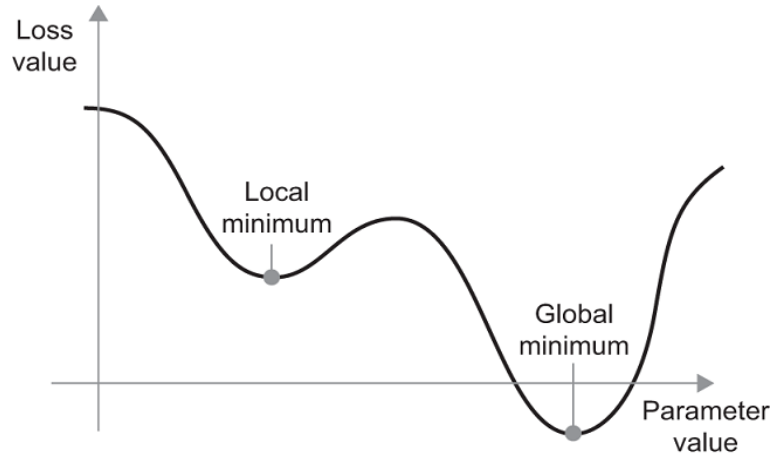


Figure 11: A loss function with global and local minimum for different parameter values

Gradient decent in its simplest form includes all the examples of the input in the gradient computation before updating the weights of the network. It is therefore computationally expensive. The

[17] https://en.wikipedia.org/wiki/Gradient_descent

most basic improvement of the gradient decent optimizer is called stochastic gradient decent(SGD). SGD reduces the computational cost of the basic gradient decent by only including a random subset of the data points during gradient computation. This reduces the computational load, however, the optimizer also gets more vulnerable to get stuck in a local minimum. Additionally, since the optimizer applies a random subset of the records in the dataset the loss values for each iteration may vary to a high degree.

Adam (Adaptive Moment Estimation) is another, slightly more sophisticated optimizer. It adopts the computational benefit included in the SGD, however, it maintains a separate adaptive learning rate for each of the parameters in the function. I.e. it learns how large each parameter-update should be for each iteration.

2.4.6 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is a class of deep neural networks, and is specialised for capturing spatial dependencies in multidimensional data. Spatial dependencies includes the arrangement of the values and the relationships between adjacent values in the input data. This capability makes CNNs highly applicable for image and video processing. A CNN would be the preferable type of deep neural network to apply in the image segmentation problem visualized in Fig. 8.

By sequentially exploiting convolution, the CNN is able to extract features from raw input images. Convolution is a mathematical operation where two functions are combined into one. The resulting function expresses the effect of the first function on the other. The mathematical expression of convolution is given as the integral of the product of two functions f and g, where one of the functions is shifted with a size τ. The operator is denoted $\star$ and displayed in Equation 12.

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\ d\tau \tag{12}$$

CNNs mainly consist of two types of layers, **convolutional layers** and **pooling layers**. The three dimensional (height, width, depth) input data is passed through several of these layers. The output from the layers tend to decrease in height and width, while the depth increases. The layers in the beginning of the network normally extract low-level features such as edges, colors and intensity.

A convolutional layer consists of several filters containing adjustable weights which are initialized with random numbers. The filters are two dimensional and normally smaller than the width and height of the input to the layer. The filters will be convolved with the entire depth of the input, but only the width and height of the filter, by utilizing a sliding window as shown in Fig. 12.

For each area the filter covers in the input image, there will be one output value. This results in an output map called a feature map. Each output value in the feature map is the sum of an element-wise multiplication between every weight in the filter and its corresponding pixel in the input. In the output from the entire layer, there will be one feature map for each filter stacked in the depth dimension. In other words, if the input volume to a layer with 16 filters has a depth of 3, the output will have a depth of 16.

To relate the filters and feature maps from the convoluitonal layers to the notion of artificial neurons, the neurons in a convolutional layer receives the weights w_i and input values x_i from the filter and the layer-input, respectively. The neuron performs the dot product of the input values and weights, and an activation function is applied.

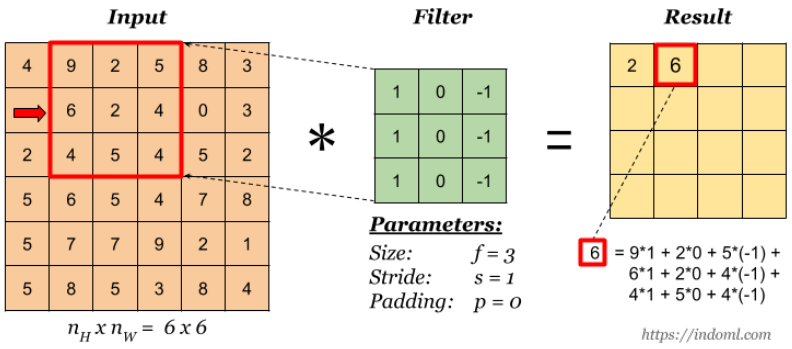


Figure 12: Convolution with input and filter, to produce result.

Source: indoml.com

The size of the output is controlled by four hyperparameters; the number of filters, stride, padding, and the size of the filter. The number of filters in a layer will, as mentioned, control the depth of the output from this layer. Stride controls the step size of the sliding window over the input layer. In Fig. 12 the stride is 1, meaning that the window moves one pixel for every iteration. This leads to a large overlap of values that are used to generate output values. Intuitively, the window starts from the upper left corner in the image and moves with the stride size to the next position. The pixels in the center of the input will be present in calculations more often than the pixels situated around the edges of the image. To enable all pixels in the input image to have the same effect on the output values, padding can be used. Padding appends artificial pixels around the edges so that every pixel can be treated as a central pixel. Padding can also be used to control the height and width of the output.

Pooling layers are used to reduce the complexity and level-of-detail of the input. Thus, the computational resources and time required to process the data are reduced. There are several different types of pooling layers in which the most common one is max pooling. In a max pooling layer, the maximum value in the pixels covered by a specified subarea is outputted as shown in Fig. 13.

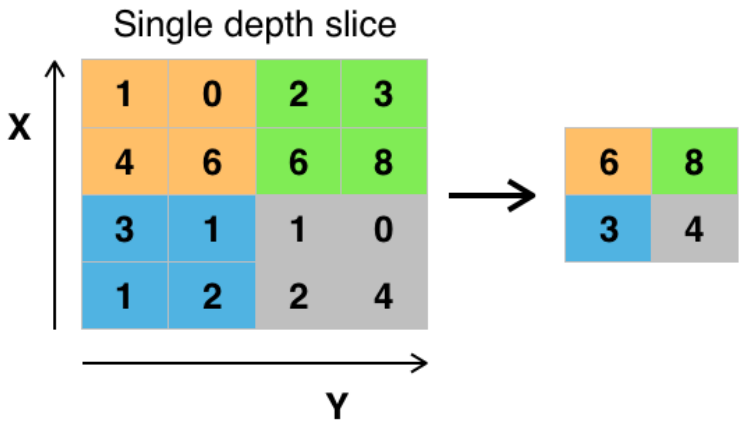


Figure 13: Max pooling performed on the input image to reduce the size

CNNs are commonly applied to image segmentation problems where the goal is to classify each pixel in the input image. In Fig. 14 a semantic segmentation has been performed on satellite imagery to produce a land cover map with classes such as Forest, Water, Grassland and Cultivated Land.

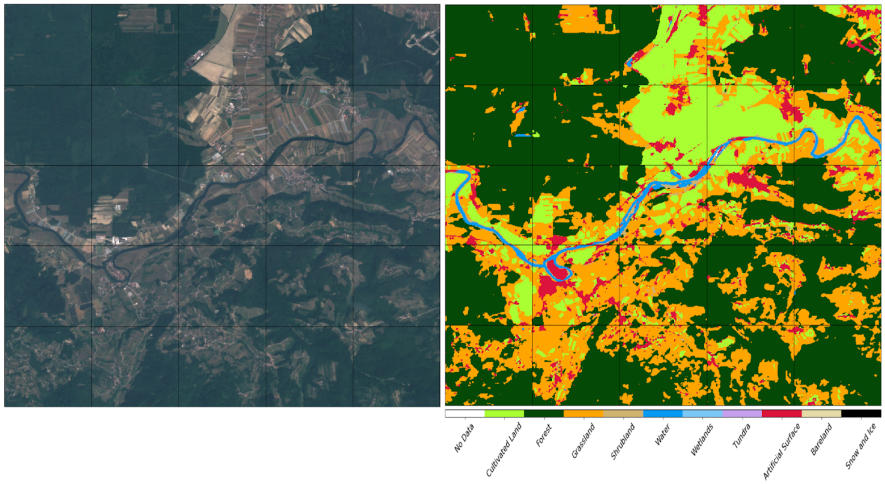


Figure 14: Max pooling performed on the input image to reduce the size

Source: Tracking a rapidly changing planet by Development Seed[18]

In semantic segmentation, the consecutive convolutional- and pooling layers in the CNN will reduce the size of the input image. Therefore, an up-sampling operation is required in order to obtain the original size for the output image. Transposed convolution is a common approach for performing up-samling in CNNs. An important aspect related to convolution is the spatial connectivity between the input- and output layers. More specifically, there is a relationship between the values in the corresponding areas in each layer. For instance the top-left corner of the output layer depends on the top left corner of the input layer. This relation has to be maintained in the up-sampling operation. Transposed convolution uses rearranged filters for maintaining the spatial relationship between the input and the up-sampled output. Additionally, the values in the rearranged filters are trainable in which allows the up-sampling to be uniquely suitable for the given situation.

2.4.7 U-Net

U-Net is a fully convolutional neural network for image segmentation originally developed for biomedical use at the University of Freiburg, presented in [16]. It uses pixel-wise classification to identify and localize the content in an image. As mentioned, CNNs consist of several convolutional layers in which tend to decrease in size. The architecture of the U-Net is composed of a contracting path, followed by an expansive path. This makes the architecture U-shaped as shown in Fig. 15. The contracting path captures the context in the input image, and the expansive path handles the precise localization ([16]). As can be observed in the figure, there are four blocks in the contracting path (left path). Each block consists of two convolutional layers with a filter size of (3x3) and is activated by a ReLU layer. The two convolutional layers are followed by a Max Pooling layer with a (2x2) filter. Moreover, the expansive path (right path) is similar to the contracting path, but instead of using max pooling to downscale, transposed convolution is used to upscale. Between the contraction and the expansion, there is a middle part including two convolutions with ReLU activation layers.

The grey vertical numbers in the Fig.15 are the width and height of the input to each layer. These sizes decrease for each block due to padding. The max pooling layers reduce the height and width by a factor of 2 due to the size of the filter. The horizontal grey numbers display the depth of each tile which is increased by convolution. The depth of the output layer in the neural network is equal to the number of desired class labels. For a binary case, the number would be two as it is in the original implementation of U-Net.

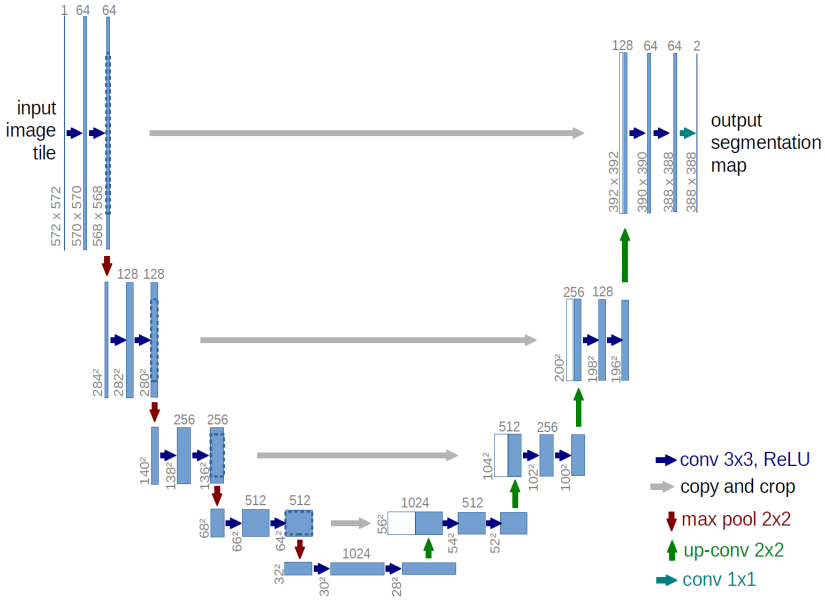


Figure 15: The general U-Net architecture

Source: Ronneberger et al. [16]

2.5 Performance Measurements

To determine whether a deep neural network performs well or not, it has to be evaluated with proper performance metrics for a specific task. There are two kinds of problems presented in this thesis; multiclass- and binary image segmentation, and these problems have to be evaluated separately. This sub-section will address the performance metrics applied to evaluate the models generated in this thesis.

2.5.1 Confusion Matrix

A common approach for evaluating supervised classifiers, such as the models presented in this thesis, is to apply a confusion matrix. This is a matrix that summarizes the number of correct and incorrect predictions for each class. As an example, the confusion matrix for a binary classifier is displayed in Fig. 16. The diagonal items, containing True Positive (TP) and True Negative (TN), represent the number of elements correctly classified by the model. False positive (FP) and False Negative (FN) represent the number of incorrect classified elements.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 16: Confusion matrix for a binary classification problem. True Positive (TP) and True Negative (TN), represent the amount of elements correctly classified by the model. False positive (FP) and False Negative (FN) represent the amount of incorrect classified elements.

A confusion matrix efficiently presents the performance of a classifier in a clear and understandable manner. Moreover, in multiclass classification, it is a good means for detecting which classes the model fails to predict, and where it is robust. If one of the elements in the diagonal has a low score (meaning low TP-rate for that class), it indicates that the model fails to predict this class. From the confusion matrix one can calculate several performance metrics, in which will be described in the rest of this section.

2.5.2 Accuracy, Precision, Recall and F1-score

Accuracy represents how close the predictions made by the model are to the true labels. As visualized in Equation 13, it is given as the ratio of correctly classified pixels to the total amount of pixels.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Precision, on the other hand, represent the proportion of predicted positives that are truly positive. The formula for calculating the precision is given in Equation 14.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

Recall is another useful metric that answers to the proportion of positives that is correctly classified. I.e. it gives the ratio of correctly classified positives to the amount of all true positives. The formula for recall is presented in Equation 15.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Precision and recall can be combined into a f1-score that aims at conveying the balance between precision and recall. The formula for the f1-score is presented in Equation 16.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{16}$$

One have to be careful when applying the accuracy metric and the f1-score to models trained on an imbalanced dataset. As accuracy is given as the proportion of correctly classified pixels, it is a natural starting point for model evaluation. However, if the dataset is highly imbalanced, this metric may lead to misinterpretations of how well the model performs. For instance, if 95% of the dataset belongs to a single class, then the model could achieve an accuracy of 95% if it predicted all pixels to be this class. Moreover, an f1-score could be even more biased. By taking the harmonic mean of precision and recall which both emphasize only the positive predictions, an f1-score does not consider the true negatives, and is thus misleading for highly imbalanced classes. In this case, correlation measures that are symmetric and evaluate both directions of predictability is to prefer[19]. However, if the dataset is balanced and all classes are equally important, the f1-metric is a good performance measure.

2.5.3 Receiver Operating Characteristic Curve (ROC curve)

It is difficult to determine the threshold between the two classes in a binary classification problem. A receiver operating characteristic curve, or ROC curve, can be applied in order to find a threshold that optimizes the performance. By varying the threshold from most restrictive to least restrictive, the false positive rate (FPR) and true positive rate (TPR) are plotted in a 2D chart. FPR, also

[19] https://en.wikipedia.org/wiki/F-score

referred to as the *probability of false alarm*, gives the probability of the model predicting a pixel to be positive while it actually is negative. TPR is the same as recall, it measures how many of the true positives that the model identifies. For each threshold there exists one FPR and one TPR and both can be calculated from the confusion matrix, as given in Equation 17 and 18.

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

$$TPR = \frac{TP}{FP + TN} \tag{18}$$

In Fig. 17, different ROC curves are illustrated. Given a model, the threshold that obtains the curve furthest away from the diagonal (random classifier)is is the optimal threshold. This curve is the one with the largest area under the curve (AUC), where the maximum area is 1. The ROC AUC score is a common performance measure for binary classifiers, and will be used to evaluate the model trained on the binary dataset in this thesis.

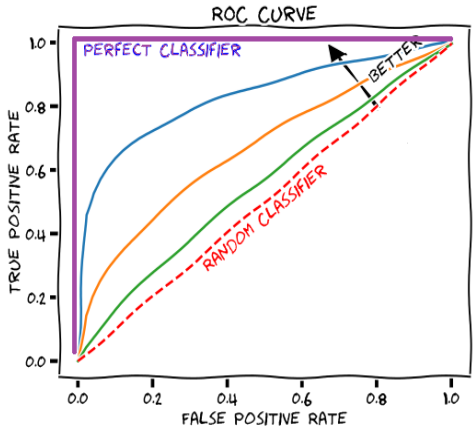


Figure 17: An illustration of how a ROC curve can look. If the classifier is perfect and the optimal threshold is chosen, the curve will look like the purple line. A ROC curve close to the curve of the random classifier (red staled line) indicates a bad threshold between the classes.

Source: *Measuring Performance: AUC (AUROC)* by Rachel Draelos[20]

2.5.4 R^2 Statistic

The R^2 statistic assesses how strong the linear relationship is between two variables in which one of them depend on the other. It is also called *the coefficient of determination*, and examines how changes in one variable can be explained by the changes in a second variable, when predicting the outcome of a given event [21].

Given a dataset of size N, there are N true values y_1, y_2, ..., y_N and N corresponding predicted values f_1, f_2, ..., f_N. The residual of each predicted value is denoted $e_i = y_i - f_i$. Moreover, the arithmetic mean of the true values is presented in Equation 19.

$$\overline{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{19}$$

[21] https://www.investopedia.com/terms/c/coefficient-of-determination.asp

The total sum of squares (SS_{tot}) and the residual sum of squares (SS_{res}) can then be calculated as in Equation 20 and 21, respectively.

$$SS_{tot} = \sum_{i} (y_i - \overline{y})^2 \tag{20}$$

$$SS_{res} = \sum_{i} {e_i}^2 = \sum_{i} (y_i - f_i)^2 \tag{21}$$

Having obtained SS_{tot} and SS_{res}, the R^2-statistic can be computed like presented in Equation 22.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{22}$$

The R^2 statistic is a common performance measure in regression problems, however, if the classes' level of measurement is ordinal, it can also be applied to discrete classification problems. A high R^2 statistic indicates a high degree of linear correlation between the true values, y_1, y_2, ..., y_N, and the predicted values, f_1, f_2, ..., f_N, which is a good measure of the performance.

2.5.5 Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) is a statistical rate that produces a high score only if the model obtained good results in all four confusion matrix categories (TP, TN, FP and FN). The score is measured proportionally both to the size of positive elements and the size of negative elements in the dataset [25].

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{23}$$

For binary classification models trained on a an imbalanced dataset, the MCC measure is a good alternative. Since this metric considers all the elements in the confusion matrix, it is viewed as a balanced measure that can be used even if the dataset is highly imbalanced [26]. However, this is also a good performance measure for models trained on a balanced dataset. The equation for calculating MCC can be observed in Equation 23.

(This page is intentionally left blank)

Chapter 3

Study Area and Data

This section will address the ASIP dataset[1] which was utilized for training and validation of the U-Net in this thesis. The ASIP dataset was published by the AI4Arctic sea ice project[2], and includes 461 S1 SAR scenes from 2018 to 2019, matched with sea ice charts from the Danish Meteorological Institute (DMI). With a one to one correspondence between the pixels in the S1 imagery and the ice charts, this dataset is well suited for deep learning.

The AI4Arctic sea ice use case was carried out in a collaboration between DMI, the Technical University of Denmark (DTU) and Nansen Environmental Remote Sensing Center (NERSC). By exploring the potential of deep learning in earth observation applications within Arctic areas, their objective is to replace the labour intensive manual generation of ice charts with an automated approach conducted on Sentinel-1 SAR imagery.

For each SAR scene there is, in the ASIP dataset, a NetCDF file containing all relevant information associated to the specific scene. As can be observed in table 2, there is, in addition to S1 imagery, an ice chart and data acquired with Advanced Microwave Scanning Radiometer 2 (AMSR2).

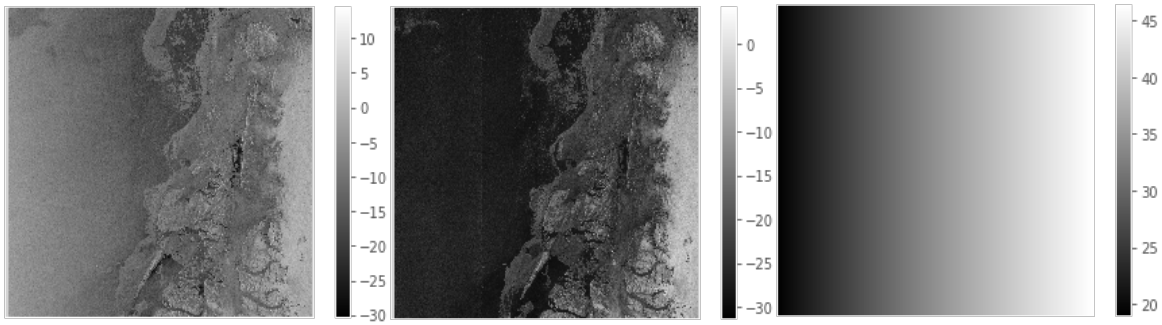
Table 2: Content of a NetCDF file

Content of a NetCDF file :
Dual polarized(HH and HV) Sentinel-1 Extra Wide Swath (EW) with **ESA** noise correction
Dual polarized(HH and HV) Sentinel-1 Extra Wide Swath (EW) with **ESA** and **NERSC** noise correction
Auxiliary Sentinel-1 image parameters
Microwave radiometer measurements from the AMSR2 sensor on board JAXA GCOM-W
Corresponding DMI ice chart based on that Sentinel-1 image

Contained in the file are dual polarized Sentinel-1 images with two different types of noise corrections. In this thesis, only the imagery with both ESA and NERSC correction will be utilized in addition to the incidence angles as input to the U-Net. The three input products are visualized in Fig. 18.

The AMSR2 data consists of microwave radiometer measurements from the AMSR2 sensor on board the JAXA GCOM-W satellite in which is created for observing the water cycle in conjunction with Earth environmental changes. By providing brightness temperature measurements, AMSR2 is intended to compliment the learning of the of sea ice concentrations. However, in this thesis,

[1] https://asip.dk/onewebmedia/Kreiner$_P hiweek - sideevent_0$1.10.2020.*pdf*
[2] https://orbit.dtu.dk/en/projects/ai-for-the-arctic



NERSC SAR HH **NERSC SAR HV** **Incidence angles**

Figure 18: Display of the ESA and NERSC corrected SAR imagery and the incidence angles

only the S1 imagery has been applied. The following subsections will address, the study area and the data distribution, followed by a description of the S1 SAR imagery and ice charts utilized in this thesis. Finally, the challenges associated to the ASIP data set will be introduced.

3.1 Study Area and Data Distribution

The SAR scenes and their corresponding ice charts are covering the waters surrounding Greenland from 59° to 82° North, and from 0° to 75° West. The Greenland Ice is one of the largest bodies of ice in the world, and is therefore playing a vital role in the global climate system. According to the European Environment Agency (EEA)[3], the cumulative ice loss from Greenland from 1992 to 2017 was 3 900 billion tonnes, which contributed to approximately 11 mm of the global sea level rise. This important and increasingly rapid change makes Greenland an interesting study area. The SAR scenes utilized in this project were acquired between March 2018 and May 2019, and are separated into 9 different regions surrounding the coast of Greenland. The regions (given in the ASIP-v2 User Manual[4]) are presented in Fig. 19.

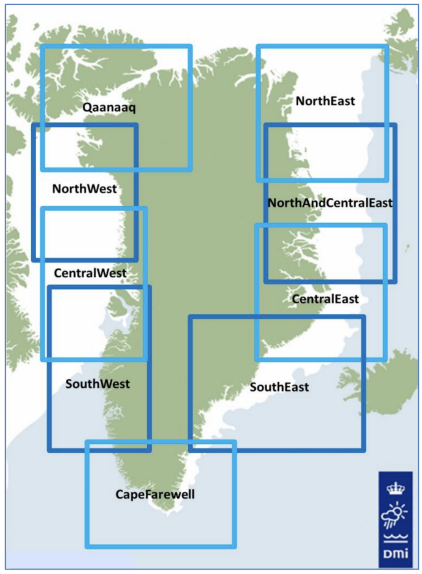


Figure 19: The 8 coastal regions presented in an overview map of Greenland.

Source: ASIP-v2 User Manual

[3]https://www.eea.europa.eu/data-and-maps/indicators/greenland-ice-sheet-4/assessment
[4]https://data.dtu.dk/articles/dataset/AI4Arctic_ASIP_Sea_Ice_Dataset_-_version_2/13011134?file=24951176

It is noteworthy that the process of downloading the ASIP dataset was computationally exhaustive for the server utilized in this thesis. After several attempts, the download was complete, however, only 240 SAR scenes were included in the download. This is a considerable drawback for the task of deep learning since the amount of data is critical for the performance of the network.

Another concern due to the loss of data was that the seasonal and regional distribution of the SAR scenes would not resemble the original distribution provided in the ASIP dataset. However, it turned out that even though an arbitrary amount of the dataset was downloaded, the distribution of the subset is sufficiently similar to the distribution of the original dataset. Fig. 20 and 21 are displaying the geographical distribution of the original ASIP dataset and the downloaded version of it containing 240 SAR scenes, respectively. As can be observed, the two distributions follow a similar pattern. From Fig. 21 it is clear that the tip of Greenland, Cape Farewell, is the best represented region, having 72 SAR scenes included in the subset. Qaanaaq followed by North And Central East and North East are the least represented regions with respectively 4, 7 and 8 scenes in the dataset. The remaining regions are represented by between 24 and 34 scenes each.

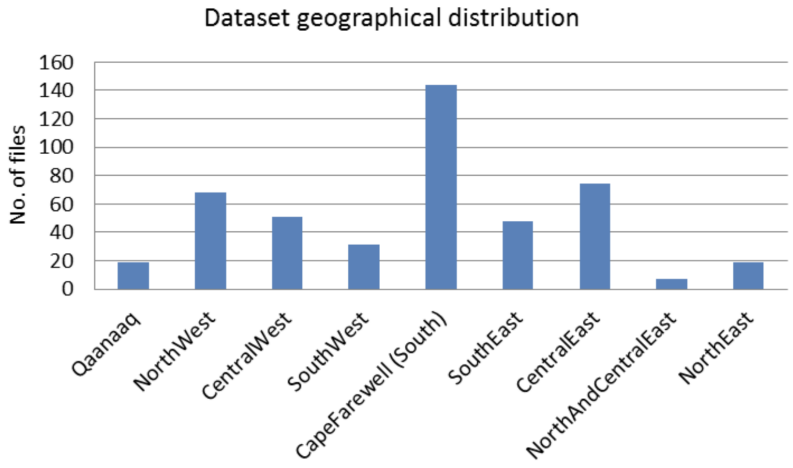


Figure 20: Geographical distribution of the SAR scenes in the original ASIP dataset

Source: ASID-v2 User Manual

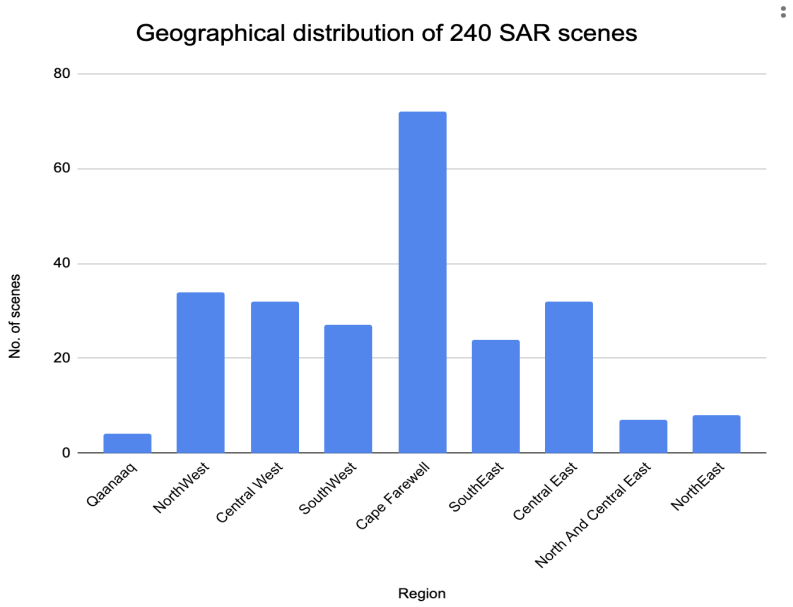


Figure 21: Geographical distribution of the SAR scenes in the downloaded subset of the ASIP dataset used in this project

The seasonal distribution along with the distribution of scenes over regions in the downloaded dataset can be observed in Fig. 22. According to the Norwegian Meteorological Institute[5], the summer melt season usually begins in March and continues until a minimum is reached in September. Hence, the seasons are split into intervals of three months each where February-April is the interval with the highest amount of sea ice, and August-October is the interval with the lowest amount. In order to train a robust neural network it is to prefer that there is an approximate even distribution of the different seasons in the dataset. Unfortunately, Fig. 22 reveals that there is an overweight of scenes in the two intervals between November and April. This is also the case for the original ASIP dataset, however, it is not as distinct as in the downloaded version. The dataset comes with an accompanying table, in which contains the percentage of ice, water and missing values pixels for each of the SAR-scenes. By inspecting this table it is evident that the dataset contains a large amount of water compared to ice. Therefore, the authors concluded that the seasonal imbalance is small enough for the training of the neural network. As adding scenes from the summer months containing more water than ice, would increase the class imbalance in the dataset.

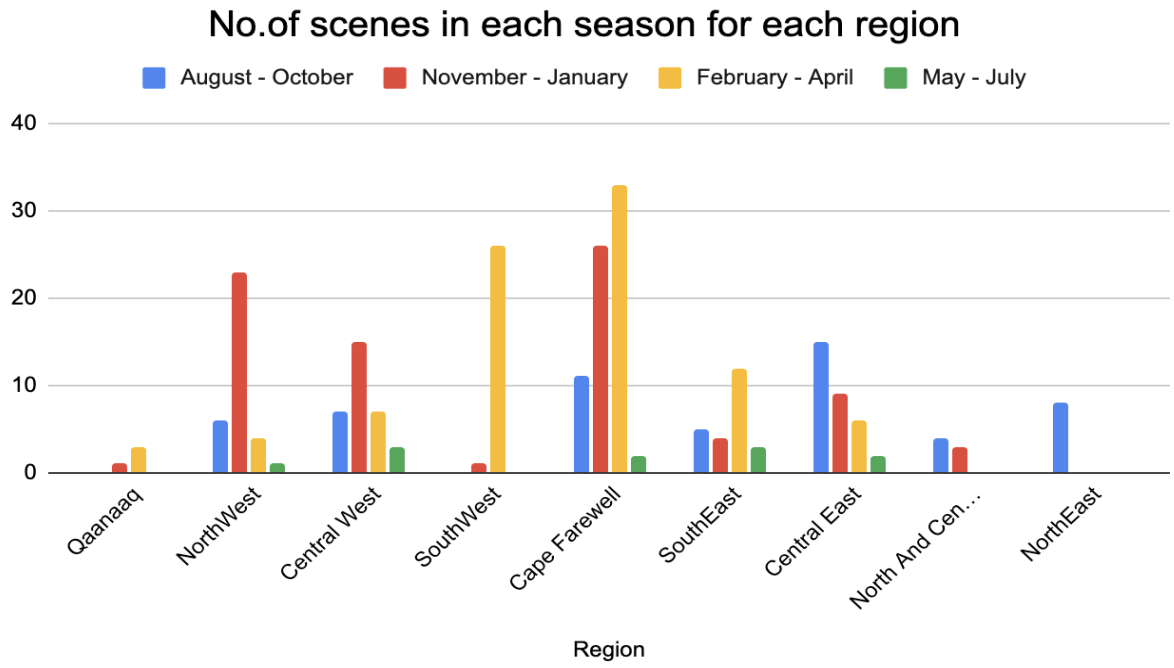


Figure 22: Seasonal and graphical distribution of the SAR scenes applied in this thesis

3.2 Sentinel-1 Imagery

The Sentinel-1 satellites acquire images in different modes with varying resolution and coverage. Each mode generate products at different SAR levels in which Level-1 Ground Range Distance Medium (GRDM) Extra Wide (EW) is the mode utilized for acquiring the images in the ASIP dataset.

As referenced in section 2.3.2, dual polarization is to prefer when discriminating sea ice from water. Therefore, the images in the ASIP dataset are acquired in dual polarization mode. The cross-polarized images (in dual polarization) comes with radiometric distortions as addressed in section 2.3.6. This is one of the greatest challenges one encounters when dealing with cross-polarized Sentinel-1 images in the polar sea. The images applied in this thesis have been de-noised by NERSC according to the noise correction presented in section 2.3.7.

[5]https://cryo.met.no/en/arctic-seaice-september-2020

3.3 Ice Charts

The ice charts included in the ASIP dataset are produced by the DMI operational sea ice service. They utilize S1 SAR imagery as their primary source for sea ice analysis. To support the analysis of the SAR imagery, DMI specify that auxiliary satellite data (optical imagery, thermal-infrared and passive microwave radiometer data) has been used when available[6]. By studying the latest S1 image of a given area, experienced sea ice analysts carries out a detailed interpretation of the sea ice conditions at the time of acquisition. An ice chart included in a NetCDF file for a SAR scene consists of polygons in which all pixels containing the same polygon id are of the same ice conditions. The ice conditions are described according to the Egg Code (ref. section 2.2), however, in this thesis, only the total sea ice concentration will be utilized. This will be further explained in section 4.1.2.

With each ice chart in the dataset there is a table associating each polygon id to a given set of ice characteristics, in which the SIGRID-3 attribute is one of them. The SIGRID-3 values are representing the sea ice concentration and will be converted to percentage values for further use in this thesis. Table 3 shows the SIGRID-3 codes and their interpretations.

Table 3: SIGRID-3 codes and their sea ice concentration interpretations.

Definition, concentration	Sigrid3 code (CT, CA, CB and CC)
Ice Free	00
Less than 1/10 (open water)	01
Bergy water	02*
1/10	10
2/10	20
3/10	30
4/10	40
5/10	50
6/10	60
7/10	70
8/10	80
9/10	90
9+/10 (95%)	91**
10/10	92

Source: ASIP-v2 User Manual

[6] https://data.dtu.dk/articles/dataset/AI4Arctic_ASIP_Sea_Ice_Dataset_-_version_2/13011134?file=24951176

(This page is intentionally left blank)

Chapter 4

Sea Ice Segmentation using U-Net

In this chapter the methodology executed in this thesis will be described. As can be interpreted from the flowchart in Fig. 23, the process is divided into three parts; 1) preprocessing, 2) preparation for U-Net and 3) sea ice segmentation using U-Net. Each part consist of several processing steps. The preprocessing mainly evolves around loading, converting and downscaling the data. In the preparation for U-Net part, the images are divided into patches and missing values in the ASIP dataset are handled. This part also includes balancing and splitting the dataset into subsets used in training and evaluation of the deep learning models. Finally, the implementation and use of the deep neural network is described. This includes the application and modification of the existing architecture for U-Net, and the use of this modified architecture to train several models.

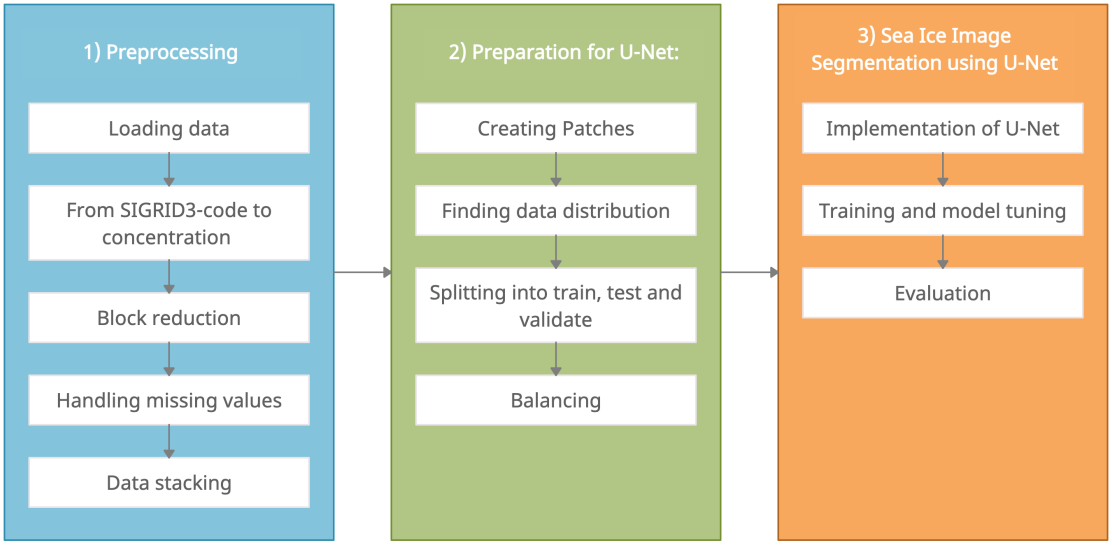


Figure 23: Overall project work flow

Two versions of the dataset were generated in this project, one multiclass- and one binary version. In the multiclass version, the ice concentration values in the ice charts were divided into 6 classes. For the binary version on the other hand, all pixels that had an ice concentration larger than 0% were assigned to the ice class, and all pixels containing 0% ice were labeled with Open Water.

Technical environment

This subsection will describe the technical solutions that were chosen for this thesis. Firstly, Python was chosen as programming language. Python is ranked as the number one programming language for Machine Learning applications. Additionally, this is the language that the authors has the most experience with. The Python implementation was written in Jupyter Notebook, a free and interactive, computational notebook which supports a combination of code and text. This combination makes the notebook suitable for code with some associated explanatory text. Moreover, Jupyter Notebook is well-documented, easy to use and easily configurable on a remote server. Regarding the next technical aspect, the entirety of the code and data has been hosted and downloaded on a remote server situated at NTNU. This remote server has a much larger storage space than a simple laptop and it provides a Graphical Processing Unit (GPU). Specifically, the GPU hardware is a NDIVIA Quadro GV100 32GB. These two aspects were crucial for this thesis. Primarily since it allowed the dataset to be downloaded, and secondly because the deep neural network could be trained within a limited time span.

There are a multitude of libraries and API's providing code to simplify the implementation of software in this category. The following list is a concatenation of the main libraries and other preimplemented code used in this thesis.

- **Numpy**[1] was used for mathematical operations performed on the matrices throughout the entire implementation in this thesis. Numpy provides support for mathematical operations on arrays and matrices in Python. It also provides functionality for storing and retrieving the mentioned data structures which was well used in this implementation.
- **NetCDF**[2] was used for accessing the files in the dataset. It is a collection of software libraries developed and supplied by Unidata Program Center. The collection provides the support for the creation, access and sharing of scientific data.
- **Keras**[3] was applied for the setup of the deep neural network in addition to processing related to the network. Keras is one of the main API's for deep learning in Python and is provided by TensorFlow. It provides out-of-the-box, simple and fast building blocks for deep learning. Moreover, it is well-documented and widely used, meaning that common problems or challenges are well discussed in open forums. Finally, it is compatible with GPU processing, in which is a great advantage.
- **Scikit-learn**[4] was used for some of the preprocessing, and for the evaluation of the deep learning models. Scikit-learn is another common library in machine learning. However, it does not support deep learning to the same extent that Keras do, and is not compatible with GPU's.
- **Matplotlib**[5] was used for visualization of the matrices and arrays throughout the thesis. It is a very popular library for plotting in Python, and is well-documented.

4.1 Preprocessing

This section will describe the preprocessing of the data. This includes loading the data, converting from SIGRID-3 codes to concentration and downsampling of the data. In conjunction with the AI4Arctic project, a github repository containing source code for loading and preparing the ASIP dataset was published[6]. The repository provides a simple function that extracts patches of valid data from the dataset. However, for the sake of being fully aware of what kind of preparation the

[1] https://numpy.org/doc/stable/reference/index.html
[2] https://www.unidata.ucar.edu/software/netcdf/
[3] https://keras.io/api/
[4] https://scikit-learn.org/stable/index.html
[5] https://matplotlib.org/
[6] https://github.com/damaha/asip-v2

dataset had gone through, the preprocessing in this project was written by the authors (with inspiration from the repository). Another factor affecting this decision was the ambition of becoming familiar with the necessary preprocessing steps in order to prepare the dataset for input to a deep neural network.

4.1.1 Loading Data

The dataset described in section 3, was provided as a zipped collection of NetCDF files. In order to process the data it had to be downloaded and extracted from the zip archive. Given the fact that the dataset was of a considerably large size, some issues were encountered in the downloading process. The issues mainly evolved around the discontinuity of internet connection throughout the downloading process, which required a continuous connection for more than ten hours. As referenced in section 3.1, this challenge resulted in loss of almost half the dataset. Aware of the importance of having enough data when training a deep neural network, this became one of the main challenges that the authors encountered in this thesis.

Once the part of the dataset utilized in this project was downloaded, it could be loaded into the Jupyter environment. This was straightforward mainly due to the usage of the NetCDF format, which is well-documented and easy to use. In Jupyter, each NetCDF file (hereafter revered to as nc-file) in the ASIP dataset was loaded into a Python dictionary. The content of the files were extracted using the NetCDF4-functionality called Dataset. According to Unidata and their documentation[7], *Dataset* is a class which provide a simple constructor to load and describe the meaning of the data, in addition to the relations among the data fields in the nc-file. Specifically for this application, each nc-file was loaded as an instance of the *Dataset* class, where each of the content items described in Table 2 was present as a so-called variable. An overview of the relevant variables in the dataset instance for every nc-file is given in Table 4. These variables were easily retrieved by using the name of the content. An example of the code for accessing the content of a variable in a *Dataset* instance is shown in 24.

Table 4: Relevant variables in the *Dataset* instance for every nc-file

Variable name	**Content**
nersc_sar_primary	HH polarized SAR data with ESA and NERSC noise correction
nersc_sar_secondary	HV polarized SAR data with ESA and NERSC noise correction
sar_incidenceangles	The incidence angle for the pixels in the image (**1D**)
polygon_icechart	DMI ice chart with polygon id's
polygon_codes	The conversion from polygon id's to SIGRID3 codes specific for each file

The variables in the table are two-dimensional arrays of the same size, except for the incidence angle which is one-dimensional. Therefore, to enable mathematical computation on all the arrays, where they comply in size and position, the incidence values were reshaped to a two-dimensional array.

$$\textbf{sar\_hh\_nersc} = dataset\_for\_file.\textbf{variable}[\text{"}nersc\_sar\_primary\text{"}][:] \tag{24}$$

In the code snippet in 24, $dataset\_for\_file$ is the dataset instance for a given file, and "$nersc\_sar\_primary$" is the variable name of the co-polarized HH S1 image with ESA *and* NERSC noise correction. The [:] at the end implies that the entire variable should be retrieved.

[7] https://unidata.github.io/netcdf4-python/Dataset

4.1.2 Converting from SIGRID-3 to Concentration

The polygon ids provided in the ice charts, described in section 3.3, had to be converted into concentration values in order to be used as labels. By employing a conversion table accompanying each SAR scene, the pixel values in the ice charts were converted from ids to SIGRID-3 codes. Thereafter the conversion table addressed in Table 3 was used to convert the SIGRID-3 code to the corresponding sea ice concentration.

The variable *polygon_codes* was loaded for each nc-file as described in section 4.1.1, and converted into a Pandas DataFrame[8]. Even though *polygon_codes* contain a wide specter of ice characteristic values, only the CT-values, referring to the total ice concentration, were taken into account.

For the conversion from SIGRID-3 codes to concentration and class labels three dictionaries were created. One dictionary was created for translating the SIGRID-3 encoded CT-values into percentages representing the actual ice concentration. This was necessary in order to visualize the sea ice concentration for comparison with the classified images in section 5. Two more dictionaries were created for relating each concentration to respectively; a multiclass label, and an binary label. The two latter dictionaries were used to create true labels for the deep network. The relation between concentration, class label and physical meaning is represented for the multiclass version of the dataset in Table 5, and for the for the binary version in Table 6 . The class labels in the multiclass version is based on the Norwegian Meteorological Institute's ice classification.

Table 5: Ice concentration intervals and their corresponding classes for the multiclass version

Concentration:	**Label**	**Description:**
<10%	1	Open Water
10% - 40%	2	Very Open Drift Ice
40% - 70%	3	Open Drift Ice
70% - 90%	4	Close Drift Ice
90% - 95%	5	Very Close Drift Ice
100%	6	Fast Ice

Table 6: Ice concentration intervals and their corresponding classes for the binary version

Concentration:	**Label**	**Description:**
0%	1	Open Water
0.05% - 100%	2	Sea Ice

4.1.3 Block Reduce

As addressed in section 2.3.3, the SAR images are affected by noise. Even though the Level-1 GRD products utilized in this thesis have been multi-looked, the images were downsampled in order to further mitigate the effect of this noise. This downsampling operation was performed by applying a function provided in the image processing toolbox from SciPy, *scikit-image*. SciPy is a widely used open-source library for scientific processing and computation in Python, providing packages including pandas and scikit-learn. The function is provided in the Measure module and is called Block

[8]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

Reduce. The block reduce function downsamples the input image by applying a certain function to neighbouring values. The user specifies the extent of the neighbouring pixels and what function to apply on them (common choices are mean or median of the pixels). For instance, an image of size 10000x10000 is downsampled to 5000x5000 if every 2x2 block of pixels in the image is averaged.

Block reduce was applied using the mean of the values inside a 2x2 window for the downsampling of the SAR images in both polarizations (HH and HV). This process reduced the noise in addition to the size of the images, and thus the required computational time. The code for executing the command is shown in 25.

$$data\_down\_sampled = \textbf{\textit{block\_reduce}}(data, \textbf{\textit{block\_size}} = (2, 2), \textbf{\textit{func}} = numpy.mean) \tag{25}$$

The average value was also applied to the Incidence angle values to preserve the size compliance between the matrices containing the SAR values and the incidence values. Similarly, downsampling was applied to the label data, however, to maintain the integer values of the data the median was applied instead of the mean function.

4.1.4 Handling Missing Values

Both the ice charts and the SAR images contain missing values due to areas not being of interest or values ranging outside some defined interval. These values could not simply be removed due to the size and position compliance in the matrices. Therefore, it was necessary to find a masking array which contained all the missing values for all the variables. A masking array is of the same dimensions as the original array and contain Boolean values (True or False) where True indicates a missing value and False represent valid values.

The process of creating a mask for all the missing values present in all the array's of the variables shown in Table 4 was performed in two steps. Firstly, the missing values in the ice charts were determined. This was done by exploiting the fact that *block_reduce* with the median operator function returns a value of 0.0 if every value in the block is a missing value. This way the masking array could easily be extracted by retrieving the elements at the positions with 0.0. This was only possible due to the fact that none of the other classes had a label value of 0.

Moreover, the missing values for the SAR and incidence angle data were extracted by using numpy's function, *isnan*(). This function takes a multi-dimensional array as input and returns a new Boolean array where True indicates the position of a NaN value (missing value) in the original array, and False represents the valid elements. To apply this function on the data simultaneously, the two SAR polarizations and the incidence angle were stacked into a multi-dimensional array of the shape (H, W, 3), where H and W is the width and height of the image, and 3 is the depth (SAR-HH, SAR-HV and incidence values). The *isnan*() function was then applied on the stacked array, resulting in a boolean array with the same dimension as the input. This array was then converted to an array of 1 (corresponding to True) and 0 (corresponding to False) values. Finally, this multi-dimensional array of 1 and 0 values, was summed over the last dimension. I.e. the 1 and 0 values representing all the elements in the SAR-HH, SAR-HV and incidence angle were summed. This resulted in a summed array with the same dimensions as the input image of (H, W), where the maximum value at every position was 3, and the minimum was 0. If the number was above 0, it meant that one of the SAR-HH, SAR-HV and incidence angle had a missing value at this position. Thereafter, a masking array was produced from the elements of the summed array with True for the elemets with a value above 0.

Finally, the two described masking arrays were combined using NumPy's function, *logical_or()*, as shown in (26). This function uses the logical expression *OR*, to combine two boolean arrays. The resulting array had *True* in all pixels where any of the input variables had a missing value in

their corresponding pixel-position, and *False* in all other pixels.

$$combined\_masking\_array = \textbf{\textit{np.logical\_or}}(masking\_array1, masking\_array2) \tag{26}$$

Since almost all the images contained missing values, an exclusion of these would be unacceptable. However, in training and evaluation of deep learning models, missing values are not supported. Therefore, after creating the combined masking array, the missing values in the images had to be replaced with a real value. Moreover, the values from the different variables in the nc-files are very different. For example, the SAR-HH values range from -32 to -7, while the incidence values range from 19 to 47. Thus, a specific replacement value for each of the variables had to be defined, and the chosen values are displayed in table 7.

Table 7: Replacement constants for the missing values in the specific variables

Variable	**Replacement Constant**
SAR-HH	-20
SAR-HV	-20
Incidence value	0.1
Ice chart	0

The replacement value of -20 for SAR-HH and SAR-HV was chosen due to the fact that it is close to the middle of the range for those values, and there are no pixels in the dataset where the SAR-HH and SAR-HV have this exact value. Additionally, if an extreme value of for instance 1000 was chosen this would have created a larger range of values for the model to adapt to in the SAR values, which is not preferable. A replacement value of 0.1 was selected for the NaN-values in the incidence angles. This was due to the fact that 0.1 is a valid incidence angle value, however it is not present in the data set for any of the images. It could have been set to 0, but this could have yielded some mathematical difficulties, therefore it was set to 0.1. The missing values in the ice chart were set to 0, because 0 was not used for any of the other sea ice concentration classes.

4.1.5 Stacking the Data

Prior to further processing it was necessary to reassure that all data belonging to one specific SAR scene was conserved together. Hence, while processing each SAR scene, its corresponding data was stacked depth wise. For this purpose, Numpy's function for stacking arrays in sequence along the third axis, *numpy.dstack()*, was utilized. This resulted in a 4 dimensional array of the shape (X, H, W, 7), where; X represents the number of SAR scenes, H and W is the shape of the SAR images, and 7 represents the data associated with the scene. In this manner, one was able to retrieve the data associated with a scene by accessing this element in the array. For instance, the data of the first SAR scene could be accessed by extracting the first element in the array.

The seven data elements for each SAR scene is listed below:

- SAR Primary (HH polarized)
- SAR Secondary (HV polarized)
- Incidence angle
- Ice chart containing polygon ids
- Concentration chart
- Class chart
- The masking array

The stacking of the elements guaranteed that each SAR scene was fixed together with its corresponding data and the process of splitting into train, test and validate could be performed seamlessly.

4.2 Preparation for U-Net

After having applied the necessary preprocessing of the dataset, the data had to be transformed to a format which the deep neural network was able to ingest. Firstly, the size of the input data had to be reduced to a consumable size for the CNN. This was done by creating patches of the original input data. After transforming the dataset into patch format, it was split into random subsets of training, validation and testing patches. Thereafter, the class distribution was found and the subsets were balanced. The balancing was only applied for the multiclass version of the dataset, due to the fact that the distribution of the classes in the binary version of the dataset is not considered to be imbalanced.

4.2.1 Create Patches

In order to utilize the stacked data as input to a CNN, it was necessary to split the images into sub-scenes. These sub-scenes, hereafter referred to as patches, are crucial in order to enable processing of the large SAR images since the input size is limited by the memory capacity. Furthermore, by allowing the CNN to examine one fraction of the image at a time, the applied filters can detect smaller features in the SAR images. In addition to the practical reasons referenced above, the patches also introduce a regularization property. By examining a smaller number of pixels each time, the network will aspire good results for each patch without knowing what is contained in the adjacent patches.

The size of the input to the deep neural networks has to be divisible by two multiple times, therefore 256x256 and 512x512 are common patch sizes. The stacked SAR scenes were split into patches of 256x256 pixels, resulting in an array of the shape of (X, 256, 256, 7) where X is the number of patches. A sliding window traversing each SAR scene was applied to extract the patches that was going to be fed as input to the CNN. For this task, the authors utilized the *sliding_window()* function provided in the Github repository (referenced in the introduction of this section). The function takes in the stacked data, window size and striding size. The window size is set to (256, 256), and by default the striding size is set to be equal to the window size, entailing no overlapping patches. Numpy's function *as_strided()*[9] is called inside *sliding_window()* and creates a view into the array.

Commonly, one would set the strides to half the size of the sliding window in order to create an overlap. However, in this project no overlapping was implemented due to a limited amount of storage space available. The impact of this decision will be further discussed in section 5.3.

The dataset contain a large amount of missing values, hence some of the patches contain solely missing values. These patches do not provide any valuable information to the sea ice segmentation task. Therefore, these patches were removed from the dataset.

4.2.2 Finding the Class Distribution of the Patches

In order to find the class distribution in the two versions of the dataset, a function was implemented to determine the class of each patch. The same class partitioning as described in Table 5 for the multi-class version and Table 6 for the binary as in the ice charts was utilized. Before defining the class of the patches it was important to have some knowledge of the distribution of the missing values, water and ice pixels in the dataset. Fortunately, the ASIP dataset comes with a table

[9] https://numpy.org/doc/stable/reference/generated/numpy.lib.stride$_t ricks.as_s trided.html$

describing the percentage of ice, water and missing values in each of the nc-files. By inspecting this table it was evident that there is a large amount of missing values and that for many of the images there are a majority percentage of water compared to ice. Therefore, to determine the class of each of the patches, the classes of all the pixels not labeled as a missing value or water in the patch were averaged.

To cope with challenges related to the overweight of Open Water and missing values, two modifications were made to the standard averaging. Before computing the average of all the pixels with a ice label in the patch, a check was made. This check tested whether the number of pixels labeled as missing values or labeled as Open Water was greater than 80% of all pixels in the patch. If this was the case, the patch would be labeled with the missing values class or the open water class respectively. Additionally, if the patch contained only missing values and Open Water, but none of them were represented with 80% more than the other class, the patch would be labeled as open water. These modifications were made to avoid situations where for instance patches containing 50% water and 50% fast ice, were labeled as the class Open drift ice(ice concentration 40-70%), when it in fact contain valuable information about the fast ice class(ice concentration 100%). However, if a patch contained more than 80% Open Water pixels, the majority would be sufficiently large to label it as an Open Water patch.

To keep an overview of the patches belonging to each class, a dictionary with the patch name and class was created for both the multiclass and binary versions. This made it easy to obtain the distribution of the patches belonging to each class in the dataset which is shown in Fig. 24 for the multiclass version, and in Fig. 25 for the binary version. From Fig. 24 it is easily interpreted that the number of patches labeled with Open Water is very high compared to the rest of the classes for the multiclass dataset.

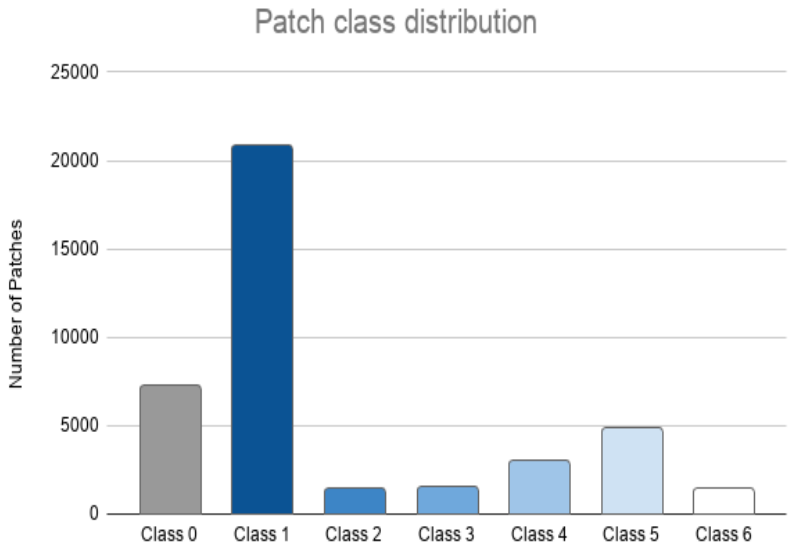


Figure 24: The distribution of the number of patches belonging to each class in the multiclass version, where the classes are; 0 : Missing value, 1: Open Water, 2: Very Open Drift Ice, 3: Open Drift Ice, 4: Close Drift Ice, 5: Very Close Drift Ice, 6: Fast Ice.

The distribution in the binary version of the dataset is given in Fig. 25. From this figure one can observe that the number of patches in each class is not as unevenly distributed as in the multiclass version, and the number in each class is considered sufficient for the scope of this thesis.

Figure 25: The distribution of the number of patches belonging to each class in the binary version, where the classes are; 0 : Missing value, 1: Open Water, 2: Ice.

4.2.3 Splitting into Train, Test and Validate Subsets

After the desired patches were produced, they had to be split into subsets for training, validation and testing of the neural networks. This was done by using the function *train_test_split()* from the module *model_selection* provided by Scikit-learn. This function splits an array into random train and test subsets, where the user can specify the size ratio between the two subsets. The function was used on an array of the names of all the patches, to find which patches to include in which subset. This was done once to split into train and test, with the fraction 0.25 for the test set. The test subset was subsequently split into validation and test subset, with a fraction of 0.4. The split resulted in the distribution of patches displayed in Table 8.

Table 8: Patch distribution in the train, test and validate subsets

Subset	**Patches**	**Fraction**
Train	30292	0.75
Validate	4040	0.10
Test	6058	0.15
Total	**40390**	**1.0**

4.2.4 Balancing

After obtaining the names of all the patches in each of the subsets (train, validate and test), the patches were moved to the corresponding subset folder. Furthermore, the number of patches belonging to each class was calculated for each of the subsets. As mentioned, there was created two versions of the dataset, one multiclass and one binary. Most work on imbalanced data is based on cases where the ratio of imbalance between the majority and the minority class ranges from 1:4 to 1:100 [27]. The specific number of patches in each class in the multiclass version is given in Table 9. In the table, it is evident that the number of patches belonging to each class in the train, validate and test subset follows the same distribution as the entire dataset. Where, specifically, the Open water class is overrepresented, and the classes 2, 3 and 6 are underrepresented. In the multiclass case the imbalance ratio between the minority and majority class is close to 1:10.Thereby it can be characterized as imbalanced. In the binary dataset however, the ratio is close to 1:2 meaning

that it is not considered imbalanced.

In machine learning unevenly distributed datasets can cause models to perform better in the overrepresented classes than in underrepresented ones. Given the fact that the objective in this thesis was to perform well in classification of several classes, where some of them were relatively underrepresented, balancing was applied.

Table 9: The number of the patches belonging to each ice class in each of the subsets for the multiclass case

Class	**Train**	**Validate**	**Test**
0 : Missing values	5443	726	1117
1 : Open water	15704	2062	3094
2 : Very open drift ice	1077	134	232
3 : Open drift ice	1084	149	242
4 : Close drift ice	2249	274	459
5 : Very close drift ice	3611	539	723
6 : Fast ice	1124	156	191
Total	**30292**	**4040**	**6058**

Table 10: The number of the patches belonging to each class in each of the subsets for the binary case

Class	**Train**	**Validate**	**Test**
0 : Missing values	22243	2978	4449
1 : Open water	1447	184	323
2 : Ice	7428	988	1451
Total	**31118**	**4150**	**6223**

Due to the fact that there are both overrepresented and underrepresented classes, a strategy to include both oversampling and undersampling was utilized in the balancing. First of all, the desired number of patches in each class was defined. This was defined to be six times the number of patches in the minority class. It is possible to rotate, flip and transpose a matrix six times without creating duplicates. The rotation provides three new patches, by rotating 90, 180 and 270 degrees. Flipping allows for two new patches to be generated by flipping up-down and left-right. Lastly, transposing the matrices yielded one new patch. This resulted in six new patches.

Moreover, even though balancing is commonly used to create a similar amount of instances in each of the classes, it was desirable that the Open Water class was slightly more represented. This was decided due to the fact that Open Water is overrepresented in the real world, and in most applications it is preferable that the models perform well in classification of water. Therefore, the desired number of patches for the Open Water class was defined as 1.5 times the desired number of patches for the other classes.

After defining the desired number of patches for each class, this number was compared to the balance of each class. For the underrepresented classes, image augmentation was applied. The augmentation was performed in an iterative manner, where a random sample of the patches existing in the given class was used. For every iteration, one of the augmentation techniques previously introduced was applied on the patch, ensuring only unique patches. Undersampling was applied the overrepresented classes. This was done by randomly removing the number of patches above the desired number. The distribution of the balanced multiclass dataset can be viewed in Fig. 26, where all classes have the same number of patches except for class 1 : Open Water. As is evident in Fig. 26, balancing was only performed on the training dataset. The test and validation data was not balanced, due to the fact that the goal of the model is to perform well on data from the real world.

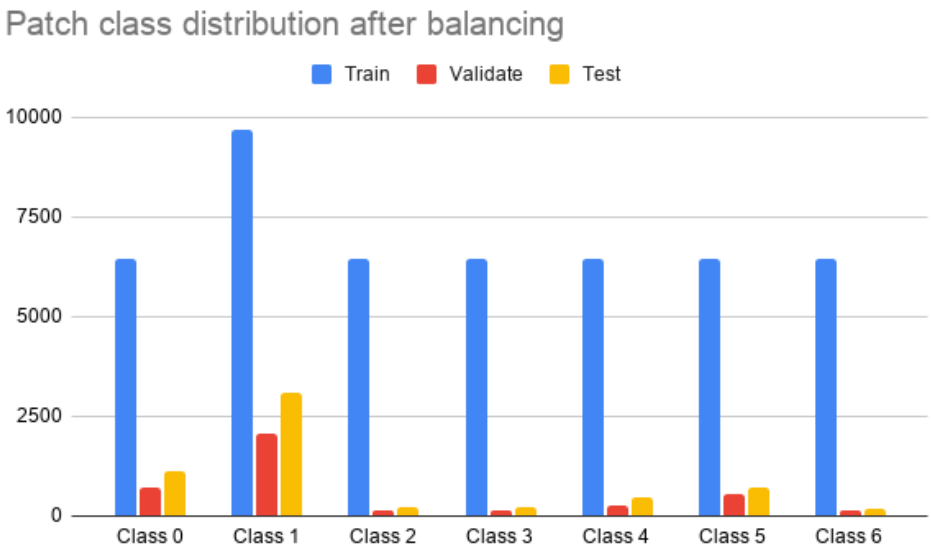


Figure 26: The distribution of patches for train in blue, validate in red and test in yellow after the balancing process.

The deep neural network required the data to be split into input values $\mathbf{x}$ (the SAR polarization values and the incidence angle), and ground truth $\mathbf{y}$ (the ice class for each of the pixels in the patches). $\mathbf{x}$ and $\mathbf{y}$ are arrays with the following dimension (i, ph, pw, k), where i is the number of patches present in this subset(i.e. train, validate or test), ph is the patch height and pw is the patch width, and finally k is the number of values. For $\mathbf{x}$, k is 3 due to there being three input parameters, including SAR HH, SAR HV and incidence angle. For $\mathbf{y}$, k is 1 because it only contains the ground truth classes.

4.3 Prediction using U-Net

As mentioned in section 2.4.7, U-Net is a fully convolutional deep neural network developed for biomedical image segmentation. It is a common network for semantic segmentation of images due to its high performance and low computational time. The modified version of U-Net applied in this thesis is based on the structure defined in the original paper [16] and the github-repository provided by Lamba[10].

In the original architecture, shown in Fig. 15, the size of the input image was (512x512x3). However, in the modified version, displayed in Fig. 27, the size of the input image is (256x256x3). Additionally, the number of filters used in each of the convolutional layers has been lowered to limit the computational load. As can be viewed in the figure, the number of filters commence at 16 in the first layer and is doubled for each of the blocks of the contracting path, and halved for each of the blocks in the expanding path. In the figure it is also evident that the modified version does not lower the height and width of the input in each of the blocks. This is due to the use of the padding option "same" in the convolutional layers. This option allows the use of padding to ensure that input is of the same size as the output.

Moreover, the original U-Net was developed for a binary segmentation task, thus, the depth of the output layer was 2. For the multiclass case in this thesis, the depth of the output layer was 7 as there was 7 classes. While, for the so-called binary case in this thesis, the output layer had 3 channels; two for each of the classes (water, ice) and one for the missing values. Additionally, the activation function used in the output of the original implementation was Sigmoid. However,

[10]https://github.com/hlamba28/UNET-TGS/blob/master/TGS%20UNET.ipynb

this is not well-suited for cases with more than two classes. Therefore, the modified version of the U-Net uses a Softmax activation function for the output layer. Lastly, the transposed convolution operation in the expansive path of the network in the original architecture uses a filter size of (2x2). In the modified version, on the other hand, a filter size of (3x3) is used. This was done to ensure that the output of the model was of the same dimensions as the input.

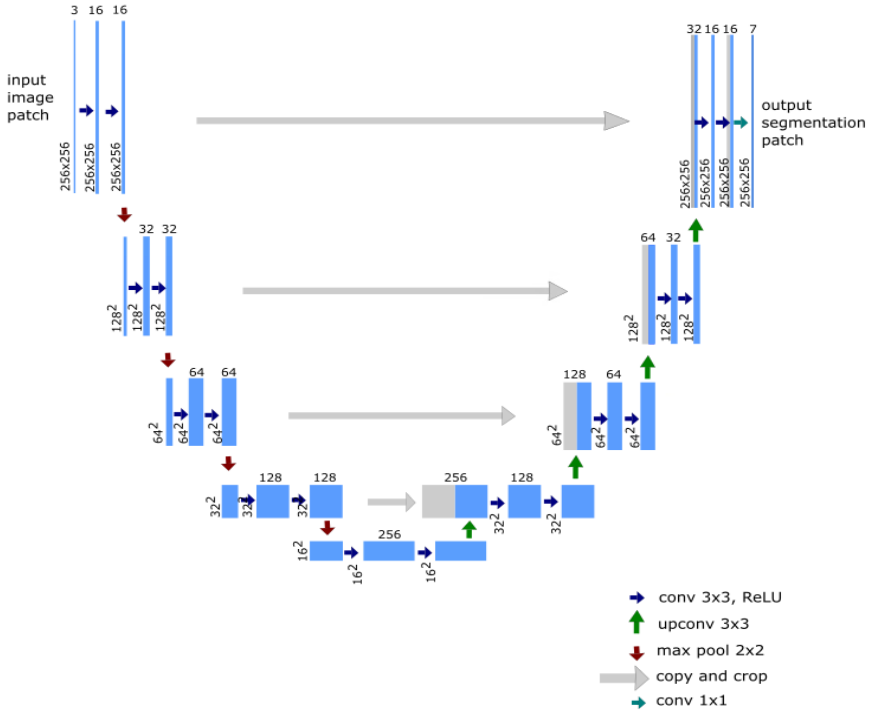


Figure 27: The modified U-Net architecture used in this thesis

4.3.1 Training of Deep Neural Networks

A common convention in deep learning is to tune the models based on hyperparameters such as learning rate, batch size and regularization techniques. However, the main focus in this thesis is the data and the related processing, and not the deep learning part individually. Therefore, the differences between the models are mainly related to different versions of the data, and some larger differences in the hyperparameters. Every model was trained with the *Cross Entropy Loss* function, to be able to compare the results obtained from training the models with different datasets. Furthermore, all the models were trained for 100 epochs, however, early stopping was enabled. This setting allows the models to stop early if the loss value does not improve for ten epochs. The batch size, i.e. the number of patches processed in each iteration before updating the weights, was also constant at 16 for all the models. This value was selected due to the fact that it was small enough to fit in the GPU's memory without occupying the entire memory space. Batch normalization was also performed in each convolutional layer, meaning that the input values to every convolutional layer were normalized. In this thesis, several different models have been trained. The four models that will be presented and their characteristics can be viewed in Table 11.

The Baseline model was trained to obtain a basis for comparison. This model utilized the SGD optimizer, ReLU activation in each block, and a constant learning rate of 0.01. A dropout of 0.05 was used. Dropout is a commonly applied technique to prevent overfitting in machine learning. It randomly chooses the specified percentage of neurons, and does not consider them. The Baseline model was trained with the balanced version of the dataset. As can be observed in Table 12, it achieved a final training accuracy, training loss, validation accuracy and validation loss of respectively 0.7248, 0.6876, 0.7137 and 0.6788. This model was trained for 33 epochs before stopping early.

Table 11: The models included in this thesis, and their characteristics

Name	**Optimizer**	**Activation function**	**Learning rate**	**Dropout?**	**Dataset**
Baseline	SGD	ReLU	0.01	Yes	Balanced
Model 1	Adam	LeakyReLU	0.001	No	Balanced
Model 2	Adam	ReLU	0.001	No	Imbalanced
Binary	Adam	ReLU	0.001	No	Binary

Model 1 is an improved version of the Baseline model, and was also trained with the balanced dataset. This model uses the Adam optimizer which enables an adaptive learning rate initialized with 0.001. LeakyReLU yielded better training results than the normal ReLU, and was therefore applied as activation function in each of the blocks. Since the Baseline model did not seem to overfit, the drop out technique was not utilized in this model. The final training metrics for Model 1, as given in Table 12, is 0.8221, 0.4451, 0.6375 and 1.1736 for the training accuracy, training loss, validation accuracy and validation loss, respectively. When comparing this model to the Baseline, it is evident that it achieves higher training accuracy and lower training loss. However, the validation accuracy is lower and the validation loss is higher than in the Baseline. This will be discussed in section 5.3. Also, the training converges earlier, in which allows the training to stop at epoch 29.

Model 2 is similar to Model 1 except that it was trained on an imbalanced dataset. Moreover, the LeakyReLU activation yielded worse results than from using ReLU, therefore normal ReLU was used in this model. The model achieves better accuracy- and loss values than Model 1 and the Baseline model. This was expected since the training- and validation dataset for this model follows the same class distribution. The model converged slower than the Baseline and Model 1 with an early stopping at 47 epochs.

Finally, the Binary model was trained on the binary version of the dataset. Similarly to the imbalanced model it achieved better results with the normal ReLU activation function as apposed to the LeakyReLU. This model achieved the overall best training performance as can be viewed in Table 12. Already in the first epoch it achieved very high accuracy. This was probably due to the fact that the classification had been significantly simplified. Still there was an improvement in the accuracy of 0.8975 to 0.9847 over the 56 epochs of training.

Table 12: The final values of the metrics used in the training and validation of the models.

Model	**Training Accuracy**	**Training Loss**	**Validation Accuracy**	**Validation Loss**
Baseline	0.7248	0.6876	0.7137	0.6788
Model 1	0.8221	0.4451	0.6375	1.1736
Model 2	0.9051	0.2382	0.8975	0.2571
Binary	0.9847	0.0409	0.9766	0.0617

4.4 Evaluation

An important aspect in machine learning is the evaluation of the model performance. This is done by comparing the ground truth and the predicted output. The evaluation metrics best suited for a given model depend on several factors, including, but not limited to, the number of classes, the distribution of examples belonging to each class and the goal of the model. In this thesis there are two different classification cases, one binary and one multiclass. These cases differ to such a degree that they require different metrics to measure their performance.

4.4.1 Choosing Performance Metrics for Binary Classification

In the case of binary sea ice segmentation it is crucial that the trained model is able to detect as many true positives (indicating sea ice) as possible. Since navigators are naturally planning their polar routes through areas predicted to be open water (i.e. the negative class), a large proportion of false negatives in the predicted data could potentially be unfortunate. As earlier referenced, recall is a good measure of how the proportion of true positives to all positive. Hence, recall should be examined when measuring the performance of the binary sea ice classifier. This metric is included in the f1-score. The binary dataset applied in this project is separated into sea ice and open water with a distribution close to 3/4 (open water) and 1/4 (sea ice). This does not qualify as an imbalanced distribution, hence, both accuracy and f1-score can be used for measuring of the performance for the models trained on the binary dataset. Moreover, a confusion matrix and the ROC AUC score was examined. As referenced in section 2.5.1 and 2.5.3, these are good indicators for assessing the threshold between the classes in the dataset. Finally, the reliable MCC score was be applied which, as referenced in section 2.5.5, is a good overall score for the binary classifier.

4.4.2 Choosing Performance Metrics for Multi-class Classification

If the multiclass classifiers, were to rely solely on the same performance measures as for the binary classifier, there would arise misinterpretations. Since f1-score, accuracy and ROC AUC only rewards the correctly predicted classes, no reward is given if the prediction is wrong but close to the true class. I.e. if a pixel is predicted to be of class 4 (meaning there is between 70% and 90% ice in that pixel), when it actually is an instance of class 5 (90% to 95% ice), the model would be penalized equally as if it predicted it to be of class 0 (open water). This way, the multiclass classification problem is more similar to a regression problem than a classification problem. And even though the binary evaluation metrics still give a small indication of how well the model performs, additional evaluation metrics have to be taken into consideration. Since the multiclass classification problem to some extent can resemble a regression problem, the R^2-score was applied as a performance measure. This score assesses the linear relationship between the true and predicted class and gives an indication of how well the classifier fits the true pattern. Finally, the confusion matrix is a good tool for evaluating the performance of each class individually, and was therefore applied in the evaluation of both the binary classifier and the multiclass classifier.

Chapter 5

Experimental Results and Discussion

In this section a presentation and discussion of the results obtained in this thesis will be given. Firstly, a quantitative analysis of the results will be presented. These results were obtained by applying the evaluation methods described in 4.4 with the independent test dataset. The test dataset contain data in which the models have never seen before, thus providing a basis for evaluating the performance of the models on new data. Subsequently, there will be given a qualitative analysis of the model performances on a visualization dataset. This dataset has also never been seen by the models, and was specifically chosen based on the diversity of the ice classes.

5.1 Quantitative Analysis of Results

In Table 13, the models and their corresponding performance metrics are presented. These metrics were achieved by evaluating each model with the test dataset. Accuracy and f1-score were applied in the evaluation of all the models. As addressed in section 2.5.4, the R^2-score measures the distance from the predicted label to the true value, and is therefore a common performance measure in regression problems. The multiclass classification problem in which contain classes of sea ice concentrations distributed from open water to fast ice, can resemble a regression problem. Therefore, the Baseline, Model 1 and Model 2 are evaluated with the R^2-score. Regarding the binary classifier, this metric will not provide any additional information since the distance between the two possible classes would always be the same. Hence it is not applied in the evaluation of the binary classifier. Instead the MCC and $ROCAUC$-score, in which are well suited for binary classification, were applied to the Binary model.

The rest of this subsection will present the models' quantitative performance and their corresponding confusion matrices. Based on the performance metrics presented in Table 13, it is evident that the Binary model outperforms the other models in all the metrics. However, this model exclusively concerns two classes in which makes the task considerably less complex. Hence, throughout this analysis, the classifiers for the two different problems will be evaluated separately.

Multiclass Image Segmentation

Regarding the the multiclass classifiers, the Baseline model obtained a relatively low R^2-score of 0.1281, and an f1-score of 0.3992. However, it achieved an accuracy of 0.7126 which is higher than for Model 1. Model 1 achieved an accuracy of 0.6290, an f1-score of 0.4843 and a R^2-score of 0.5407. The overall best performance was achieved by Model 2 with an accuracy of 0.9061, an f1-Score of 0.6222 and an R^2-score of 0.8709.

Table 13: The four models and their corresponding performance metrics

Model	Accuracy	F1-Score	R^2	ROC AUC	MCC
Baseline (multiclass)	**0.7126**	**0.3992**	**0.1281**	X	X
Model 1 (multiclass)	**0.6290**	**0.4843**	**0.5407**	X	X
Model 2 (multiclass)	**0.9061**	**0.6222**	**0.8709**	X	X
Binary	**0.9800**	**0.9706**	X	**0.9680**	**0.9412**

The confusion matrices of the Baseline model, Model 1 and Model 2 are presented in, respectively, Fig. 28, Fig. 29 and Fig. 30. Positions in the confusion matrix will be referenced as (Predicted, True), with the first axis given as the predicted label, and the second axis given as the true label. The class representing the missing values (originally labeled with 0) is removed from the confusion matrices due to the fact that it does not contain any valuable information. Hence, all the classes have been shifted one class down, meaning that class 0 represents Open Water and class 5 represents Fast Ice. The diagonal elements in the matrix represents the recall of each class. E.g. the element in position (0,0) represents the proportion of the true Open Water pixels that are successfully classified as Open Water.

Fig. 28 presents the confusion matrix for the **Baseline model**. With a recall of 0.97 (in position (0,0)), it is evident that the model performs well in finding the Open Water class. However, the performance is significantly reduced in predicting the other classes. Specifically, class 1 and 2, with the diagonal values of respectively 0.04 to 0.17, are not achieving satisfying results. The second best recall value(0.56) is achieved in the prediction of class 4. Moreover, it can be observed that the Baseline makes several wrong predictions in terms of False Positive Open Water pixels in which is the most critical class. If polar navigators were to apply the ice charts created by this model, they could potentially encounter several areas with sea ice in which were labeled with Open Water.

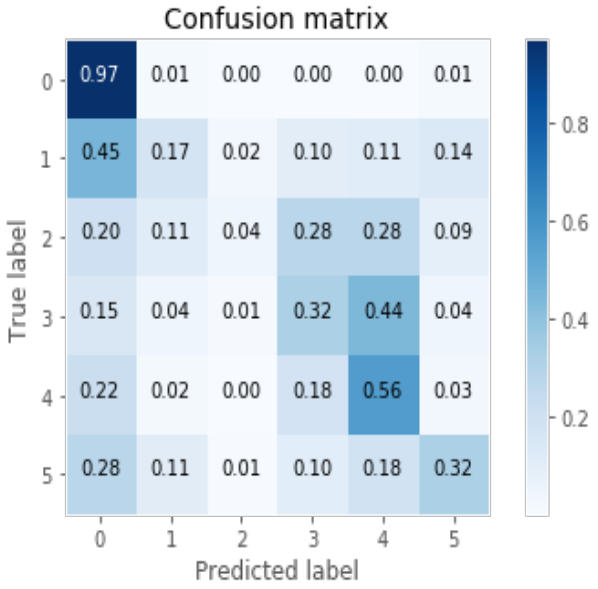


Figure 28: The confusion matrix for the Baseline model
The labels are given as follows ; 0 : Open Water, 1: Very Open Drift Ice, 2: Open Drift Ice, 3: Close Drift Ice, 4: Very Close Drift Ice, 5: Fast Ice

There are several reasons for the bad performance of this baseline. Primarily, when applying the SGD optimizer with a high learning rate, it can overshoot the global optimum such that the optimizer fails [28]. The Baseline model is initialized with a relatively large learning rate of 0.01 and the SGD optimizer does not adjust the learning rate during the training. Hence, it is evident that this set of parameters is not optimal. Moreover, dropout was applied as a regularization technique in the Baseline model. This is common practice in image segmentation problems in order to avoid overfitting. However, the dropout might disturb the network from optimizing its weights if the model is not prone to overfitting.

The confusion matrix representing the performance of **Model 1** is displayed in Fig. 29. A more prominent diagonal pattern can be observed in this figure. This indicates that Model 1 finds a more appropriate fit to the ground truth than the Baseline model. When comparing the matrices in Fig. 28 and Fig. 29 it is clear that the Baseline is better at predicting the Open Water class (class 0 in the confusion matrix). However, the R^2-score achieved by Model 1 (0.5407) is more than four times higher than the R^2-score of the Baseline model (0.1281). These two discoveries indicate that Model 1 makes fewer correct predictions than the Baseline model, but, at the same time, it achieves a higher degree of linear correlation between the true- and predicted values. Additionally, even though the number of False Positive predictions for the Open Water class is still quite high, it is lower than for the Baseline model.

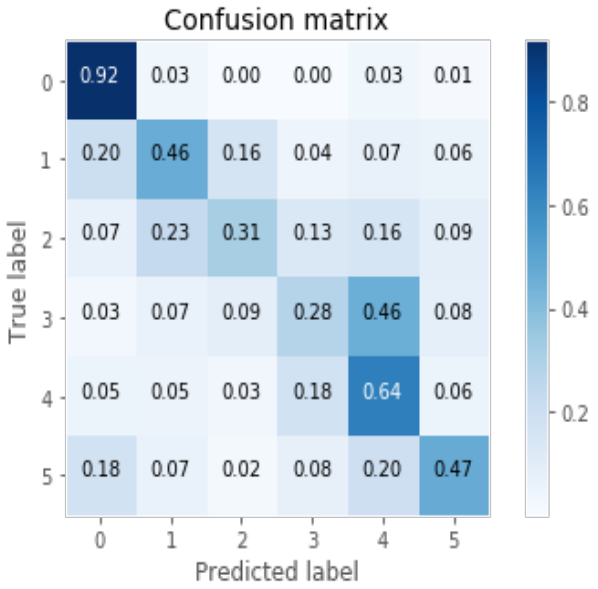


Figure 29: The confusion matrix for Model 1.
The labels are given as follows ; 0 : Open Water, 1: Very Open Drift Ice, 2: Open Drift Ice, 3: Close Drift Ice, 4: Very Close Drift Ice, 5: Fast Ice

The most significant difference between the implementation of the Baseline model and Model 1 is the choice of optimizer. The Baseline model utilizes the simple SGD optimizer, while Model 1 uses the Adam optimizer (addressed in section 2.4.5). The difference in the performance in the two models reflects the importance of choosing the correct optimizer for the given application.

The confusion matrix for the last multiclass model, **Model 2**, given in Fig. 30, is superior to the confusion matrix for Model 1. All the diagonal values, except for class 2, are higher for Model 2 than for Model 1. Since the diagonal corresponds to the recall for each class, it is clear that Model 2 makes more correct predictions than Model 1. The relatively high R^2-score obtained by Model 2 is also reflected in the focused diagonal in the confusion matrix, and the neighbouring cells.

The results obtained by Model 2 can be compared with other studies. Malmgren-Hansen et al.

[29] trained and evaluated a CNN with the ASIP dataset, in which a data fusion of both the SAR-data and the AMSR2-data was performed. In their study they were able to obtain an R^2-score of 0.89, when predicting sea ice concentrations. Model 2 achieved a slightly lower R^2-score of 0.87. However, the use of AMSR2 data was not utilized in this thesis, and only 240 SAR-scenes were applied, in which can explain the difference. Therefore, it can be argued that Model 2 obtained comparable results to Malmgren-Hansen.

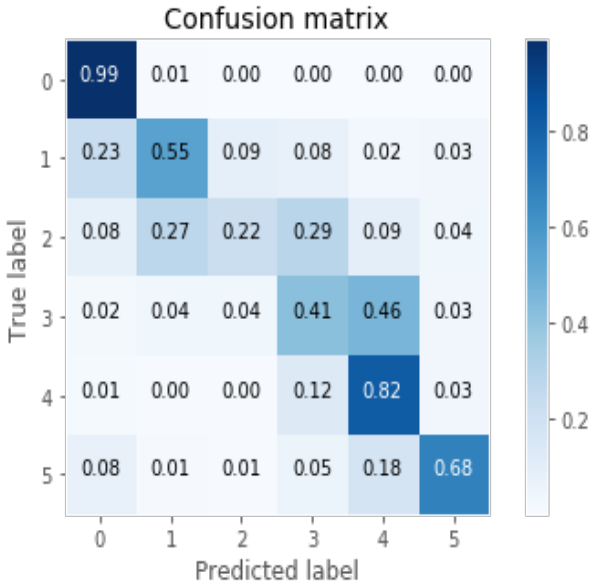


Figure 30: The confusion matrix for Model 2.
The labels are given as follows ; 0 : Open Water, 1: Very Open Drift Ice, 2: Open Drift Ice, 3: Close Drift Ice, 4: Very Close Drift Ice, 5: Fast Ice

In contrast to what the authors of this thesis hypothesized, the Baseline model and Model 1 achieved a lower performance than Model 2. Both these models were trained with the balanced data set. Thereby, they have been exposed to more instances of the underrepresented intermediate classes (1, 2, 3 and 4) than Model 2, which was trained on the unbalanced dataset. When training a model on a given dataset, it becomes familiar with the class distribution of that dataset. If the same model is given a dataset of another distribution subsequently, it can get "confused". This might have been the case with the models in section 5.2.2 and 5.2.1. This will be further discussed in section 5.3.

Regarding all the confusion matrices presented above, the two classes with the most correct predictions are the Open Water class (class 0) and the class for Very Open Drift Ice (class 4). It is interesting that each model performs quite differently on class 2 (Open Drift Ice). Specifically, the Baseline model performs very poor when predicting this class. By inspecting the class distribution in the imbalanced dataset given in Fig. 24, it is clear that there are more patches labelled with Open Water and Very Close Drift Ice than the other classes. This explains the high performance of these two classes in Model 2. However, the models trained on the balanced dataset have been exposed to an equal number of all the classes (except for Open Water), and it would be natural to expect that the performance was close to similar for all the ice classes. This is not the case for the Baseline model and Model 1.

The varying class performance of the two models trained on the balanced dataset (addressed in the previous paragraph) is due to the balancing approach applied in this thesis (Section 4.2.4). Since the original amount of instances of each class deviates from each other to a large extent, the number of necessary augmentation iterations varies as well. I.e. the class representing Very Close Drift Ice, which is more than 3 times larger than the class for Very Open Drift Ice, will be augmented a significantly lower number of iterations than the other. The augmentation method applied in this thesis (described in section 4.2.4) utilizes transposition, rotation and flipping. Hence, no

changes are made to the internal structure of the patches. The amount of new information introduced by augmenting a patch is significantly lower than the information value in an original patch. Therefore, there is a larger amount of information in the classes which are more represented in the original dataset.

Binary image segmentation

The confusion matrix, representing the performance of the Binary classifier, is presented in Fig. 31. The recall for Open Water is 0.99 which is the highest performance for that specific class achieved in this thesis. Overall accuracy, precision and recall are commonly applied metrics for evaluating binary classifiers. And for the sake of being able to compare the results obtained by the Binary model with other implementations, these metrics are given in Table 14.

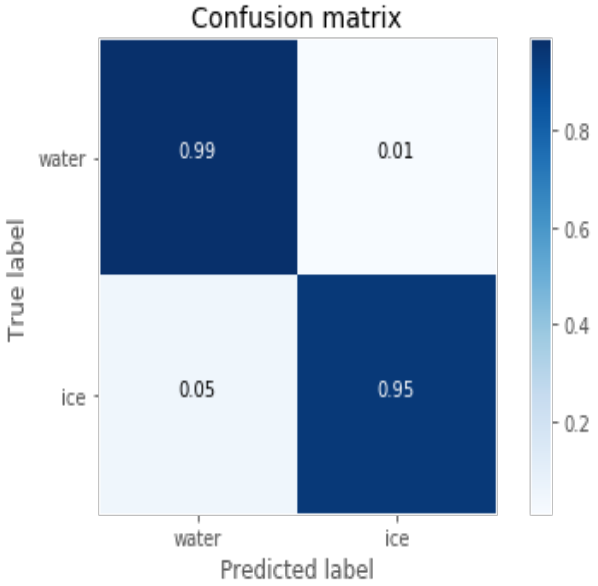


Figure 31: The confusion matrix for the Binary model, the labels are given as follows ; 0 : Open Water, 1: Very Open Drift Ice, 2: Open Drift Ice, 3: Close Drift Ice, 4: Very Close Drift Ice, 5: Fast Ice

Table 14: Binary performance metrics

Metric	**Value**
Accuracy	0.9800
Precision	0.9819
Recall	0.9782

The Binary model obtained an accuracy of 0.9800, a precision of 0.9819 and a recall of 0.9782. In their implementation of the U-Net, Ren et al. [30] achieved an accuracy, a precision and a recall of 0.93, 0.94 and 0.96, respectively. They also implemented an improved Dual attention U-Net [31], which achieved accuracy, precision and recall values of 0.97, 0.96 and 0.98, respectively. Thereby, the Binary model implemented in this thesis is superior to both the approaches presented by Ren et al. in all the three metrics.

A ROC AUC score of 0.9680 for the Binary model was obtained. As a reference, a perfectly fitted model with correctly adjusted thresholds between each class would obtain a score of 1. Thereby, the ROC AUC score achieved by this model indicates that the threshold between the two classes

is descent. In contrast to accuracy, precision and recall, the MCC evaluates how well the model performed in *both* classes. It is measured proportionally, both to the size of positive elements, and the size of negative elements in the dataset. I.e. it takes all the elements of the confusion matrix into consideration instead of focusing on the positive class. The Binary model achieved an MCC of 0.9412 which indicates that the model is robust in both the Ice class and the class for Open Water.

The results obtained by the Binary model are relatively superior to what any of the multiclass classifiers achieved. However, this Model was implemented to assess how well the U-Net architecture differentiates between Open Water and Sea Ice, and had the advantage of only considering two classes. Having merged all the sea ice classes (where the uncertainty was relatively high compared to the Open Water class), it is natural that the model performs better.

Furthermore, it is noteworthy that the Open Water class in the binary dataset only represents the pixels labeled with up to 5% sea ice concentration. On the other hand, the Open Water class in the multiclass dataset represents a sea ice concentration up to 10%. The Binary model was evaluated with the ROC AUC score for several different thresholds between the two classes. A threshold at 5% generated the best ROC AUC score and this threshold was chosen for the binary dataset.

5.2 Qualitative Analysis of Results

In the case of image segmentation problems, it is reasonable to apply a visual evaluation of the results in addition to numerical measurements. Hence, a qualitative analysis of the results will be addressed in this section. The authors have selected five SAR scenes located in different parts of the Greenland coast which will participate in the visual interpretation of the results. These scenes are presented in Fig. 32. In the figure the NERSC noise corrected SAR HH and SAR HV images, and their corresponding polygon id charts produced by the sea ice expert are displayed, in the left, middle and right column respectively. The figure is included in this thesis to give the reader an understanding of how the ice charts are produced, and the complexity of this task.

Figure 32: The NERSC SAR HH, NERSC SAR HV and polygon id labeled ice charts present in the visualization data set.

5.2.1 Baseline - Multiclass Image Segmentation

The qualitative results obtained by applying the Baseline model on the visualization dataset is displayed in Fig. 33. The figure holds (respectively from left to right) 1) a concentration chart, 2) a multiclass ice chart and 3) the corresponding model prediction of the five visualization scenes. Throughout this section the images in the figure will be referenced as numbers from the top.

The first and most apparent tendency in the figure is that the model often mistake ice for Open Water. This is evident in the first four images, but most prominent in image number two. In this image there is a large area covered with Very Close Drift Ice in the lower right corner of the ice chart, in which the model has predicted to be Open Water. In the other images this tendency is shown as spots of Open Water in areas which are labeled with the ice in the true ice charts. This tendency is quantitatively supported by the elevated values in column 0 in the confusion matrix(Fig. 28).

Moreover, the model does not distinguish well between the intermediate classes. This is obvious in the first image in which the multitude of ice classes in the middle of the ice chart are mainly predicted to be Very Close Drift Ice. This is also evident the three next images, especially in image three where the large portion of Open Drift Ice is predicted to be Very Close Drift Ice. This is also supported by the increased values in column 4 in the confusion matrix.

For the last image on the other hand, the performance of the model is quite good. It is able to predict an ice chart very similar to the real one, except the small area close to land in the lower left corner. Given the subjectivity of the ice analysts and complexity of the ice charting, some areas can be prone to mistakes. Specifically, it can be difficult to label areas with high proximity to land. Therefore, there might be ice in areas labeled with water close to land in the ice charts.

The authors are not surprised by the poor visual performance of the Baseline model. This is due to the low scores in the numerical performance measures f1 and R^2, presented in Table 13. However, the model is evidently performing well in predicting Open Water. This explains the relatively high accuracy score, and functions as an example of how accuracy is not a suitable performance measure in this case.

5.2.2 Model 1 - Multiclass Image Segmentation

The predicted ice charts from Model 1 is given in Fig. 34. The layout of the images follows the same convention as in Fig. 33.

Firstly, it is evident that Model 1 performs better than the Baseline model in predicting ice in general. This improvement is evident in the four first images, where the predicted ice cover is more continuous and deprived of the Open Water spots present in the Baseline predictions. Secondly, there is a clear improvement in the prediction of the intermediate ice classes compared to the Baseline model. E.g. while the Baseline model, in the first image, predicted the large ice cover to be mainly Very Close Drift Ice, Model 1 was able to capture the transitions between the classes. These two improvements, encapsulated in Model 1 compared to the Baseline model, is also evident in the confusion matrices. In the confusion matrix of Model 1 in Fig. 29, the values are more evenly distributed along the diagonal than in the Baseline confusion matrix.

A drawback of Model 1 is highly apparent in the second and last image of Fig. 34. In these images it is clear that the model often mistakes Open Water for Very Close Drift Ice. This might be the cause of the low accuracy score obtained by Model 1 compared to the Baseline model. Moreover, this mistake is large as there are several classes between Open Water and Very Close Drift Ice. Thereby contributing to a significant reduction in the R^2 score of the model. Nevertheless, Model 1 achieves a tremendously improved R^2 score compared to the Baseline model, implying that there is a large improvement in the prediction of the intermediate classes. Moreover, a noise-like square pattern can be observed in the predictions of both the Baseline model and

Model 1. This will be further discussed in section 5.3.

For marine navigators, the consequences of predicting Very Close Drift Ice when in reality it is Open Water is less significant than predicting ice to be Open Water. Due to the fact that if they plan a route with these predicted ice charts as a basis they will not unexpectedly encounter ice where the model has predicted Open Water.

5.2.3 Model 2 - Multiclass Image Segmentation

The visual performance of the model trained on the imbalanced multiclass dataset is presented in Fig. 35, and will be discussed in this subsection.

Judging by visual performance, this is the best multiclass model presented in this thesis. It outperforms both the Baseline model and Model 1. This model outperforms the two other models in several ways. First of all, the model is better at detecting the ice classes in general. The ice charts predicted by the Baseline model had spots of Open Water in the continuous ice class covers, this model's predictions, however, are more similar to the real ice charts. Moreover, the white boundary artifacts that were present in the Open Water patches predicted by Model 1 in Fig. 34 are not present here. In general, Model 2 performs very well in predicting the Open Water class. This is clearly evident in all the images in the visualization dataset.

Regarding the ability to predict the intermediate classes, it is evident in the four first images that this model performs better than the two previous models. This fact is also mirrored by the diagonal pattern in the confusion matrix(Fig. 30) and the high R^2 score of Model 2 in Table 13. However, the predictions of this model is not perfect. In the lower right corner of image two it is evident that the model in fact does mistake Very Close Drift Ice to be Open Water. A physical interpretation of this mistake is that there can be melted water on top of the ice. The ice analysts might not bother to exclude these areas from the polygon as it is inside a larger portion of Very Close Drift Ice. The information of the presence of Open Water in areas like these is not very valuable for marine navigators as they tend to avoid them in the first place. In the SAR HV imagery for this ice chart displayed in Fig. 32, these areas does have a darker color in which is more similar to the color of Open Water, compared to the surrounding pixels. This might imply that there actually is water on top of the ice in these areas.

Moreover, in image three there is an unmistakable difference between the true ice chart and the predicted one. Specifically, the ice class cover in the middle of the image in the true ice chart is labeled as Open Drift Ice. The predicted ice chart however, labels the ice class cover more similarly to the concentration chart. In image number four it is evident that the model mistakes Fast Ice for Very Close Drift Ice in the ice covered region to the right. This is a minor mistake considering the fact that the Fast Ice class implies 100% ice while the Very Close Drift Ice is 90-95%. This fact is supported quantitatively by the elevated value for predicting the Very Close Drift Ice while the true class is Fast Ice in the confusion matrix of Model 2 in Fig. 30. In general, where the predictions from the model and the true labels do not agree the difference is small, usually one class.

5.2.4 Binary Model - Binary image segmentation

This subsection will regard the qualitative analysis of the Binary model which was trained on the binary dataset. The qualitative performance of this model is presented in Fig. 36.

Primarily, the model is able to predict the Open Water class very well, in which is especially evident in image three and five. In the second image nonetheless, there are some small areas of Open Water in the upper left area of the true ice chart which have been predicted to be Ice. Given the fact that ice in this dataset is considered to be every concentration larger than 0%, this might

not be a large mistake by the model. A similar tendency is evident in the prediction by Model 2 in the same image given in Fig. 35.

Moreover, in all the images it is conspicuous that the model is able to capture most of the edges between Open Water and larger portions of Ice. This is most prominent in image three, where the ice edge in the true ice chart is coinciding with the edge in the predicted chart. In the figure, however, one can observe that some edges and areas of ice has been predicted to be water which is not desirable. This is highly evident in the upper left part of image four where a large portion of ice has been predicted to be Open Water. In the concentration chart it is visible that this area contain pixels with a low concentration of sea ice. This decreases the significance of the misclassification in this area.

As in the predictions from Model 2, there are some areas located inside larger portions of Ice that has been predicted to be Open Water. This is prominent in image two, three and four, and might be due to the same reasons as for Model 2. In Fig. 31 on the other hand, it is evident that the amount of wrongly predicted ice pixels is still relatively low. And overall, this model performs very well in the qualitative analysis.

It can be argued that since the main focus of this thesis lies in aiding marine navigation, Open Water is the most important class. And merging the middle classes, thereby removing some uncertainty issues, will not affect the purpose of the sea ice charts. However, by including all values of sea ice concentration above zero in the ice class, there will evidently be a loss of information. An area containing only 5% which to some vessels is easy to passage through, will not be considered by the navigators since it is merged together with the other classes. This issue has to be considered when using the predicted ice charts.

For the application of marine navigation, it is evident in Fig. 36 that the predictions by the model could be used with caution. Nonetheless, the multiclass classification does yield more information for the navigator. This information enables the users of the ice charts to make educated decisions. For instance, to plan a route through an area covered by Very Open Drift Ice, knowing that it is possible to navigate around the ice present in these areas.

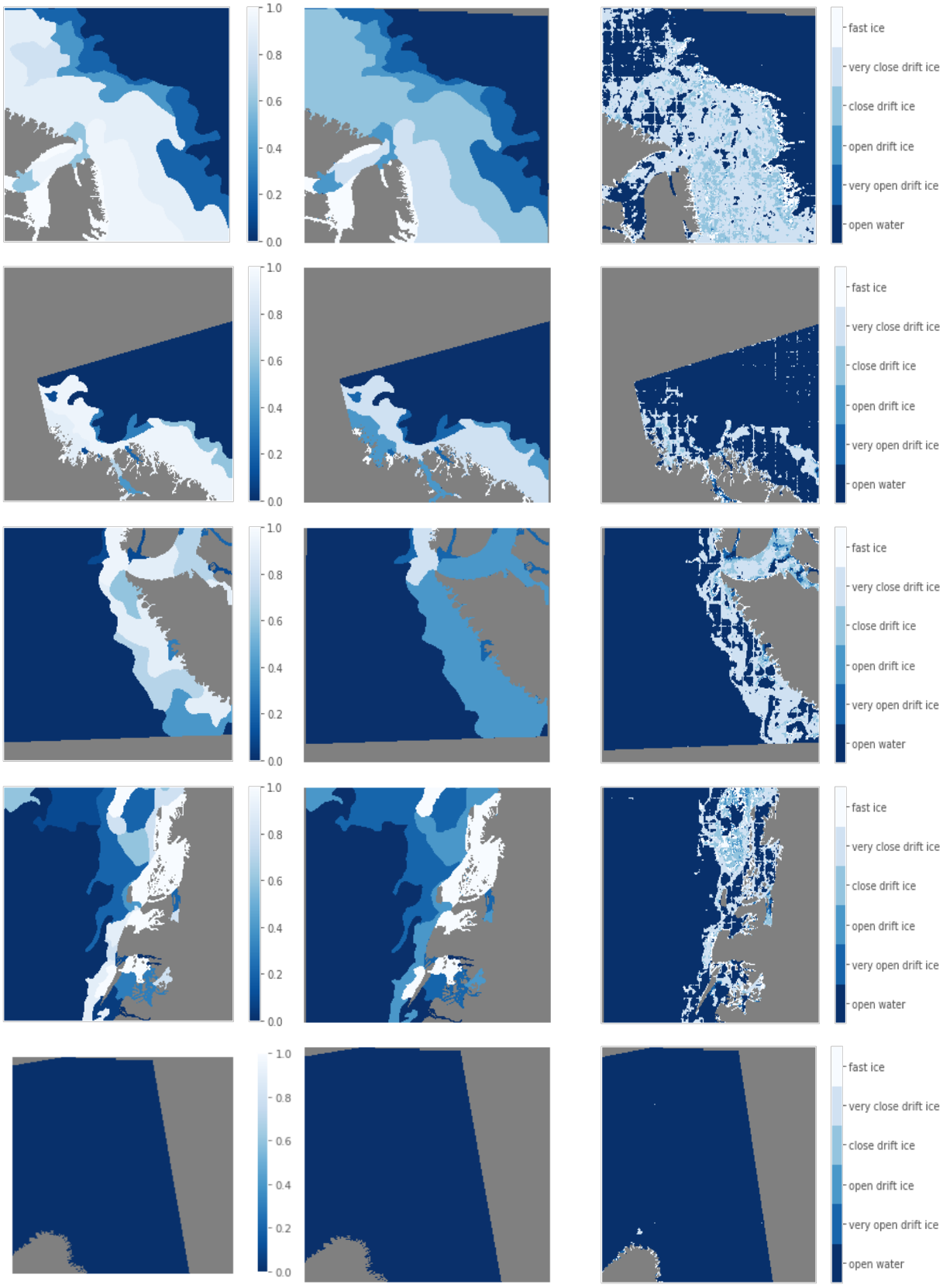


Figure 33: Visualization of the concentration ice chart, the true ice chart and the ice chart predicted by the Baseline model.

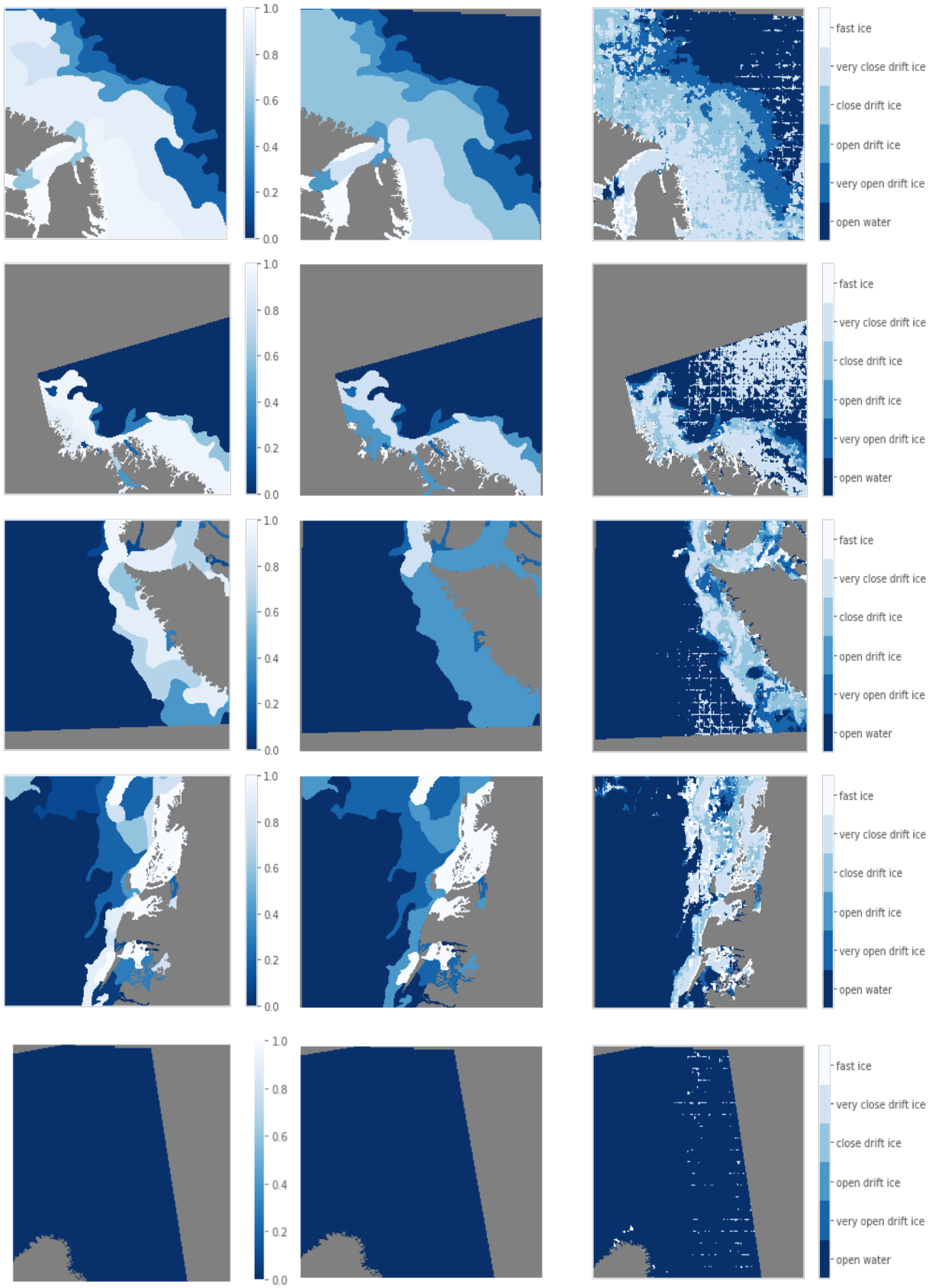


Figure 34: Visualization of the concentration ice chart, the ground truth ice chart and the ice chart predicted by Model 1

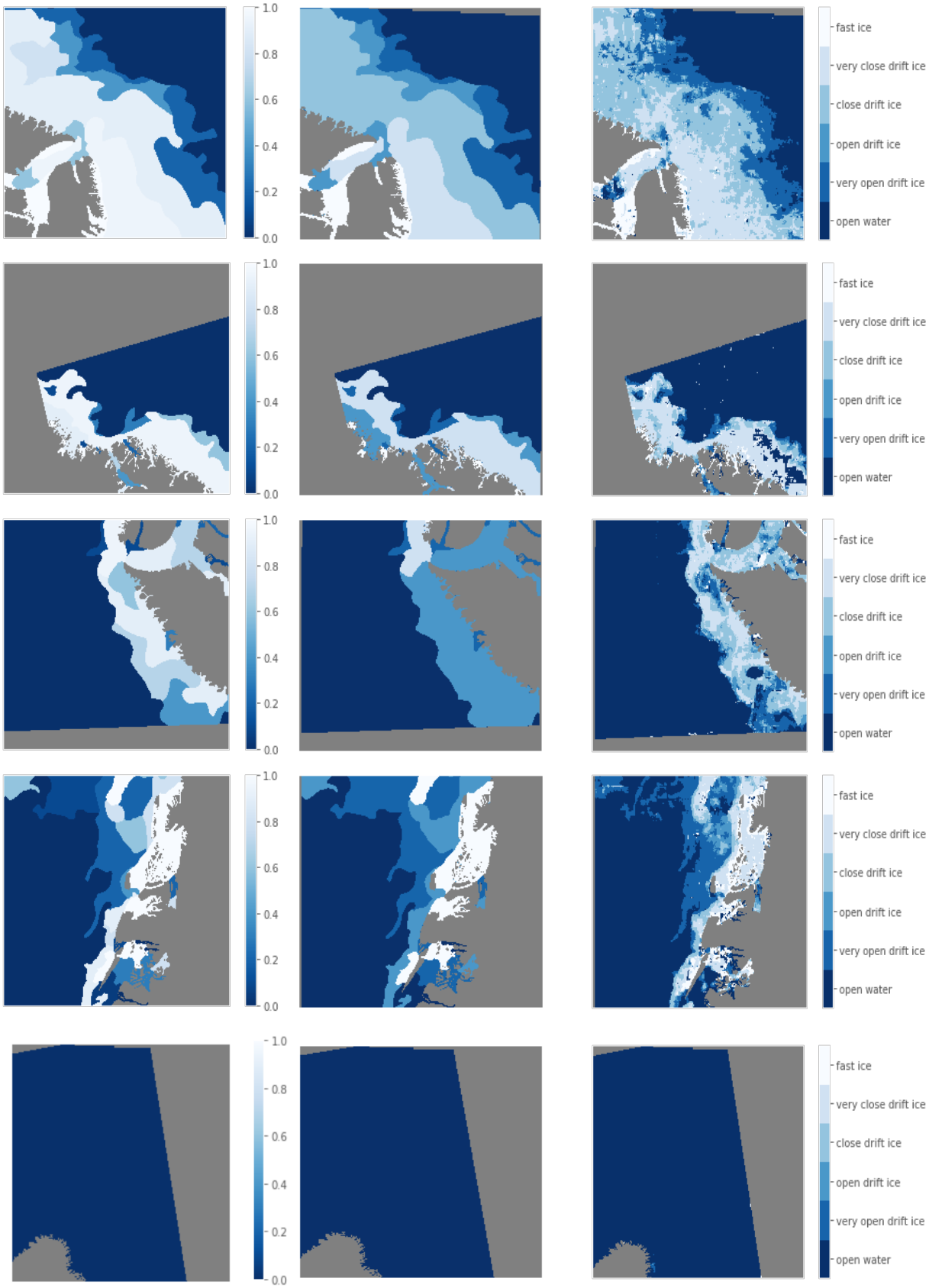


Figure 35: Visualization of the concentration ice chart, the true ice chart and the ice chart predicted by Model 2

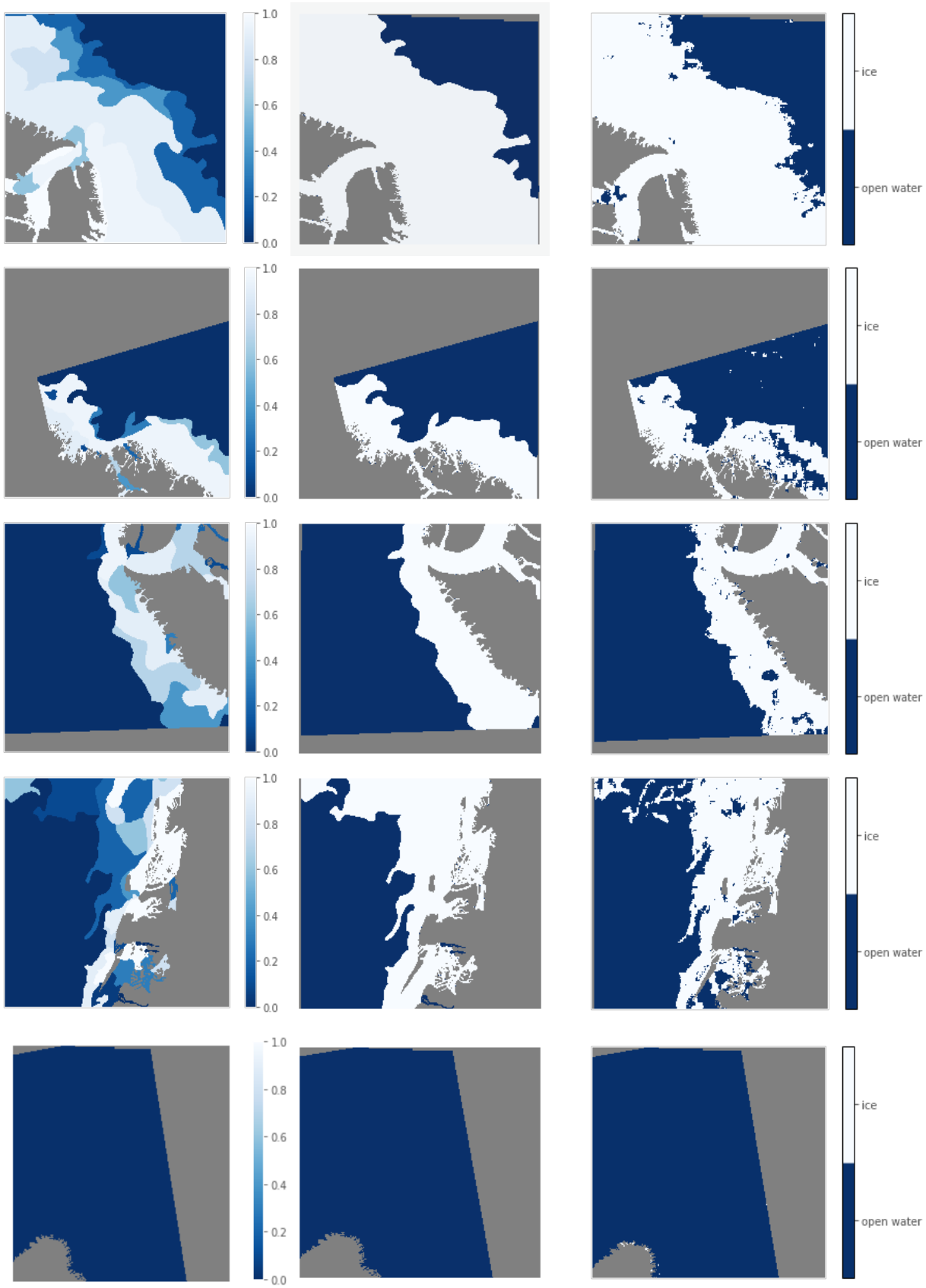


Figure 36: Visualization of the concentration ice chart, the ground truth ice chart and the ice chart predicted by the Binary model.

5.3 Further Discussion

In this section, aspects beyond the obtained results will be discussed. These regard the applied dataset and ice charts, the preprocessing, and the deep learning implementation.

The dataset utilized in this thesis originally contained 461 SAR scenes with a total size of 315 GB in a compressed zip-format. After several time-consuming attempts, the dataset was finally downloaded. However, in the extraction of the zipped nc-files only 240 SAR Scenes were retrieved. The zip file containing the dataset might have been damaged during download, making some files irretrievable. However, the downloaded subset still contained a large amount of data, and its seasonal and geographical distribution resembled the original distribution. Moreover, given that the downloading process was challenging, and the storage space on the remote server limited, the decision of using the incomplete dataset was made.

The sea ice concentration estimates in the ice charts are based on the subjective judgement of a sea ice analyst. Therefore, when training a network based on manually classified ice charts, some extent of subjectivity will always be present in the predictions. An uncertainty related to this subjectivity is a common denominator for all networks trained on manually labeled ice charts. Hence, the charts applied in the deep neural networks in this thesis have no explicit associated uncertainty. It is not trivial to measure this degree of uncertainty. However, studies have shown that ice charts covering the same area, but produced by different National Ice Centres vary significantly [32]. This makes it hard to compare the results obtained in this thesis to the results obtained in other studies based different datasets. One approach for addressing this challenge could be to train a model on ice charts produced by several different Ice centres.

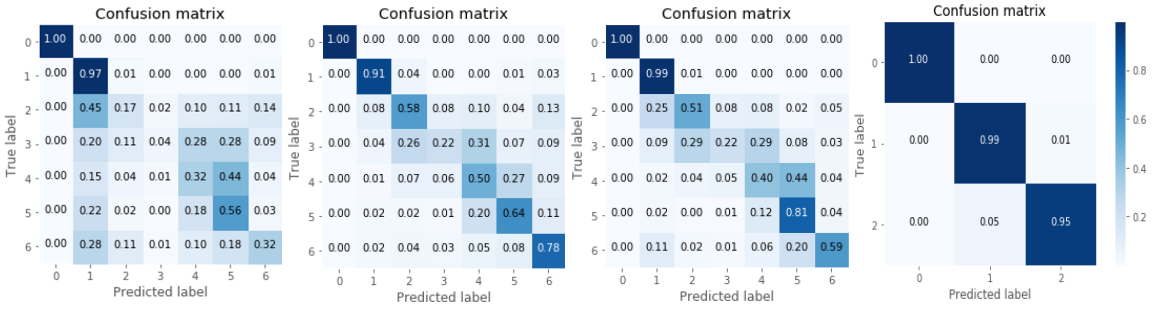
Another aspect related to the ice charts is the varying degree of uncertainty in each ice class. This is reflected in the results of this thesis. The models trained on the multiclass datasets performed worse on the intermediate classes than the classes for Open Water and Fast Ice. This is why the binary classifier achieved better accuracy. Moreover, ice analysts commonly pay more attention to the classification of pixels close to the "ice/water"-edges. This is discussed in the manual for the ASIP dataset referenced in section 3.3, and is stated to be because the edge-cases are the most important areas for the marine navigators. Moreover, since there is no navigation inside the polygons characterized as ice, these are not prioritized. This is reflected in the predictions made by the models in this thesis.

The class partitioning utilized in this thesis correspond to the classes applied by the Norwegian Meteorological Institute. This choice was natural as it is the convention Norway utilizes. Moreover, by reducing the number of sea ice concentration intervals (classes) from 11 to 6, the associated uncertainty related to the intermediate intervals was reduced. However, the concatenation of the intervals comes at the cost of information included in the chart. Provided with enough data, a model could be able to distinguish more classes. Given the amount of data available in this thesis, distinguishing eleven intervals would be difficult for the model, and the decision of using less classes was made.

When comparing Model 1 to the Baseline after training in Table 12, it is evident that Model 1 achieved higher training accuracy and lower training loss. On the other hand, the validation accuracy was lower and the validation loss higher than in the Baseline. This was unexpected. However, by inspecting the variation of the validation metrics during training, it was evident that they were highly volatile for both the Baseline model and Model 1. This may be because the optimizers are selecting a random subset of the validation data in validation. Thereby, there is a chance that the distribution of the validation data is different from epoch to epoch. If the random subset, selected from the validation dataset, for one epoch contains a high amount of the class in which the model is good at predicting, the accuracy will be high. While, if the subset contain more of the other classes in the next epoch, the accuracy will be low. This volatility may therefore be the reason for the Baseline's high validation accuracy and low validation loss, compared to Model 1.

Instead of deleting the missing values, they were handled like a separate class, class 0. This decision was made based on the desire to maintain the size and location compliance between the different

data in the dataset. It was crucial that the detection of the Missing Values class did not affect the performance in detecting the other classes. As can be interpreted from the confusion matrices given in 37, none of the models misinterpreted a missing value. This is evident in each confusion matrix where there are zero values for all the elements in the first row and column except in the (0,0) position.



Baseline Model 1 Model 2 Binary

Figure 37: The confusion matrices of the models evaluated with the test dataset, where the missing values class is included.

Since the SAR scenes are representing snapshots of the Arctic Sea in which water is the main element, there is a natural imbalance in the class distribution. As referenced in section 4.2.4, an unevenly distributed dataset may cause the model to perform better in the over represented classes compared to the underrepresented ones. Hence, the authors attempted to balance the dataset in order to improve the performance in all the classes. The multi-class dataset in this thesis was balanced by undersampling patches of the water class and augmenting patches of the ice classes. Thus, the class distribution was evened. However, this balancing approach did not seem to be appropriate for this problem. A reson for this may be that it also contributed to an unrealistic class distribution in which the model would expect to see in the test set as well.

The models trained on the balanced dataset contain noise in terms of of square patterns. More specifically, there are white boundary effects around the patches in some of the images in Fig. 33 and 34. The authors of this thesis hypothesize that there is a combined reason for this. First, due to issues related to storage space on the server, the patches were not overlapping. This incurred a larger uncertainty in the edges of each patch since these pixels had been evaluated less times than the pixels in the center of the patches. Secondly, the classification of the ambiguous pixels could be affected by the distribution of the training dataset. I.e. it would attempt to compensate for the skewed class distribution in the visualization SAR scenes (which are not balanced) by labelling the pixels of high uncertainty with ice.

In deep learning, the choice of network is highly dependent on the problem at hand. In this thesis the U-Net structure was utilized. This choice was made because U-Net is a well-known structure for medical image segmentation in which resemble the problem of sea ice segmentation. Moreover, in both cases there is a lot of "background"-values (Open Water in our case). Finally, U-Net is an acknowledge architecture for sea ice segmentation using remote sensing.

It is not trivial to measure how the noise correction developed by NERSC has influenced the results in this thesis. Only the images corrected for noise with the NERSC-method were applied. Hence, there exists no tangible evidence supporting that the models developed in this thesis benefit from the improved noise correction. However, there are several studies supporting the statement that the noise correction provided by ESA is insufficient, and that there still is a significant amount of noise left in the cross-polarized S1 images [18], [33], [19], [34] and [24]. Moreover, the noise correction method developed by NERSC is highly acknowledged in its research field and has been further developed by Sun and Li [18] and Lee et al.[33]. Hence, the authors of this thesis argue that there is a high probability that the results have benefited from the NERSC noise correction.

Chapter 6

Conclusions and Further Work

The monitoring of Arctic sea ice is crucial for both marine navigation and global warming. There are mainly two requirements related to this monitoring. The data has to be up-to-date, and accurate. Today, sea ice information from these areas is provided by the national sea ice services of the states within the Arctic circle. This information is generated by sea ice analysts which interpret satellite images to manually draw polygon ice charts. These charts are difficult and time consuming to make, and simultaneously influenced by human subjectivity. Therefore, it is desirable to develop an automatic approach for the generation of high quality sea ice charts, swiftly after acquisition of the satellite imagery.

In this thesis, a comprehensive study of automatic approaches for sea ice segmentation has been presented. Two different versions, including a multiclass- and a binary version, of the ASIP dataset were generated. Furthermore, a balancing approach was implemented and applied to the multiclass dataset. The resulting datasets were utilized in the training process of four separate instances of the U-Net. The performance of the multiclass classifiers was evaluated and compared prior to a separate evaluation of the binary classifier.

Both sea ice segmentation models trained on the balanced multiclass dataset were outperformed by the model trained on the imbalanced one. Having a relatively high learning rate, and a basic optimizer in which did not adjust the learning rate during training, the Baseline model achieved the lowest performance. Its values for accuracy, F1-score and R^2-Score reached respectively 0.7126, 0.3992 and 0.1281. This was also reflected in the qualitative evaluation. The improved model trained on the balanced dataset achieved a lower accuracy than the Baseline model. However, with an F1-score of 0.4843 and an R^2-score of 0.5407, it was superior to the Baseline model. This improvement is additionally evident in the qualitative results.

The model trained on the imbalanced dataset, obtained an R^2-score of 0.8709 which makes it the best model to find an appropriate fit for the true labels in the dataset. I.e. the data points has 87.09% less variation around the function encapsulated by the model, than around the simple mean of the data points. These results are comparable to similar studies, and are considered good given the uncertainties present in the dataset. The high performance of this model is also evident in the qualitative results. It can thereby be concluded that this is the best suited multiclass model for the automatic generation of ice charts presented in this thesis.

Moreover, the authors conclude that the balancing approach applied to the multiclass dataset did not improve the model performance, and is therefore not suited for this application. Presumably, this is due to the insufficient amount of supplementary information added to the underrepresented classes by the augmentation approach. Another reason can be that the balanced models were trained and tested on datasets with different class distributions. In contrast, the imbalanced model was trained and tested on data with the same distribution.

In an eventual future work, another balancing approach should be applied to the dataset. Spe-

cifically, Syntetchic Minority Otverasmpling Technique (SMOTE) [35] should be explored. This balancing approach randomly chooses an instance of the minority class and finds its k nearest neighbours. Thereafter, it generates synthetic instances by combining one of the k nearest neighbours with the chosen instance in a convex combination. SMOTE is slightly more sophisticated than the balancing method applied in this thesis. It would therefore be interesting to explore its effects on the results of the balanced U-Net instances.

The model trained on the binary dataset obtained promising results. With an accuracy of 0.9800, a precision of 0.9819, and recall of 0.9782, it outperformed the results obtained in other similar studies. Moreover, the model obtained an MCC score of 0.9412 in which indicates good performance in both the prediction of Open Water and Ice. The confusion-matrix representing the performance of the binary model also showed that 99% of the true water pixels were correctly classified. It was expected that the binary model would perform better than the models trained on the multiclass datasets. This was because the intermediate sea ice concentration classes, in which possess the largest uncertainty, were merged. It could be interesting, in the future, to apply the predictions made by the binary model as additional input for a multiclass U-Net.

In the future, it could be interesting to apply the imbalanced dataset with other similar architectures. Several variations of the U-Net exist. Among others, U-Net3+ combines multi-scale features which could be relevant for the implementation of support for AMSR2 data [36]. Moreover, another version of the U-Net architecture, the HED U-Net, was proposed in [37]. This archtitecture combines semantic segmentation and an edge detection framework (HED). Since the edge-cases are especially important for marine navigators, this method could interesting to apply on the ASIP dataset.

The SAR imagery applied in the U-Nets were corrected for noise with NERSC's method presented in [19] and [20]. Further developments of the noise correction method applied in this thesis have already been published [18], [33]. Since they were published after the authors of this thesis started searching for data, they were not considered. However, in the future, it could be interesting to apply one of these noise corrections to the SAR scenes in the ASIP dataset for comparison with the results obtained in this thesis.

In deep learning, the amount of available data is a critical factor for the model performance. Only 240 out of the 461 SAR scenes were included in the work of thesis. One of the first priorities in an extention of this work work will be to download the complete dataset. Moreover, it would be interesting to apply the AMSR2 data provided in the ASIP dataset together with the SAR imagery in a data fusion by taking inspiration from Malmgren-Hansen et al. [29]. This supplement could especially aid the network in situations where there are ambiguity issues related to sea ice and open water in wind-roughened seas. Since they both appear bright in these cases, the temperature measurements of the sea surface provided by the AMSR2 data could help the network to classify the pixels.

Moreover, as described in section 2.2, the ice charts contain a great amount of supplementary information about the ice characteristics besides the concentration. This information could be used to extend the models presented in this thesis, or to train other sophisticated models. For instance, a classifier could be trained to determine the stage of development of the ice, e.g. new ice, first year ice, multi-year ice or thin first year ice. This could be applied to analyse the amount of relatively new ice in the more recent years compared to previous years, and provide information about the effect of global warming on this type of ice.

The class partitioning for the multiclass classification in this thesis was based on the convention utilized by the Norwegian Meteorological Institute. In a continuation of this work it could interesting to investigate the effect of using other classes. For instance, the number of classes could be set to 11, which is the same number as the number of concentration intervals used in the original polygon ice charts. For the binary classification, the ROC AUC score was applied to determine the optimal threshold between the classes. A similar approach could be adopted for the multiclass case.

To sum up, in this thesis several modifications of the automatic approach for sea ice segmentation

based on the U-Net architecture were developed. Three different versions of the ASIP dataset were applied, including one binary- and two multiclass versions, in which one was balanced. A Python software package for implementing the solutions was produced. The multiclass model that obtained the best performance was trained on the imbalanced dataset. Therefore, it was concluded that the applied balancing approach was insufficient, and that there is a need for a more sophisticated method. The binary model proved to be competitive with other state-of-the-art binary sea ice segmentation approaches. The noise correction developed by NERSC increased the image quality, especially in the cross-polarized S1 SAR images. Hence, there is reason to believe that this enhancement improved performance of the models. In a potential continuation of this work, it would be beneficial to exploit the data provided in the ASIP dataset to a larger extent. And it would be interesting to apply other variations of the U-Net as well as the advanced variations of the noise correction. In addition, a more sophisticated balancing approach should be explored. During the work for this thesis, the authors have gained a large amount of knowledge in the field of remote sensing and deep learning.

Bibliography

[1] S. I. Hay. 'An overview of remote sensing and geodesy for epidemiology and public health application'. In: *Advances in Parasitology* 47.May 2000 (2000), pp. 1–35. ISSN: 0065308X. DOI: 10.1016/s0065-308x(00)47005-3.

[2] P. R. Teleti and A. J. Luis. 'Sea Ice Observations in Polar Regions: Evolution of Technologies in Remote Sensing'. In: *International Journal of Geosciences* 04.07 (2013), pp. 1031–1050. ISSN: 2156-8359. DOI: 10.4236/ijg.2013.47097.

[3] F. A. Ifremer, M. Brandt and K. Dmi. 'For OSI TAC Sea Ice products'. In: Dmi (2019), pp. 1–70.

[4] D. Haverkamp, C. Tsatsoulis and S. Gogineni. 'The combination of algorithmic and heuristic methods for the classification of sea ice imagery'. In: *Remote Sensing Reviews* 9.1-2 (1994), pp. 135–159. DOI: 10.1080/02757259409532219. URL: https://doi.org/10.1080/02757259409532219.

[5] D. Haverkamp, L. K. Soh and C. Tsatsoulis. 'A comprehensive, automated approach to determining sea ice thickness from SAR data'. In: *IEEE Transactions on Geoscience and Remote Sensing* 33.1 (1995), pp. 46–57. DOI: 10.1109/36.368223.

[6] L. K. Soh, C. Tsatsoulis, T. Bowers and A. Williams. 'Representing sea ice knowledge in a Dempster-Shafer belief system'. In: *IGARSS '98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No.98CH36174)*. Vol. 4. 1998, 2234–2236 vol.4. DOI: 10.1109/IGARSS.1998.703797.

[7] D. A. Clausi, A. K. Qin, M. S. Chowdhury, P. Yu and P. Maillard. 'MAGIC: MAp-Guided Ice Classification System'. In: *Canadian Journal of Remote Sensing* 36.sup1 (2010), S13–S25. DOI: 10.5589/m10-008. eprint: https://doi.org/10.5589/m10-008. URL: https://doi.org/10.5589/m10-008.

[8] S. Ochilov and D. A. Clausi. 'Operational SAR Sea-Ice Image Classification'. In: *IEEE Transactions on Geoscience and Remote Sensing* 50.11 (2012), pp. 4397–4408. DOI: 10.1109/TGRS.2012.2192278.

[9] J. Lohse, A. P. Doulgeris and W. Dierking. 'An Optimal Decision-Tree Design Strategy and Its Application to Sea Ice Classification from SAR Imagery'. In: *Remote Sensing* 11.13 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11131574. URL: https://www.mdpi.com/2072-4292/11/13/1574.

[10] A. Buetti-Dinh et al. 'Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition'. In: *Biotechnology Reports* 22 (2019), e00321. ISSN: 2215017X. DOI: 10.1016/j.btre.2019.e00321. URL: https://doi.org/10.1016/j.btre.2019.e00321.

[11] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev and N. Chervyakov. 'Application of the residue number system to reduce hardware costs of the convolutional neural network implementation'. In: *Mathematics and Computers in Simulation* 177 (2020), pp. 232–243. ISSN: 0378-4754. DOI: https://doi.org/10.1016/j.matcom.2020.04.031. URL: https://www.sciencedirect.com/science/article/pii/S0378475420301580.

[12] O. Ghorbanzadeh et al. 'Evaluation of Different Machine Learning Methods and Deep-Learning Convolutional Neural Networks for Landslide Detection'. In: *Remote Sensing* 11.2 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11020196. URL: https://www.mdpi.com/2072-4292/11/2/196.

[13] H. Boulze, A. Korosov and J. Brajard. 'Classification of Sea Ice Types in Sentinel-1 SAR Data Using Convolutional Neural Networks'. In: *Remote Sensing* 12.13 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12132165. URL: https://www.mdpi.com/2072-4292/12/13/2165.

[14] L. Wang, A. Scott, L. Xu and D. Clausi. 'Sea Ice Concentration Estimation During Melt From Dual-Pol SAR Scenes Using Deep Convolutional Neural Networks: A Case Study'. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.8 (2016), pp. 4524–4533. DOI: 10.1109/TGRS.2016.2543660.

[15] R. Kruk, C. Fuller, A. Komarov, D. Isleifson and I. Jeffrey. 'Proof of Concept for Sea Ice Stage of Development Classification Using Deep Learning'. In: *Remote Sensing* 12.15 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12152486. URL: https://www.mdpi.com/2072-4292/12/15/2486.

[16] O. Ronneberger, P. Fischer and T. Brox. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells and Alejandro F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[17] A. Singh, H. Kalke, M. Loewen and N. Ray. 'River Ice Segmentation With Deep Learning'. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.11 (Nov. 2020), pp. 7570–7579. ISSN: 1558-0644. DOI: 10.1109/tgrs.2020.2981082. URL: http://dx.doi.org/10.1109/TGRS.2020.2981082.

[18] Y. Sun and X. Li. 'Denoising Sentinel-1 Extra-Wide Mode Cross-Polarization Images over Sea Ice'. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.3 (2021), pp. 2116–2131. ISSN: 15580644. DOI: 10.1109/TGRS.2020.3005831.

[19] J. Park, A. Korosov, J. Won, M. Babiker and S. Sandven. 'Efficient Thermal Noise Removal for Sentinel-1 TOPSAR Cross-Polarization Channel'. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.3 (2018), pp. 1555–1565. ISSN: 01962892. DOI: 10.1109/TGRS.2017.2765248.

[20] J. Park, J. Won, A. Korosov, M. Babiker and N. Miranda. 'Textural noise correction for sentinel-1 TOPSAR cross-polarization channel images'. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.6 (2019), pp. 4040–4049. ISSN: 15580644. DOI: 10.1109/TGRS.2018.2889381.

[21] A. Lwin. 'GEOMORPHOLOGICAL MAPPING WITH RESPECT TO AMPLITUDE, COHERENCEAND PHASE INFORMATION OF ERS SAR TANDEM PAIR'. In: 2008.

[22] W Dierking. 'Dierking2013'. In: 26.2 (2013), pp. 100–111. URL: https://tos.org/oceanography/article/sea-ice-monitoring-by-synthetic-aperture-radar.

[23] D. Hong and C. Yang. 'Automatic discrimination approach of sea ice in the Arctic Ocean using Sentinel-1 Extra Wide Swath dual-polarized SAR data'. In: *International Journal of Remote Sensing* 39.13 (2018), pp. 4469–4483. ISSN: 13665901. DOI: 10.1080/01431161.2017.1415486. URL: https://doi.org/10.1080/01431161.2017.1415486.

[24] U. Balss, H. Breit and T. Fritz. 'Noise-related radiometric correction in the TerraSAR-X multimode SAR processor'. In: *IEEE Transactions on Geoscience and Remote Sensing* 48.2 (2010), pp. 741–750. ISSN: 01962892. DOI: 10.1109/TGRS.2009.2035443.

[25] D. Chicco and G. Jurman. 'The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation'. In: *BMC Genomics* 21.1 (2020), pp. 1–13. ISSN: 14712164. DOI: 10.1186/s12864-019-6413-7.

[26] S. Boughorbel, F. Jarray and M. El-Anbari. 'Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric'. In: *PLoS ONE* 12.6 (2017), pp. 1–17. ISSN: 19326203. DOI: 10.1371/journal.pone.0177678.

[27] B. Krawczyk. 'Learning from imbalanced data: open challenges and future directions.' In: (2016). DOI: 10.1007/s13748-016-0094-0. URL: https://link.springer.com/article/10.1007%5C%2Fs13748-016-0094-0.

[28] S. Charmchi. 'Optimized U-Net for Left Ventricle Segmentation by Sadegh Charmchi A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science Department of Computing Science University of Alberta'. In: (2018).

[29] D. Malmgren-Hansen et al. 'A Convolutional Neural Network Architecture for Sentinel-1 and AMSR2 Data Fusion'. English. In: *I E E E Transactions on Geoscience and Remote Sensing* 59.3 (2021), pp. 1890–1902. ISSN: 0196-2892. DOI: 10.1109/TGRS.2020.3004539.

[30] Y. Ren, H. Xu, B. Liu and X. Li. 'Sea Ice and Open Water Classification of SAR Images Using a Deep Learning Model'. In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. 2020, pp. 3051–3054. DOI: 10.1109/IGARSS39084.2020.9323990.

[31] Y. Ren, X. Li, X. Yang and H. Xu. 'Development of a Dual-Attention U-Net Model for Sea Ice and Open Water Classification on SAR Images'. In: *IEEE Geoscience and Remote Sensing Letters* (2021), pp. 1–5. DOI: 10.1109/LGRS.2021.3058049.

[32] J. Karvonen, J. Vainio, M. Marnela, P. Eriksson and T. Niskanen. 'A Comparison Between High-Resolution EO-Based and Ice Analyst-Assigned Sea Ice Concentrations'. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8.4 (2015), pp. 1799–1807. DOI: 10.1109/JSTARS.2015.2426414.

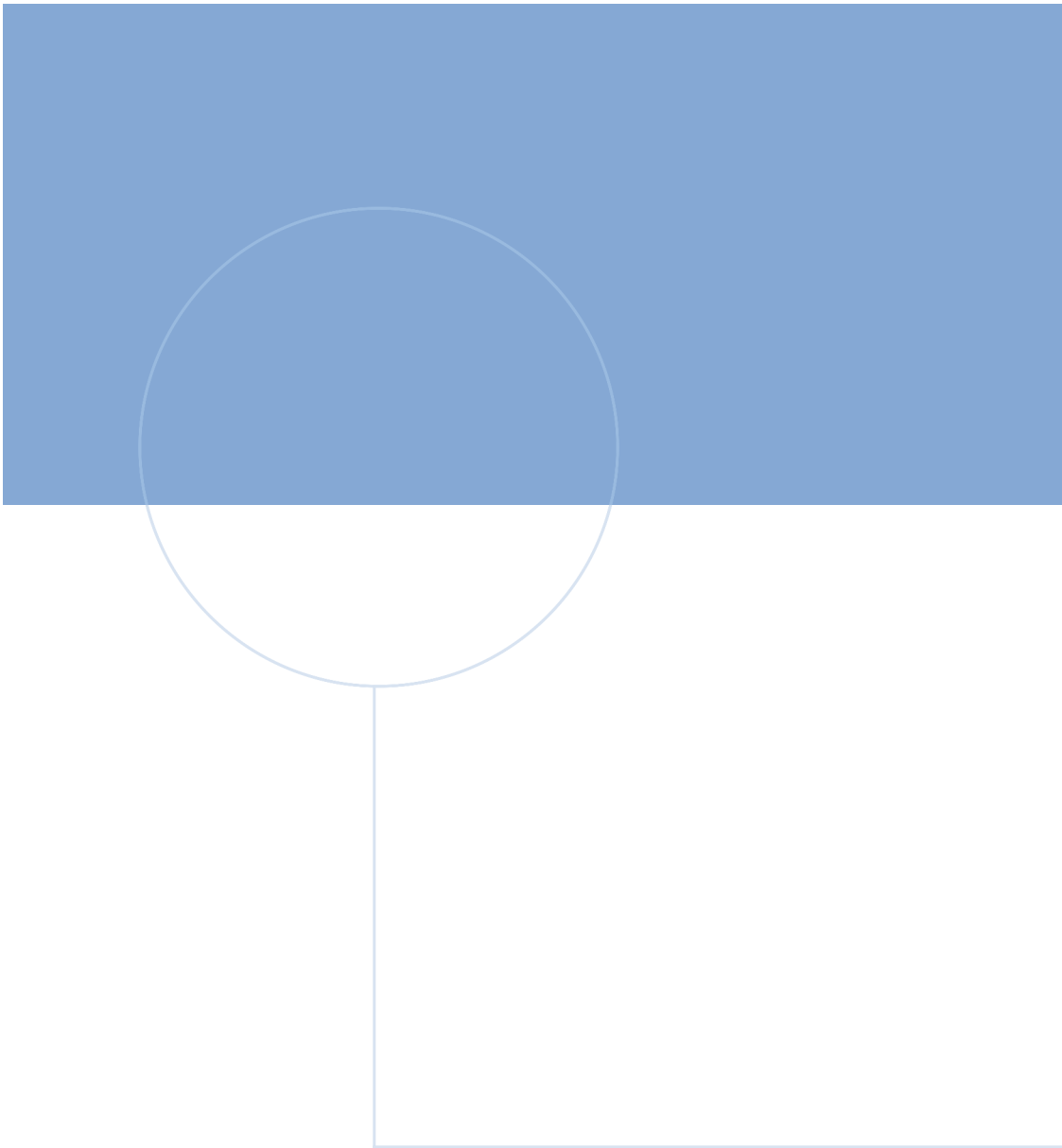
[33] P. Q. Lee, L. Xu and D. A. Clausi. 'Sentinel-1 additive noise removal from cross-polarization extra-wide TOPSAR with dynamic least-squares'. In: *Remote Sensing of Environment* 248.October 2019 (2020), p. 111982. ISSN: 00344257. DOI: 10.1016/j.rse.2020.111982. URL: https://doi.org/10.1016/j.rse.2020.111982.

[34] Karvonen. 'Baltic sea ice concentration estimation using SENTINEL-1 SAR and AMSR2 microwave radiometer data'. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.5 (2017), pp. 2871–2883. ISSN: 01962892. DOI: 10.1109/TGRS.2017.2655567.

[35] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer. 'SMOTE: synthetic minority over-sampling technique'. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[36] H. Huang et al. 'UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation'. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2020-May.iii (2020), pp. 1055–1059. ISSN: 15206149. DOI: 10.1109/ICASSP40776.2020.9053405. arXiv: 2004.08790.

[37] K. Heidler, L. Mou, C. Baumhoer, A. Dietz and X. X. Zhu. 'HED-UNet: Combined Segmentation and Edge Detection for Monitoring the Antarctic Coastline'. In: *IEEE Transactions on Geoscience and Remote Sensing* (2021), pp. 1–14. DOI: 10.1109/TGRS.2021.3064606.



NTNU

Norwegian University of
Science and Technology