Aalborg University, Copenhagen
Geoinformatics

Classification of ocean objects detected in Sentinel-1 SAR, Iceberg-Ship discrimination

Master of Science (MSc) in Technology (Surveying and Planning)
04 June 2020
Page Count: 45

Written By
Frederik Hass

Supervisor
Jamal Jokar Arsanjani

Abstract

Synthetic aperture radar (SAR) is an established part of ocean surveillance, with capabilities of detecting oil spills, icebergs, and marine traffic both day and night and regardless of weather. Detecting unidentified ocean objects, such as ships, is very relevant to authorities and interested parties, since these could be ships sailing illegally or be an iceberg on collision course with important structures.
The detection of ocean objects in SAR relies on well-established methods, mostly adaptive thresholding algorithms. In most waters, the object is probably a ship, allowing for automatic detection, whereas in arctic waters the vast majority of objects are icebergs drifting in the ocean. Because the objects can look very much a like in SAR images, the determination of what objects actually are, still rely on manual work and human interaction. With the great amount of ice in the waters around Greenland, this issue makes is difficult to detect ships that are not identifying themselves.

The livelihood of traditional Arctic communities is still very dependent on local natural resources, making the economies fragile and exposed to overexploitation of the very resources they depend upon. The United Nations 17 Sustainable Development Goals (SDG) aims at securing social, environmental, and economic developments while preserving and restoring our psychical environments.
The great sizes and vast emptiness of the arctic is making it difficult to monitor ocean traffic, which is still heavily dependent on volunteering reporting, exposing areas to unregulated fishing and illegal dumping of waste. SDG 14 "Life Below Water" is directly aimed at preserving ocean environments and preventing overexploitation of the oceans resources, this project is addressing target 14.4 and 14.C, preventing overfishing and enforcing maritime laws. With increasing interest and accessibility of the arctic regions, these goals can only be met with proper monitorization of ocean traffic, and satellite data is having a crucial role here.

The project aim to explore and expand the capabilities of ocean monitoring, by developing a methodology for discriminating between icebergs and ships. The objects in question, will be detected in dual polarization Sentinel-1 IW data in the waters surrounding western Greenland, with the further approach to train a deep learning algorithm to perform a discrimination, classifying an object as either a ship of an iceberg.
The training data for the deep learning algorithm will be objects detected in waters without icebergs, and objects detected where there is a great presence of icebergs. The model is validated with ship AIS data and iceberg data from the Danish Meteorological Institute.

Resumé

Brugen af radar-satellit (SAR) data er en fastforankret del af daglig maritim overvågning, da det i radar data er muligt at observere oliesplid, isbjerge og den generelle maritime trafik, dag og nat og uafhængig af vejret. Overvågning af uidentificerbare objekter på havene er af yderst relevans for myndigheder og andre interessenter, da disse kan være skibe engageret i ulovlig aktivitet eller isbjerge med kollisionskurs mod dyrebare konstruktioner. Detektering af sådanne objekter med SAR data bygger på veludviklede metoder, primært bestående af algoritmer som ud fra adaptive grænseværdier lokalisere objekter. De fleste steder vil et objekt på havet med størst sandsynlighed være et skib, hvorimod det i arktiske egne sandsynligvis vil være et isbjerg. I SAR data vil disse to objekter ofte have samme udstråling og udformning, hvilket gør at der stadig er behov for menneskelig validering for faktisk at bestemme hvilken type objekt der er tale om. Med så store mængder is i vandet som der ses omkring Grønland, gør denne problemstilling det vanskeligt at lokalisere skibe som ikke udsender nogen form for identificering.

En stor del af de traditionelle arktiske samfund afhænger stadig af lokale naturressourcer, hvilket eksponerer deres økonomi og levebrød for overudnyttelse af naturens ressourcer. De Forenedes Nationernes 17 Verdensmål retter sig mod at sikre social, miljømæssig og økonomisk udvikling alt imens det fysiske miljø bevares og beskyttes.
Grundet dets store størrelse og svær tilgængelighed er det i dag vanskeligt at overvåge den maritime trafik i Arktis, og overvågningen afhænger derfor i dag i høj grad af frivillig indrapportering, hvilket muliggør ureguleret fiskeri og ulovlig afskaffelse af affald. Verdensmål nr. 14 "Livet I Havet" retter sig direkte mod at bevare havmiljøet og sikre at der ikke sker overudnyttelse af havets ressourcer, dette projekt adresserer delmål 14.4 og 14.C som omhandler effektiv regulering af fiskeri samt implementering af international maritim lov. For at nå disse mål er det nødvendigt med effektiv overvågning af den maritime trafik, og her spiller satellit data en afgørende rolle.

Projektet retter sig mod at undersøge og udvikle metoderne indenfor maritim overvågning, ved konkret at udvikle en metode til at differentiere mellem isbjerge og skibe. De to objekttyper vil blive kortlagt ved brug af dobbelt polariserings Sentinel-1 IW-data i havene omkring det vestlige Grønland, for derefter at blive brugt i træningen af en deep learning algoritme som skal kunne klassificere de to objekter.
Træningsdata til algoritmen vil bestå af objekter detektering i områder uden isbjerge samt områder hvor der findes store mængder isbjerge. Den endelige model vil blive valideret ved brug af skibspositions AIS data samt data over isbjerge fra Dansk Meteorologisk Institut.

Preface

This masters thesis has been carried out over the period of 4 months, from start February to start June 2020. The thesis is the final project at the Master of Science in Geoinformatics at Aalborg University Copenhagen.
The overall focus of the project is to implement state of the art technology in mapping and monitoring of arctic environments, and covers one aspects of a fast developping and exiting future that is facing the Arctic. The project was made and written in Nuuk Greenland, at the office of **Asiaq - Greenland Survey**, i wish to acknowledge them for hosting me and facilitating the project.
I also wish to sincerely thank Evelina Mikneviciute and **COWI** for providing continuous assistance and interests in the project, and thank Jørgen Buus-Hinkler from **DMI** for sharing his knowledge within the field and providing important data for the project. Also thanks to **Gatehouse** in Aalborg for providing data crucial for the project to succeed.

Lastly i wish to acknowledge and thank my supervisor Jamal Jokar Arsanjani, not only for supervising this thesis, but for supervision and cooperation throughout my whole master.

Nuuk, Grønland. 4 Juni 2020

Frederik Hass

Contents

Abstract i

Resumé ii

Preface iii

Abbreviations vi

List of Figures vii

List of Tables viii

1 Introduction 1

1.1 Problem statement and research questions . . . 3

2 Background 4

2.1 Sentinel-1 . . . 4

2.2 Current Detection Methods . . . 6

2.3 Deep Learning . . . 8

2.4 Deep learning in remote sensing . . . 10

3 Methodology 13

3.1 Study Areas . . . 13

3.1.1 Denmark . . . 14

3.1.2 Greenland, Nuup Kangerlua . . . 14

3.1.3 Greenland, Disko Bay . . . 15

3.1.4 Summary . . . 16

3.2 Data Acquisition and Pre-processing . . . 17

3.3 Object localisation . . . 18

3.3.1 CFAR Iceberg Detection . . . 18

3.3.2 Iceberg Validation . . . 19

3.3.3 Ship localisation . . . 19

3.3.4 Summary . . . 20

3.4 Deep Learning Framework . . . 21

3.5 Deep Learning Data Preparation . . . 22

3.5.1 Raster to images . . . 22

3.5.2 Vector to labels . . . 23

3.5.3 Data Setup and Training . . . 24

4 Results 26

4.1 Model Performance and Accuracy . . . 26

4.1.1 Precision, Recall and F1 . . . 26

4.1.2 Generalized Intersection over Union . . . 28

4.1.3 Mean Average Precision (mAP) . . . 29

4.1.4 Validation Objectness & Validation Classification . . . 29

4.1.5 Summary . . . 30

4.2 Test area validation . . . 31

4.2.1 April 18, 2020 . . . 32

4.2.2 April 25, 2020 . . . 35

4.3 Summary . . . 38

5 Discussion 39

5.1 Training Issues . . . 39

5.1.1 Icebergs . . . 39

5.1.2 Ships . . . 41

5.2 Research Question 1 . . . 42

5.3 Research Question 2 . . . 43

6 Conclusion 45

Bibliography 46

Abbreviations

AIS . . . . Automatic Identification System
API . . . . Application Programming Interface
AWS . . . . Amazon Web Services
CFAR . . . . Constant False Alarm Rate
CNN . . . . Convolutional Neural Network
COWC . . . . Cars Overhead With Context
CPU . . . . Central Processing Unit
DEM . . . . Digital Elevation Model
DMI . . . . Danmarks Meteorologiske Institut
ESA . . . . European Space Agency
EW . . . . Extra Wide swath
FN . . . . False Negative
FP . . . . False Positive
GADM . . . . Database of Global Administrative Areas
GIoU . . . . Generalized Intersection over Union
GIS . . . . Geographic Information System
GPU . . . . Graphics Processing Unit
GRD . . . . Ground Range Detected
HH . . . . Horizontal-Horizontal
HV . . . . Horizontal-Vertical
IoU . . . . Intersection over Union
IW . . . . Interferometric Wide swath
mAP . . . . mean Average Precision
NOAA . . . . National Oceanic and Atmospheric Administration
RGB . . . . Red-Green-Blue
SAR . . . . Synthetic Aperture Radar
SLC . . . . Single Look Complex
SNAP . . . . Sentinel Application Platform
SRTM . . . . Shuttle Radar Topography Mission
SSD . . . . Single Shot Detector
TN . . . . True Negative
TP . . . . True Positive
VH . . . . Vertical-Horizontal
VV . . . . Vertical-Vertical
YOLO . . . . You Only Look Once

List of Figures

2.1 Visualisations of SAR properties foreshortening and shadows. Illustration from [Wang et al., 2019] . . . 5
2.2 The geographical distribution of Sentinel-1 modes and polarisations, map by ESA 5
2.3 Left: The basics components of a CFAR sliding window. Right: Visualization of how the sliding window is utilised in raster data. . . . 6
2.4 Different kind of deep learning applications. From left to right: image segmentation, object detection and image classification. Image from AWS. . . . 9
2.5 Illustration of how radio waves are reflected on a ship. Image from [Ouchi, 2016] 11

3.1 Outlined study areas and Sentinel-1 scenes. Basemap by ESRI . . . 13
3.2 Sentinel-1 scenes outlined on a AIS ship density plot. Ship density map from the Danish Maritime Authority (Søfartsstyrelsen) . . . 14
3.3 Sentinel-2 image of the Nuup Kangerlua study area, image captured on October 21, 2019. . . . 15
3.4 Sentinel-2 image of the Disko Bay validation area, image captured on April 21, 2020. . . . 16
3.5 Diagram visualising the Sentinel-1 pre-processing workflow . . . 17
3.6 Examples of iceberg and ship targets . . . 20
3.7 Individual polarizations to RGB composite . . . 23
3.8 Example content of a text label file. The bounding box values are pixels relative to the size of the image . . . 24

4.1 Precision, Recall and F1 curve. At the last training epoch: Precision = 0.48, Recall = 0.60, F1 = 0.53. . . . 27
4.2 Training GIoU loss and validation GIoU loss. . . . 28
4.3 mAP curve, measuring with IoU=0.5. At the last training epoch mAP = 0,557 = 55,7% accuracy . . . 29
4.4 Training and Validation loss for Objectness and classification . . . 30
4.5 April 18, 2020, Disko Bay Sentinel-1 RGB Composite . . . 32
4.6 April 18, 2020, Validation area detection output. . . . 33
4.7 April 18, 2020. Closeup examples detection, left images show iceberg polygons from DMI and right images show model detections. . . . 34
4.8 The only AIS point in the validation area, left image is the Sentinel-1 RGB composite and right is the model detection output . . . 35
4.9 April 25, 2020, Disko Bay Sentinel-1 RGB Composite . . . 36

4.10 April 25, 2020, Validation area detection output and validation data. 36
4.11 April 25, 2020. Closeup examples detection, left images show iceberg polygons from DMI and right images show model detections. 37
4.12 The 5 AIS data points in the validation area, top images are Sentinel-1 RGB composites and bottom images are the model detection output 38

5.1 Image from a camera station placed in the bottom of the fjord Nuup Kangerlua. Photo taken the 13th of August 2018 40
5.2 Example of Sentinel-1 iceberg training data 40
5.3 Example of ship detections in Denmark 42

List of Tables

3.1 Summary of the study areas, Sentinel-1 scenes and their project purpose 16
3.2 Iceberg detection parameters 19
3.3 Summary of detection process and number of targets 21
3.4 Satellite data information 21

4.1 Validation data Satellite information 31
4.2 April 18, 2020. Validation overview 35
4.3 April 25, 2020. Validation overview 37

5.1 May 17, 2020. Ship validation overview 43

1 Introduction

With still increasing interest and activity in the Arctic regions, ship traffic in Greenlandic and international waters is estimated to increase as well. The Norwegian DNV-GL and the Danish ministry of defense is estimating an overall growth in ship traffic of 24% by 2027, including a 100% increase in cruise ship traffic in Greenlandic waters [DNV GL, 2015][Danish Ministry of Defence, 2016]. For the Arctic waters as a whole, the estimated increase in ship traffic is anywhere between 100 - 500% until 2030 [Ocean Conservancy, 2017]. Overall, the estimates all vary, but there is a broad consensus that there will be an increase in ship traffic. The driving factor behind the increase in ship traffic is the combination of the decreasing sea ice and the economic growth in the region. A single mining project in Greenland can increase bulk ship traffic by an order of magnitude [DNV GL, 2015], and Russian and Norwegian oil and gas exploration is having a similar impact on traffic [Ocean Conservancy, 2017].
Due to increase in ship traffic, the task of monitoring the ocean is also becoming more difficult. This encompasses both preventing and responding to accidents, and also preventing illegal activities such as illegal fishing. A major threat to ships and offshore activities is floating icebergs. According to the National Research Council Canada, an average of 2.2 iceberg collisions is happening every year [Hill, 2010]. Ship collision with icebergs can have major environmental consequences, and a rescue mission can be complicated and dangerous, especially when ships carry passengers. Along with the prospect of increasing exploration of resources in the Arctic, forecasting the threat of possible colliding icebergs in still of rising importance

For monitoring Greenlandic waters, the Danish Defence is using a mix of different systems to identify and track ship positions. The main one is AIS (Automatic Identification System), a system required for fishing, passenger and ships weighing more than 300 tons. But since there are no AIS ground station receivers in Greenland, the AIS system is all satellite based, meaning that it must be supported by two other identification systems; Kyst-Kontrol (Cost-Control) and Greenpos [Danish Ministry of Defence, 2016]. The common ground for all these systems is that they rely on self-reporting from the ships, meaning that if a ship for some reason, is not capable, or does not wish to be identified, it can turn these systems off.
To get a full picture of ship traffic, areas are regularly monitored by aircrafts and ships are

asked to report unidentified vessels. Lastly, data from radar satellites, such as Sentinel-1, is used to locate unidentified objects [Danish Ministry of Defence, 2016].
Synthetic Aperture Radar (SAR) is very capable in ocean monitoring, mostly in regards to detecting oil spills and locating unidentified ships. Since SAR is based on an active sensor, it has capabilities to function both during day and night through any weather conditions. Spaceborn SAR products therefore allow for constant monitoring of very large areas.

Mapping and object detection through SAR data is based on the different surface textures of different object types. Depending on the application, different methods are applied in SAR mapping, sea-ice charting is mostly based on backscatter values measured from observations [Le Traon et al., 2019] where iceberg detection is relying on a adaptive threshold algorithm, called CFAR (Constant False Alarm Rate), functioning by calculating the difference in backscatter between the object and the ocean [Eastwood et al., 2019].
The same algorithm is used in ship traffic monitoring[Ouchi, 2016]. By using this method, large areas are analysed and objects not corrosponding to AIS or other reporting signals is of interest to further investigation [Danish Ministry of Defence, 2016].

SAR object detection within different fields, has a solid ground on well-established methodologies and is proven successful by many different parties. Basing all object detection on adaptive thresholding is not without issues though. Telling what an object actually is, is still up to human interpretation. When detecting icebergs, all objects detected are presumed to be icebergs, and likewise when detecting ships.
In areas where both types of objects are present, the consequence is that there is no solid way of telling the object types apart from each other. In a ESA study, published in 2016, detecting ships in 2000 Sentinel-1 SAR images covering Arctic waters using the CFAR, the conclusion states:
The presence of sea ice is a constant challenge. Accurate automatic tools to discriminate ice from open water are needed to increase the reliability of the SARbased ship detection, both reducing the number of false alarms and increasing the number of ships detected. This includes the need for automatic ship-iceberg discrimination capability.
[Santamaria et al., 2016]

Based on the established detection algorithm CFAR, this study aims at carrying out a further classification of these detected objects. By deploying a deep learning workflow and algorithm trained on detected objects, the goal is to perform an iceberg-ship discrimination of unidentified objects detected in Sentinel-1 SAR data.
Training of the model will be based on data consisting of objects detected in waters without icebergs, and objects detected where there is a great presence of icebergs, in that way interference of object types is avoided.

1.1 Problem statement and research questions

Using SAR data for ocean and ice monitoring has been done for years, and the applied methodologies have been proven strong in many cases. In the areas of the world where there is an interference of similar objects, such as ships and icebergs, a strong classification methodology is still needed for telling these objects apart. With the still increasing usability and documentation of deep learning object detection algorithms and workflows, an object detector trained on known SAR data objects should be able tell these objects apart. Earlier studies have shown promising results [Power et al., 2018][Heiselberg and Heiselberg, 2017], and with the continuous development within object detection capabilities these can still be improved.

The following research questions arise:

1. How does the different polarizations used in the Sentinel-1 data acquisition affect the model results?
2. With the use of deep learning, what degree of accuracy can be achieved in detecting and classifying floating ocean objects?

2 Background

2.1 Sentinel-1

Sentinel-1 is a satellite constellation under the ESA Copernicus program. Sentinel-1 consists of two Synthetic Aperture Radar (SAR) satellites A and B, in polar-orbit operating on the C-band frequency. The frequency of the C-band radio waves makes data acquisition possible independently of atmospheric conditions, as the radio wave is penetrating clouds. The satellite has two main operating modes, Extra Wide Swath (EW) and Interferometric Wide Swath (IW) mode, the Stripmap (SM) and Wave (WV) modes are also available, but to a much lesser extent. Each mode has multiple polarisations, both single and dual, describing how the satellite antenna transmit and receive the signal. The polarisations are available as follows:

- HH - Horizontal transmission, Horizontal receiving - Single polarisation
- HV - Horizontal transmission, Vertical receiving - Single polarisation
- VV - Vertical transmission, Vertical receiving - Single Polarisation
- VH - Vertical transmission, Horizontal receiving - Single Polarisation
- HH+HV - Dual polarisation
- VV+VH - Dual polarization

The modes and polarisations are available in different product types, Level 0 - Raw data, Level 1 SLC, Level 1 GRD, and Level 2 OCN.
Each of these levels describes the data processing done by ESA prior to making data available to the user. Different SAR applications takes advantages of the different products types. The SLC product is strong in monitoring vertical ground movements, as the contained phase data detects very small changes. Where the GRD data contains only backscatter value, representing the strength of the returned signal and thereby the surface properties of the object.

Because SAR is an active sensor, the returned 'image' is distorted in relation to the incidence angle of the wave. This means that objects at lower angles will appear more flat or might disappear behind taller objects. These phenomena are described as foreshortening and shadowing, and is illustrated in figure 2.1.

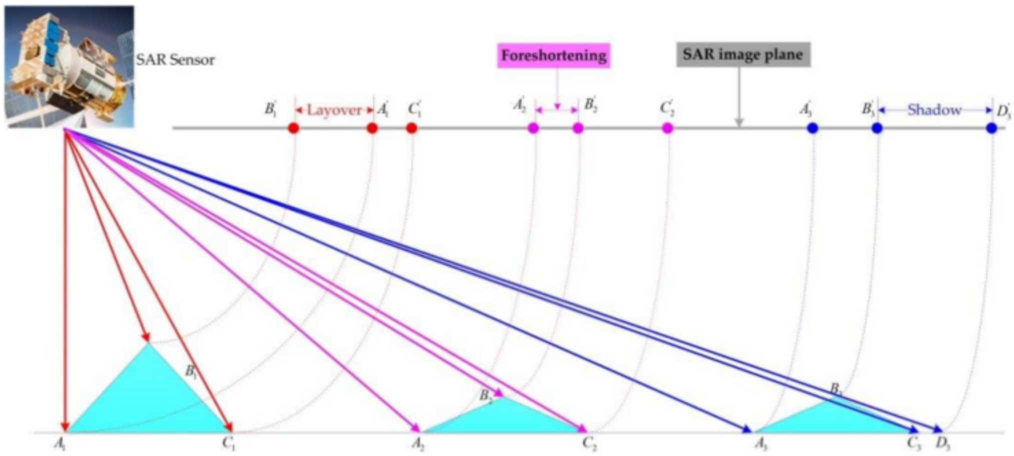

Figure 2.1: Visualisations of SAR properties foreshortening and shadows. Illustration from [Wang et al., 2019]

The data acquisition in the different data modes and polarisations is not distributed evenly throughout the globe. Below figure 2.2 shows the distribution of data in the respective modes and polarisations.

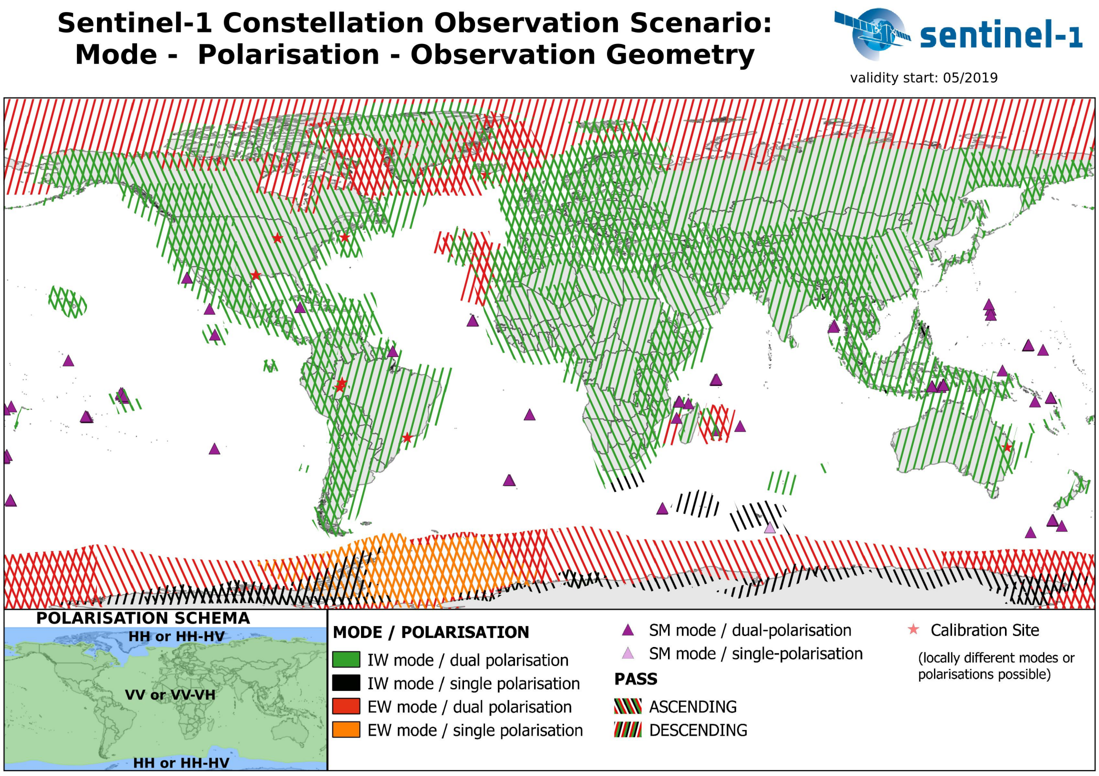

Figure 2.2: The geographical distribution of Sentinel-1 modes and polarisations, map by ESA

With two satellites orbiting the Earth in 12 days, the official revisit time for Sentinel-1 is six days. Because the satellites are in polar orbit, the revisit time increases at more

northern latitudes (and southern) due to overlap between orbit lines.
This makes Sentinel-1 especially capable of marine traffic monitoring, iceberg mapping and ice charting in European waters and into Arctic areas, as these areas have coverage from every third day down to every day.

2.2 Current Detection Methods

In order to lay out the current issues within vessel detection, an understanding of the underlying algorithm and methods within the field is necessary. As stated in the introduction (section 1), the algorithm mostly used for vessel detection is the Constant False Alarm Rate algorithm (CFAR) in either traditional form or modified versions. Major institutes, such as ESA (EU), NOAA (US) and Kongsberg (NO) all use this algorithm [Ouchi, 2016].

CFAR is an algorithm made for SAR data with the purpose of detecting objects against a homogeneous background [Eastwood et al., 2019], mostly open waters and oceans. The algorithm is not made specifically for vessel detection, but has proven strong since objects detected in open waters for the most part are very likely to be marine traffic. At more northern latitudes the same algorithm is applied in iceberg detection, and hereby the arriving of the issue stated in chapter 1.

The CFAR algorithm is an adaptive threshold algorithm, utilizing the different backscatter values reflected by different types of objects. The algorithm can be found in many different forms, that all rely on the same key components; a sliding window and local calculated threshold.

The sliding window, seen in figure 2.3, consist of 3 parts/windows; test cell, guard cells and reference cells. These vary in size, based on input data and desired detection outcome.

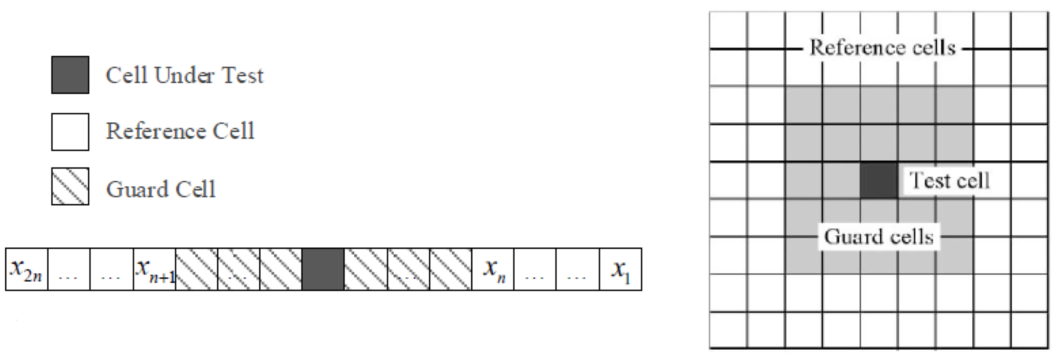

Figure 2.3: Left: The basics components of a CFAR sliding window. Right: Visualization of how the sliding window is utilised in raster data.

The reference cells are representing the background in the sliding window, and these sets are the basis for the locally calculated threshold. There are multiple ways of calculating the threshold, in the simplest algorithm type (Cell-Averaging CFAR) the threshold is calculated by a simple average of the cells in the window [Zhao et al., 2019]. The use of guard cells ensures that there is no interference between the test cell and the reference cells.
If the value of the test cell is above the calculated threshold, the cell is marked as being present of an object.

This is the simplest way of calculating the probability of false alarm i.e. if an object is present or not, but this simple form of CFAR does not handle sea clutter and background noise very well. Many further developed variations of CFAR share the same goal of providing a solution to account for different and complex real life situations that cause noise in the detection [Ouchi, 2016].

Another issue with the CFAR detector is its ability to detect multiple targets in close proximity to each other. Due to the fact that the threshold is being calculated in a pre-defined area around the cell under test, there are only limited ways of ensuring that other objects do not interfere with the calculation. As the threshold is supposed to be calculated for and represent the background, if other objects of interest are present in the background, the threshold will be calculated too high and all objects will remain undetected.
A typical way of accounting for this issue is to iterate over multiple CFAR detections, with different parameters and combining the results for each iteration. Such methods are mostly seen in iceberg mapping, such as DMI and Copernicus products [Eastwood et al., 2019], as in these situations targets proximity to each other is difficult to predict [Ressel et al., 2015].

The CFAR is a robust algorithm, that thanks to a wide range of adaptations is documented to perform well in object detection [Zhao et al., 2019].
However, CFAR is only an object detector not at all an object classifier, still leaving the definition of the object up to human interpretation. Naturally, the shift towards more complicated deep learning algorithms is already happening, and within the last years, the performance and detection capabilities of deep learning algorithms are being tested, documented and implemented.

2.3 Deep Learning

Within the last decade, there has been a significant evolution in the technology and results of deep learning models. The 2015 ImageNet competition on image recognition and classification [Russakovsky et al., 2015] is considered a benchmark in deep learning, proving the capabilities of Convolutional Neural Network (CNN). A major driver in the evolution of deep learning models is the equal rise in available training data. In order to train and evaluate any deep learning model, an accurate labelled data set is required. Both in terms of labelled objects in the data and the simple amount of images, training data has grown in volume consistently throughout the decade.

Initially the PASCAL VOC [Everingham et al.] was the most comprehensive data set available, consisting of close to 10.000 images labelled in 20 different classes. The data set was since then expanded into the new ImageNet, consisting of more than a million images labeled in 1000 classes.
In terms of data set size for general image recognition, ImageNet is still leading, followed by Open Images [Kuznetsova et al., 2020] and MS COCO [Lin et al., 2014].

The mentioned data sets are all examples of cases of deep learning applications related to image processing. While these different applications are not completely separable from each other, they can largely be divided in three different categories, segmentation, object detection and classification. See figure 2.4 for examples.
Image segmentation classifies each pixel in the image to a class. This will give the exact extent of each object located in the image.
Object Detection will locate an object and mark it with a bounding box, which might include a confidence score.
Classification will take an image as input, usually an image chip being less than 100x100 pixels. The output will be a classification of what is in the image, but not any marking of the actual object.

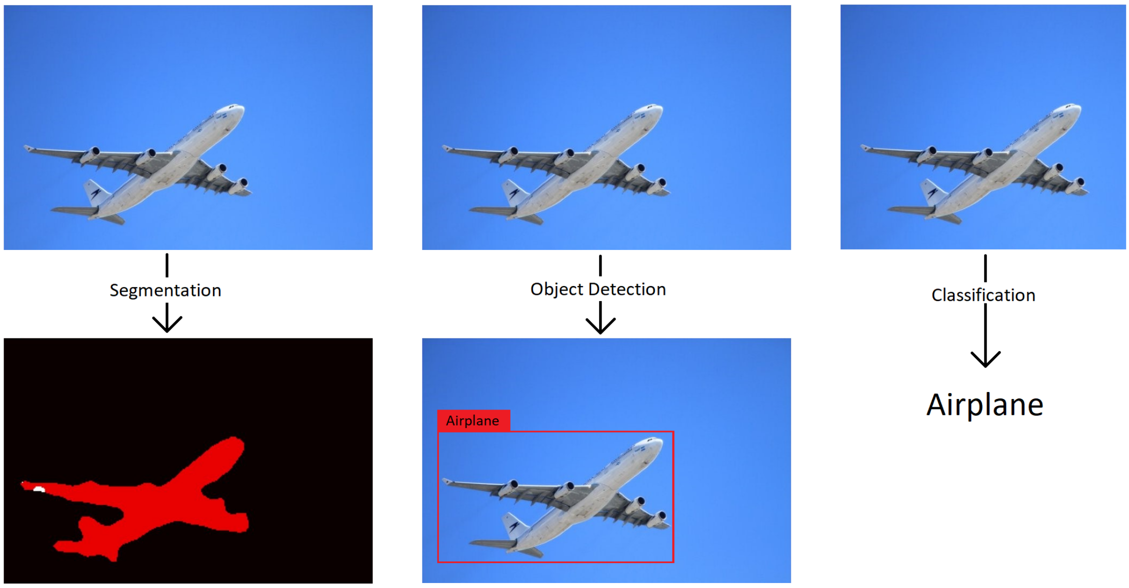

Figure 2.4: Different kind of deep learning applications. From left to right: image segmentation, object detection and image classification. Image from AWS.

The different applications each have their advantages and disadvantages, usually it depends on the desired output which method is most suiting. An important difference separating the classification from the two others, is the ability to detect multiple objects in the same image, the classification will only give one output per image. The two former methods will output every detected object in an image.
The underlying architectures are not defined by the three examples, it is to a large degree the same algorithms at work. The important aspect of the different applications is the difference in input and output data, and the fact that different training data sets are available in different formats.

In the wake of the deep learning competitions throughout the years, private organisations and researchers have achieved better and better results by continuous improvement and further development of deep learning architectures. The 2014 winner of the ImageNet competition, saw drastic accuracy improvements by increasing the depth of the existing of convolutional network architecture [Simonyan and Zisserman, 2014]. The added network layers, comes at a cost of computational power. The shift from CPU processing to GPU processing has opened up to much available processing power, making the transition from convolutional network into deep neural networks possible [Raihan et al., 2019].

Today, the CNN architecture has developed into multiple different frameworks, utilizing very deep neural networks, making image processing possible in real-time or close to. Some of these frameworks are:

- Faster R-CNN - Region-based Convolutional Neural Networks) [Ren et al., 2015]
- SSD - Single shot multibox detector [Liu et al., 2016]
- YOLO - You Only Look Once [Redmon and Farhadi, 2018]
- Resnet - Residual Neural Network [Szegedy et al., 2017]

The different frameworks are all being tested and benchmarked against the large data sets, such as ImageNet or MS COCO. These data sets consist of every day images, usually less than 1000x1000 pixels which is ideal for development and training.
Moving into remote sensing, the content and dimensions of image data are of a whole other scale. While this does cause some hindrance, image-processing frameworks can still be applied in remote sensing.

2.4 Deep learning in remote sensing

Applying deep learning image processing frameworks in remote sensing products, such as aerial and satellite imagery is with great prospect but has some challenges as well.
A main issue is objects relation to each other and the background in the image. In remote sensing products, the background to an object is very complex compared to normal images as objects are much more evenly distributed and all having roughly the same distance to the sensor [Van Etten, 2019]. As an example, a car in a normal image would be present in a decent amount of the photo and have its wheels towards the ground, have two visible head or rear lamps and so on. In medium resolution satellite imagery, a car would just be a colored box consisting of relative small amount of pixels and a random rotation.

The size of remote sensing products is also causing issue, as the processing frameworks are designed to work on images only a few 100 pixels in size, where satellite imagery can easily be 10-100 times bigger. Image classification of remote sensing image chips has also seen better results than object detection (difference shown in figure 2.4), mainly due to the background complexity being more simple [Long et al., 2017].
When applying deep learning in remote sensing, there is no way of getting around that the algorithms cannot handle images in that size. Downsampling is not an option, since most image information would be lost to reach a sub 1000 pixel level. Instead the go-to technique is to only process a small portion of the image at a time, this can either be achieved through a sliding window technique such as the one proposed by [Van Etten, 2019] or by cropping images and stitching the output back together as done in [Zhang et al., 2018].

With more prepared data sets being available for training such as SpaceNet [Van Etten et al., 2018] and COWC [Mundhenk et al., 2016], cars, buildings, and roads has become benchmark objects for deep learning task in remote sensing. Building and road extraction is usually done as image segmentation [Zhang et al., 2018][Vakalopoulou et al., 2015] where the exact geographical extent is mapped in opposition to locating more single objects like cars and airplanes which is usually done as object detection [Long et al., 2017][Van Etten, 2019].

All the work findings of the papers cited above, is done on aerial and satellite imagery RGB composite images. Shifting into SAR data brings other image capabilities but the frameworks can still be applied here. Especially within vessel detection and classification, deep learning has showed promising results.
Open waters make a less complex background, making vessel detection and even classification of ship types possible [Dechesne et al., 2019], though shoreline areas and moored ships still pose a challenge as the background become more complex [Wang et al., 2019].

Because of a ships appearance in waters, it will appear bright in SAR data as the reflections are often reflected onto multiple surfaces, as illustrated in figure 2.5. The properties of SAR described in section 2.1 and visualized in figure 2.1 means that ships will be subject to foreshortening and shadows. This does not pose an immediate issue, but states the importance of spatially distributed training object since ships will appear differently depending on their relation to the sensor.

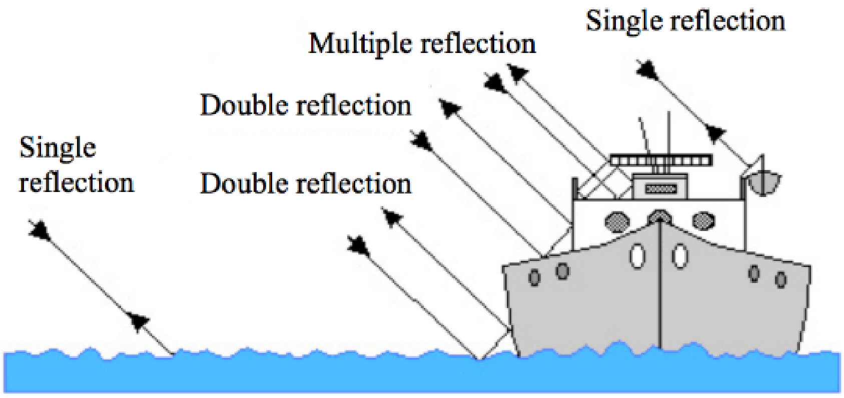

Figure 2.5: Illustration of how radio waves are reflected on a ship. Image from [Ouchi, 2016]

Recent studies have documented the performance of different deep learning frameworks and proven promising accuracy results in SAR ship detection.
In the creation of a training data set for SAR ship detection, [Wang et al., 2019] utilised and evaluated different deep learning frameworks such as SSD, Faster R-CNN and RetinaNet. Between the frameworks, a significant difference in training time was observed not to have a great impact on overall accuracy. RetinaNet achieved the highest accuracy but at the cost of the longest training time. All frameworks achieved accuracies between 88 and 91%.

[Dechesne et al., 2019] managed to classify types of ships in Sentinel-1 images, by training OpenSAR data on a multi-task neural network. The study achieved accuracies of 96-97% on image chips and 85% on Sentinel-1 scene patches.

While [Wang et al., 2019] did not test the Yolo object-detector it has proven to be fast and accurate [Redmon and Farhadi, 2018][Benjdira et al., 2019]. [Chang et al., 2019] compared training and detection between Faster-R-CNN and YoloV2. The study managed to achieve 90% accuracy with the Yolov2 detector, 20% higher than Faster-R-CNN, and proved significant better training and detection times, all this is however based on a data resolution between 1 and 5 meter.
Lastly, the study showed slightly better results when using larger images as input, though again at the cost of computation time. With the requirement of handling data either in slices or in a sliding window, using larger image chips might prove useful when preparing large data sets.

Methodology 3

As outlined in section 1.1 the goal of the study is to train a deep learning model to detect and classify objects in Sentinel-1 SAR data, with the objects being either ships or icebergs. The proposed methodological framework of the project will consist of multiple parts, starting at processing of Sentinel-1 SAR data and performing object detection with CFAR which is then evaluated and converted into a deep learning training format. Lastly covered, is the implementation and setup of the chosen deep learning framework.

3.1 Study Areas

The study areas consist of three different locations, two in Western Greenland and one in Denmark, each study area is outlined by two selected Sentinel-1 scenes.

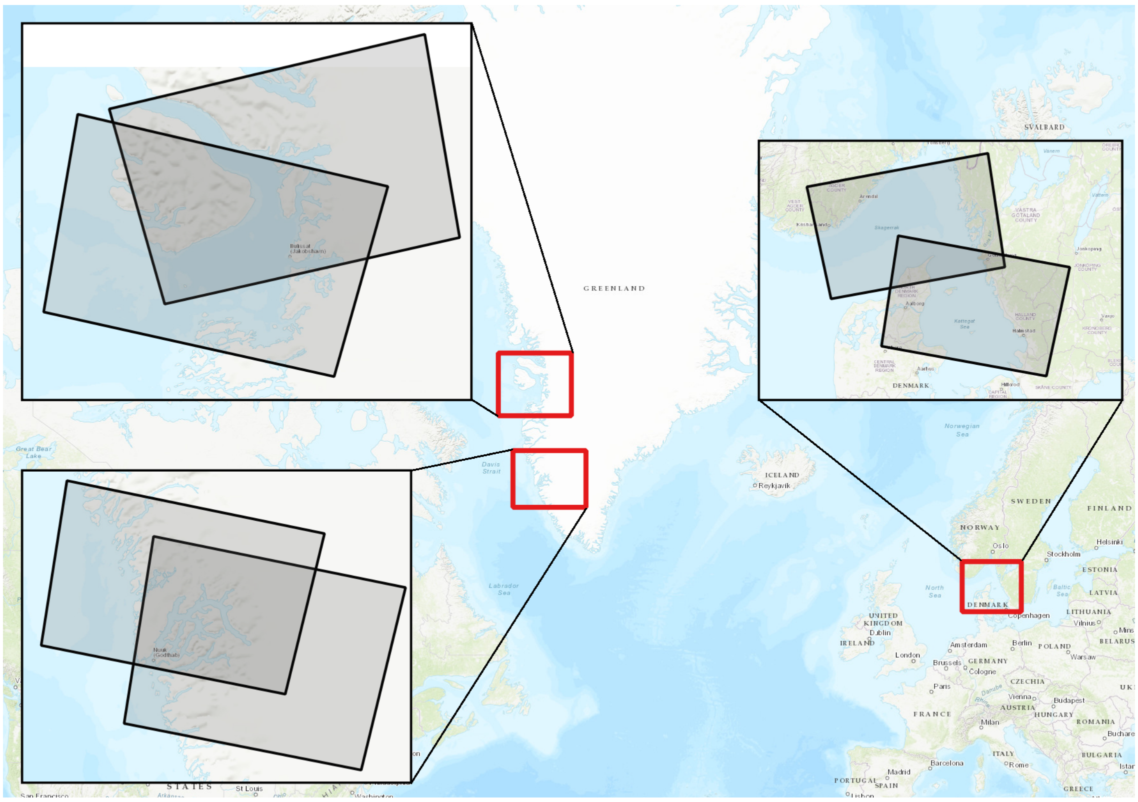

Figure 3.1: Outlined study areas and Sentinel-1 scenes. Basemap by ESRI

Each study serves an individual project purpose, two for collecting training data and one for testing the accuracy of the final model.

3.1.1 Denmark

The Danish study area covers the waters of Kattegat and Skagerrak, as these waters are the only access to the Baltic Region they both handle busy shipping routes. The main traffic comes from the Baltic Sea and goes into the Northern Sea, the selected Sentinel-1 scenes both covers the shipping lines North-South and East-West directions. Together with ships going to destinations in Denmark and traffic between Denmark, Sweden and Norway, a large variety in types of ships and in different directions, is covered within the study area.

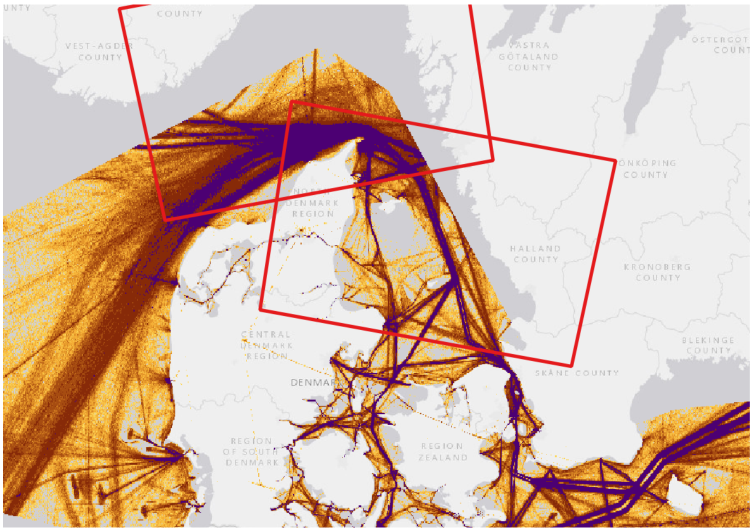

Figure 3.2: Sentinel-1 scenes outlined on a AIS ship density plot. Ship density map from the Danish Maritime Authority (Søfartsstyrelsen)

The density plot in figure 3.2 also cover the area where the Danish Maritime Authority provide historical AIS data free of charge and free to use.
This AIS will be used to locate ship targets within the study areas to be passed on as ship training data for the model.

3.1.2 Greenland, Nuup Kangerlua

One of the two Greenland study areas, the southern one, is centered around Nuuk and the surrounding fjord Nuup Kangerlua (Godthaabsfjorden). This is a large fjord complex, stretching from the capital Nuuk all the way to the Greenland Icecap. Because there are two glaciers flowing into the fjord, vast amounts of ice and icebergs are present most of the year. The amount of ice in the fjord does change through the seasons, with most ice and icebergs in summer and least in winter, as higher temperatures cause more ice to break off the glaciers.

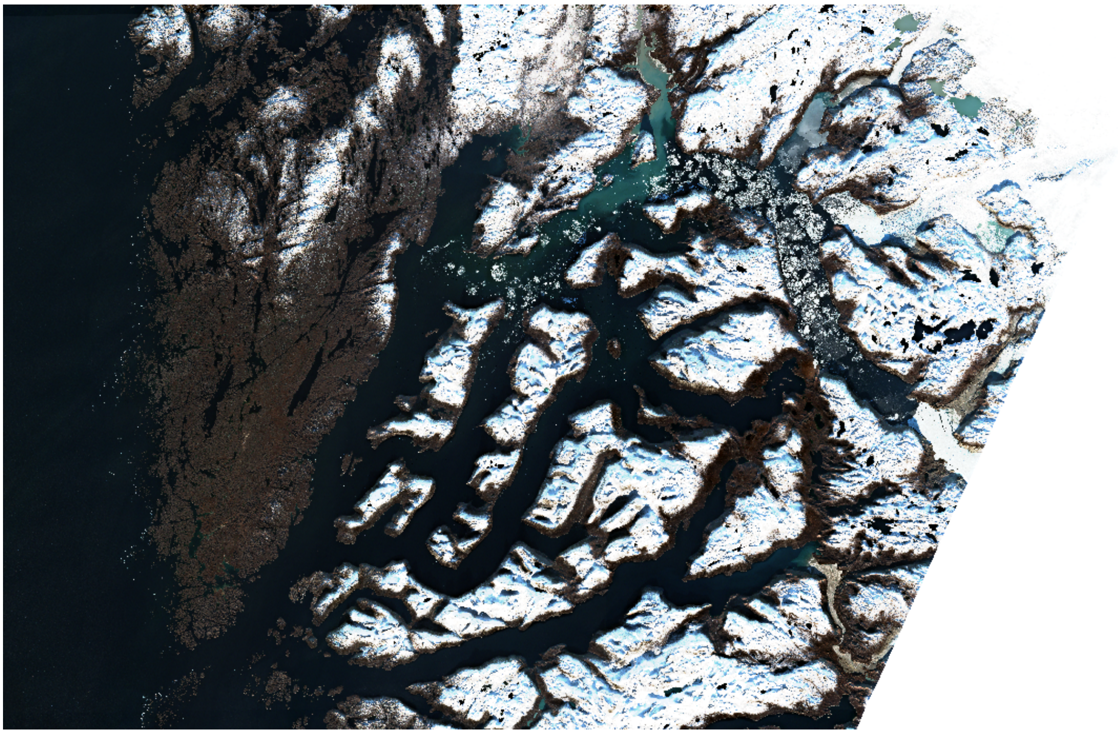

Figure 3.3: Sentinel-2 image of the Nuup Kangerlua study area, image captured on October 21, 2019.

Besides around the harbour of Nuuk, there is only limited commercial ship traffic in the fjord. There is one weekly passenger boat to a smaller settlement at the bottom of the fjord and some commercial tourist traffic is also seen over the summer, but for the most part the traffic consist of small private vessels.
The large amount of ice and limited ship traffic makes the fjord ideal for detecting icebergs. Sentinel-1 objects detected with CFAR in the fjord, will be used as iceberg training data. With the use of AIS data, the objects are known not to be ships.

The area is complety covered by two Sentinel scenes, see figure 3.1 for reference.

3.1.3 Greenland, Disko Bay

The final area (northern most in figure 3.1) will not be used to create training data, but instead serve as validation area for the trained model.
The Disko Bay is a large area in western Greenland, with the third largest town illulisat located here together with a number of smaller towns and settlements. Sermeq kujalleq (Jacobshavn Gletscher) is one of the most active glaciers in the world and feeds ice directly

in to the bay. The Disko Bay is the most visited area by tourists in Greenland and sees a large amount of commercial fishing as well.

With the types of ships in the bay being both fishing, passenger, and goods delivery, and with the constant flow of icebergs, Disko Bay is the ideal test area for a deep learning model identifying icebergs and ships.

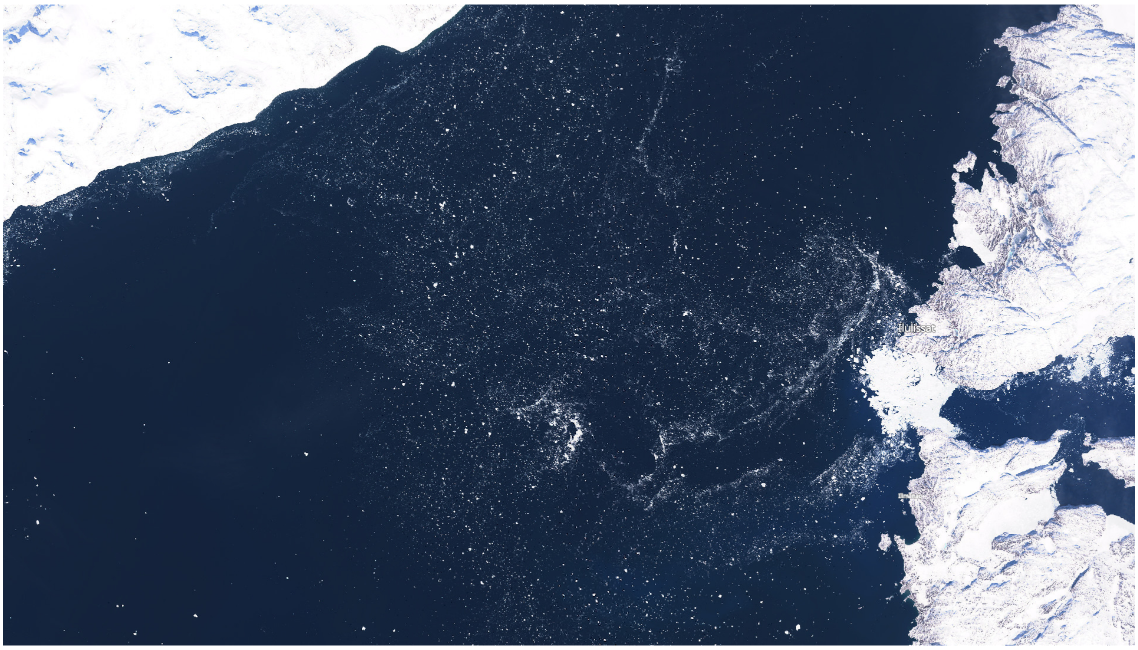

Figure 3.4: Sentinel-2 image of the Disko Bay validation area, image captured on April 21, 2020.

Figure 3.4 shows the central Disko Bay validation area. This area is covered by the two selected Sentinel-1 scenes (see figure 3.1) so within this area the main part of the validation will happen. The glacier outflow is seen in the right side of the image, letting out ice and icebergs into the Disko Bay and toward Disko Island (top left in image).

3.1.4 Summary

Table 3.1: Summary of the study areas, Sentinel-1 scenes and their project purpose

Location	**Sentinel-1 IDs**	**Purpose**	**Object**
Kattegat, Skagerrak - DK	Orbit: 44, 66 Slice: 19, 2	Training data	Ship
Nuup Kangerlua - GL	Orbit: 54, 127 Slice: 1, 5	Training data	Iceberg
Disko Bay - GL	Orbit: 98, 90 Slice: 2, 9	Validate results	Ship & Iceberg

3.2 Data Acquisition and Pre-processing

The acquisition for Sentinel-1 data must be in coherence with available AIS data. For Denmark, this not an issue, as historic AIS data is freely available, but for Greenland, there is no available data source for AIS data. Some online traffic services exist, but these only allow the watch data in real-time but not any actual access to the data.
From Gatehouse, who is the main supplier of AIS data in Denmark and Greenland, a full AIS data set for Greenland has been provided for the dates of **November 24, 2019** and **November 30, 2019**, and for the validation areas for the dates of **April 12 - April 25, 2020**.

The Sentinel-1 data is acquired for the given dates at each study area, the product type used in the study is the GRD product in IW mode in both types of dual polarization (HH+HV, VH+VV). See available polarizations for each study area in figure 2.2. The data in the GRD products are projected onto a WGS84 ellipsoid, converting data from radar geometry to evenly spaced pixels [Bourbigot et al., 2016]. In the IW mode, the satellite capture data in a resolution of 22 x 20 meters and this is then projected into 10 x 10m pixels.

Prior to use, the Sentinel-1 data is pre-processed in roughly accordance to the workflow suggested by [Filipponi, 2019]. The carried out workflow is visualized in figure 3.5

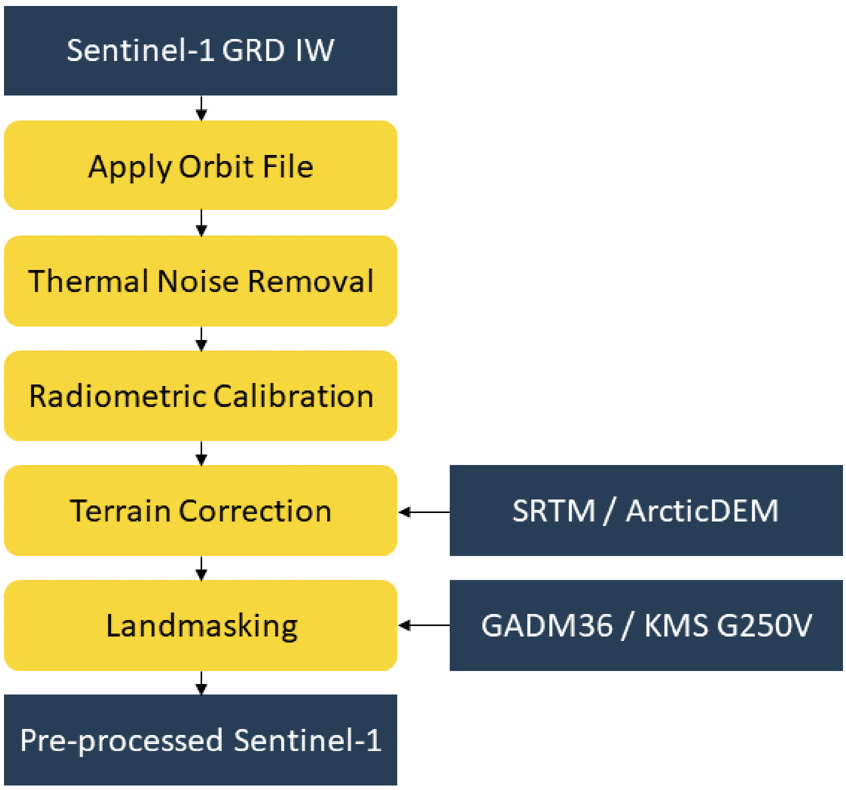

Figure 3.5: Diagram visualising the Sentinel-1 pre-processing workflow

The processing workflow is carried out in the ESA SNAP Python API (plication Programming Interface) - "Snappy". This Python API contains most functions from the SNAP software and can be scripted in a Python environment.

For performing the terrain correction, two different DEM products are necessary. In Denmark, the SNAP software provides a direct access to the SRTM DEM [Rodriguez et al., 2005] through a temporary auto download. SRTM is however, only a available uptill 60° North, so the terrain correction at the Greenland study areas are carried out with the use of ArcticDEM [Porter et al., 2018] which is available from 60° North. ArcticDEM is a product available for free download, but requires some processing before usage, a premade DEM mosaic of the study areas is used for the terrain correction.

The landmasking is based on coastlines in vector format. The coastline for Denmark and its surrounding countries is from the GADM version 3.6 [Hijmans et al., 2018] level 0 data set. In Greenland, the coastline is obtained from historic KMS (Kort og Matrikelstyrelsen) data, today distributed through kortforsyningen[1].
To account for inaccuracies in the coastline data, especially for Greenland, each coastline is expanded by a 250 meter buffer.

3.3 Object localisation

For the creation of training data within the two study areas of Nuup Kangerlua and Denmark, objects must be localised and outlined. At the Nuup Kangerlua where icebergs are the objects of interest a CFAR detection is carried out, fitted to detect icebergs.
At the Danish study area, the training objects are ships and these are localised and outlined manually with the use of AIS data.

3.3.1 CFAR Iceberg Detection

The detection is carried out, using the snappy python API. For each product polarization, two individual detection processes are performed, a total of four detections for each scene. The goal of running two detection is to locate targets at different sizes, the first detection aims at locating small targets and the second aims at locating bigger targets.
Based on recommendation from DMI, the guard window and background window, is a constant value relative to the test cell. The guard window is $12 \times testcell$, and the background window is $37 \times testcell$, see figure 2.3 for reference.

[1] SDFE - Kortforsyningen. Grønland 1:250.000

Table 3.2: Iceberg detection parameters

Detection	Test Cell	Guard Window	Background Window
Small targets	20 meter	240 meter	740 meter
Large targets	40 meter	480 meter	1480 meter

3.3.2 Iceberg Validation

As icebergs are very dynamic natural objects, constantly moving and changing shapes and sizes they are challenging to validate. Currently, the only way of getting actual ground truth data is by manual observation, for example from a camera station or by doing overflights.
Using other satellite products, such as the optical Sentinel-2 satellite is not useful either. The time difference between Sentinel-1 and Sentinel-2 is circa 9 hours, meaning that the icebergs cannot be found in the same location in the two products.

With the use of the Greenland AIS data from Gatehouse, it has been assured prior to detection that no ships are present in the area. The ships in the port of Nuuk are masked out by the 250 m coastline buffer.

To minimize the risk of false targets, a number of validation steps is carried out. Most importantly, only objects detected in both polarizations are kept, this eliminates noise in the data being detected, as the level and distribution of noise is different in the two polarizations. This is a key measure in reducing false targets, and is also used out by DMI in their iceberg charting [Eastwood et al., 2019].

The Greenlandic coastline is very complex and as stated earlier, the coastline vector used for land-masking is of poor quality, so surface rocks, small islands and fjords are not perfectly masked out. Targets detected outside the main fjord and within groups of very small islands are deleted as well.
Due to the poor coastline vector, even with the 250 m buffer land areas are not completely masked out. Objects that are intersecting or are very close to the coastline vector are removed as these are likely to be land areas.

Lastly, objects detected at glacier outflows are removed manually. Glacier outflows are not covered in the coastline vector and their outreach is also season dependent. By examining the Sentinel-1 data manually, the objects at these locations have been removed.

3.3.3 Ship localisation

A CFAR ship detection would have been possible to carry out. But given the relative low number of targets within each scene in contrast to the scene size and together with the fact that AIS data is available, it is more efficient to create the training data manually. The validation of CFAR detection would have required manual effort as well, as the AIS is not completely accurate.

The historic AIS data obtained from Søfartsstyrelsens FTP-Server[2]. From the FTP-server, AIS data is downloaded as a CSV file containing every received AIS signal from that given day. To reduce the file size and workload, the file is first sorted according to the DK study areas (see figure 3.1), and afterwards sorted to match the timestamp of the Sentinel-1 data. An AIS signal is transmitted from the vessel every 10 seconds and is roughly received likewise, so the timestamp can be closely matched to the timestamp of the satellite acquisition.
AIS is mandatory on vessels above 300 tons, passenger ships, and fishing vessels above 15 meters, smaller ships can chose to fit an AIS system as well, but are not imposed to do so.

Each AIS point within the scene is assessed manually, and any visible target nearby is roughly outlined. The accuracy of the target outline does not have to be perfect, as these are converted to bounding boxes later on.

3.3.4 Summary

The detection of the two object types, outputs a total of 2279 targets, these targets can be passed on the as training data for the object detector. Examples of the two objects types are seen below in figure 3.6.

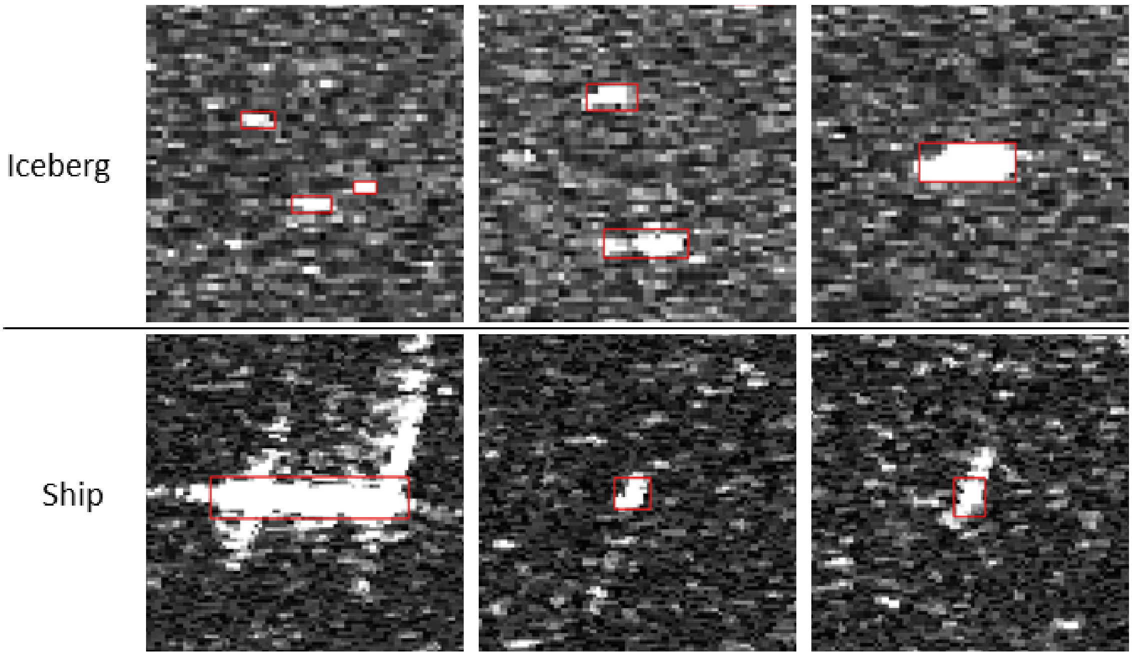

Figure 3.6: Examples of iceberg and ship targets

[2] Søfartsstyrelsen - Historiske AIS Data, FTP-Server

Table 3.3: Summary of detection process and number of targets

Object	Actions	nr. of targets	S1 acq. date
Icebergs	Locate objects with multiple CFAR detections Only keeping features detected in both HH and HV Removing features outside of fjord AOI Removing features in contact with or close to land Removing features at glacier outflows	1760	20191124 20191130
Ships	Ships outlined manually	519	20191008 20191119 20191123 20200116 20200223

Table 3.4: Satellite data information

Target Object	Satellite	Acquisition Time	Path	Incidence Angle *
Iceberg	Sentinel-1A	24/11/2019 09:45:23 - 09:45:52	Descending	38.8352°
Iceberg	Sentinel-1B	30/11/2019 09:44:41 - 09:45:10	Descending	38.7391°
Ship	Sentinel-1A	08/10/2019 05:32:31 - 05:32:56	Descending	38.6916°
Ship	Sentinel-1B	19/11/2019 05:31:39 - 05:32:04	Descending	38.5835°
Ship	Sentinel-1A	23/11/2019 17:02:05 - 17:02:30	Ascending	38.6377°
Ship	Sentinel-1B	16/01/2020 17:01:20 - 17:01:45	Ascending	38.5773°
Ship	Sentinel-1B	23/02/2020 05:31:35 - 05:32:00	Descending	38.5870°

* Incidence angle is measured at mid swath, angle towards swath edges are about +/- 8° - 9°

3.4 Deep Learning Framework

As stated in chapter 2.3, there is a large variety of deep learning frameworks to choose from. The YoloV3 object detector [Redmon and Farhadi, 2018] has shown good performance and accuracy in earlier studies, and in remote sensing cases aswell [Van Etten, 2019]. The choice of deep learning framework for this study will therefore be the YoloV3 object detector. The Yolo name, You Only Look Once, states the detectors capabilities in assessing a whole image at a time, without the need of dividing it into sub regions. The YoloV3 is the third version of the original Yolo detector.

YoloV3 and its predescessor YoloV2, is built on the Darknet network which is a multi-layer convolutional neural network. The development of Yolo is based on the improvements made to Darknet, the latest version of Darknet is Darknet-53, a 53 layer deep convolutional neural network with shortcut connections.

The implementation of YoloV3 is done with the proposed setup developed by Ultralytics[3] [Jocher et al., 2019]. This setup uses Pytorch, a python library for machine- and deep learning applications, and NVIDIA Cuda which allows processing to utilize the GPU (given that it is a NVIDIA GPU). The setup in compatible with Windows, where other proposed Yolo setups only run on Linux. The setup is executed in a Python Conda environment, this can be found on GitHub[4].

Lastly, Utralytics made a number of pretrained weights that can be used in the deep learning model. Training a model from scratch is a very time consuming and computationally heavy task, so training upon pretrained weights is a great advantage, since the model already knows basic statistics and pattern recognition. Even though the aim of the detector is to find objects currently unknown the model, these elements still improve the detection.
The amount of new classes and derived values from these, are specified in a model configuration file, allowing the model to properly train on top of the existing weight file.

3.5 Deep Learning Data Preparation

3.5.1 Raster to images

The Ultralytics deep learning setup is made for object detection in standard every day photos and videos, this means that the Sentinel-1 data has to be converted into a format readable by the script. To mimic an classic RGB photo, the polarizations must be placed into separate bands and the datatype of the bands must be converted as well.

Visualization of Sentinel-1 data has a big impact on what is visible in the data and in many cases the visulazation can be adapted to fit the purpose of the application. To avoid human bias in the visualization, each band is visualized with a 2% - 98% culumative cut, this will for the most part make the objects visible and ignore data outliers.
The Sentinel-1 float32 bit depth is too high for an image object detector to process, so each band is converted into 16bit unsigned intergers, with values stretching from 0 - 65535. This is the highest bit depth possible in PNG format, to which the Tiff files are converted from.

Lastly, each of the two polarizations (HH + HV or VH + VV), are placed into separate bands. The HH or VH polarization are placed in the last band, as studies suggest that ships in general are more visible in this polarization [Pelich et al., 2019][Wang et al., 2019]. The RGB Sentinel-1 composite will therefore be a PNG file consisting of:

[3] YOLOv3 in PyTorch - Ultralytics Github
[4] GitHub - frhass/yolo_env

- R = HH or VH
- G = HV or VV
- B = HH or VH

In figure 3.7, the individual bands and the RGB composite derived from those, are visualized.

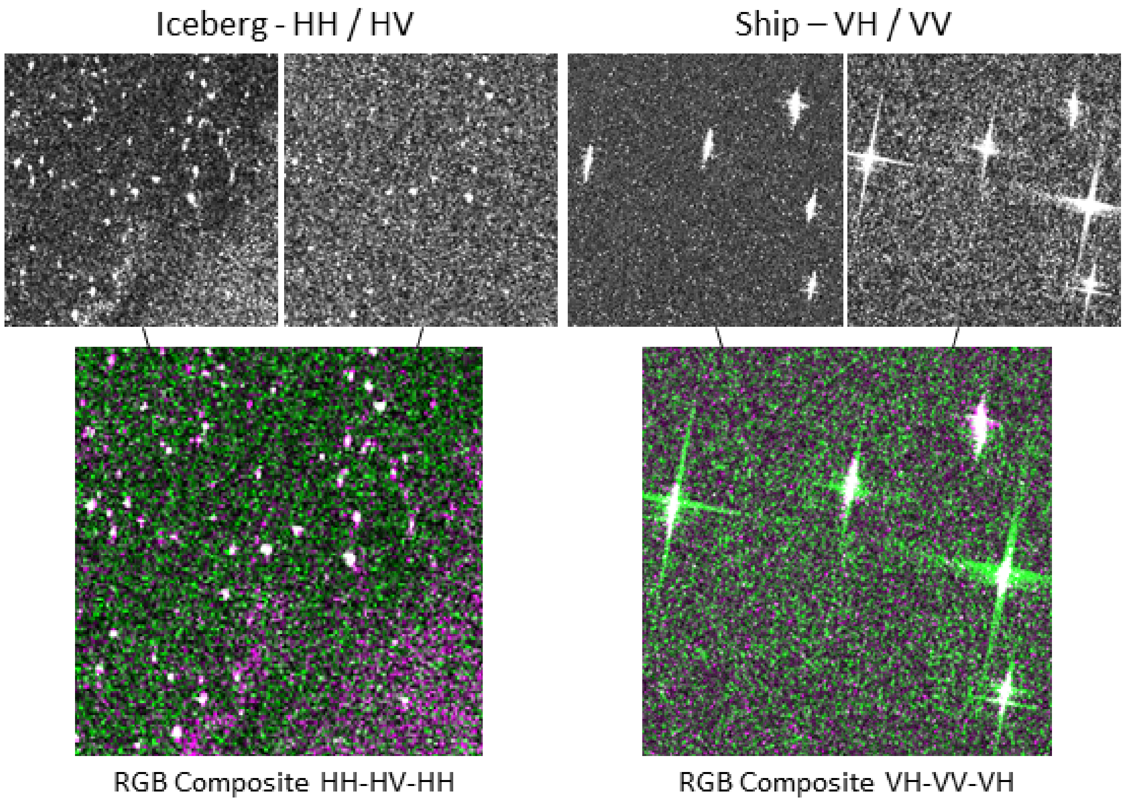

Figure 3.7: Individual polarizations to RGB composite

3.5.2 Vector to labels

There are multiple methods of labeling training data and each method is dependent on the specific setup of the framework used, YoloV3 uses the Darknet format for labeling training data. The Darknet format labels consist of a set of images with a corresponding text file which have the labels for the given image. The labels are defined by a class and a bounding box in normalized coordinates relative to the image, meaning that all values are between 0-1 and represent exact pixel locations in the corresponding image. See figure 3.8 for reference.

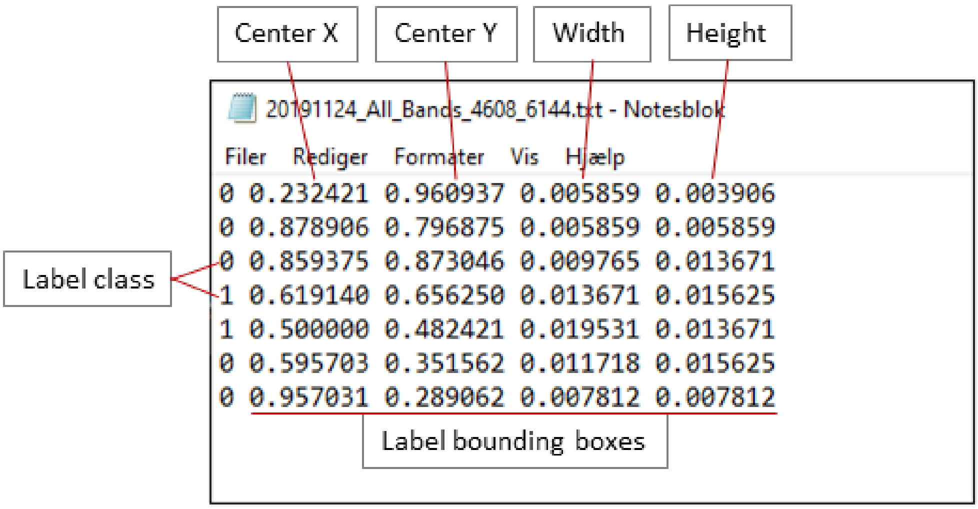

Figure 3.8: Example content of a text label file. The bounding box values are pixels relative to the size of the image

The Yolo training script search for labels corresponding to available images, thus the name of the label file must have en exact match to an image file. The label classes are defined in a *name* file, for the case of this project; 0 = iceberg and 1 = ship.
As described in section 3.3, the initial training data is made up of shapefiles outlining the objects of interest. Shifting from shapefiles with geographical coordinates into object with pure image coordinates can be challenging, as most GIS software does not handle this type of data. Some tools in ArcGIS PRO are available to convert shapefiles into image coordinates, but these did not quite fit the Darknet format, so instead the job was done using a dedicated Python script. The full script can be found on Github[5].

3.5.3 Data Setup and Training

Yolo takes various input image sizes, as long as they are divisible by 32 or 64, which is the base size of the CNN filter outputs. Image sizes can range between 200 and 600 pixels, were larger images tend to achieve better result but at the cost of computational power [Wu et al., 2015]. Images greater in size can be processed as well, but going above ~600 pixels only has the documented effect of increased processing time [Redmon and Farhadi, 2018][Ammar et al., 2019].

The python script takes one Sentinel-1 RGB composite scene and a vector shapefile with objects, and slice the image into smaller 640×640 pixel images and create a corresponding text file with the image coordinates of the vector objects. These are placed into separate *image* and *label* folders.

[5] GitHub - Frhass/yolo_dataprep

The data is then split randomly into training and validation data, a split of 80/20 is used for each object class. Each object class is having its own split, since the large difference in number of objects would cause to few ships in the validation data set.

The model is trained on a NVIDIA Quadro M4000 GPU, for a total of 350 epochs. An epoch is equal to the entire training data set being passed on to the model, going through the data set 350 times resulted in a total training time of 26.87 hours.

Results 4

The following chapter is divided into two sections, model performance and test area validation. Model performance will be outlined based on the models own accuracy assessments, which is calculated from the validation data given to the model. Secondly, the predictions made by the model at the Disko Bay project site are validated against ground truth data. The ground truth data consist of iceberg polygons from DMI, detected with the use of a CFAR algorithm, and ship positions obtained from satellite AIS.
At the test area, ground truth data is used to carry out the primary validation. This will tell how well a real world scenario can be predicted independently of how well the model appears to be trained.

4.1 Model Performance and Accuracy

To measure model performance and accuracy, the model was given a validation data set with 385 objects contained in 114 images. This data is 20% of the images in the whole data set, randomly selected with an equal amount of icebergs and ships. In a deep learning context, this amount of validation data is a bare minimum in measuring any kind of performance, the statistics provided by such a small validation data set is only usable to some extent.
While measuring actual model performance might not be sensible, measuring validation accuracy assessments can give good indication of how the training went and help to avoid model overfitting. Each model accuracy metric has its own curve to follow and sudden major deviations from this can be a sign of training issues or model overfitting.

4.1.1 Precision, Recall and F1

To measure the models detection capabilities, the *Precision, Recall* and *F1* scores are used [Manning et al., 2008]. The formulas include the classification terms:

- *TP*, True Positive - Model detects a true object
- *FP*, False Positive - Model detects a false object
- *FN*, False Negative - Model did not detect a true object

The scores are calculated as:

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$
$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The precision and recall metrics are both measuring the model detection capabilities, but are each accounting for different factors in the detection.
Precision is a measure of how accurate the model is at making positive predictions i.e. objects detected by the model. Since only detected positives are used in the formula, the *precision* is likely to remain high as long as very few objects are detected. The *recall* accounts for this by measuring false negatives i.e. objects not detected, and is thereby a measure of how much is detected out of what should have been detected. These two measures are often moving opposite to each other, high precision causes low recall and vice versa. The *F1* score accounts for these measure biases, and is thereby perceived as a more overall accuracy [Manning et al., 2008]. A high F1 score means a small amount of both false negatives and false positives. All measures are valued between 0 - 1, with 1 being a perfect validation result.

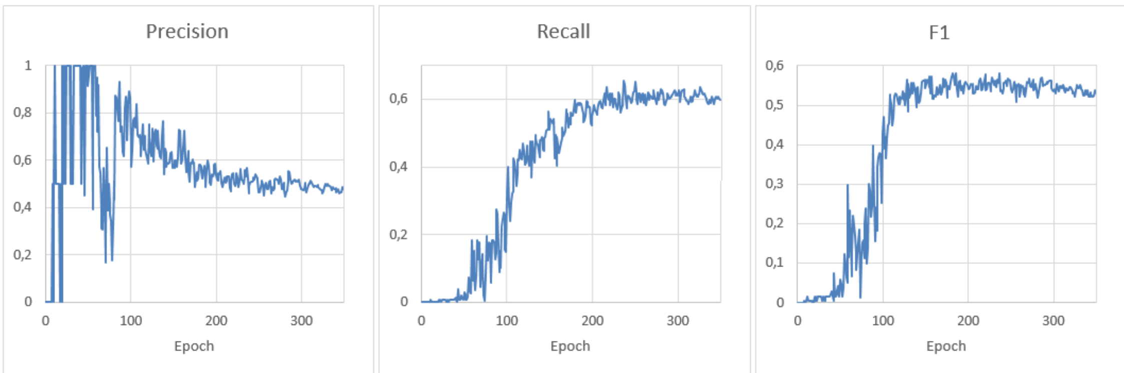

Figure 4.1: Precision, Recall and F1 curve. At the last training epoch: Precision = 0.48, Recall = 0.60, F1 = 0.53.

Assessing the graphs, the precision graph comes out as difficult to evaluate. The cause for the odd looking graph is probably found in the reasons mentioned above, most likely a very low amount of detections. The Recall and F1 graph are in this training case of more relevance, these follows a smooth curve indicating a satisfactory training.

4.1.2 Generalized Intersection over Union

The $GIoU$ (Generalized Intersection over Union) metric is a measure for how well the model predicts object bounding boxes. The GIoU is developed from the standard IoU (Intersection over Union), a metric measuring how well a predicted bounding box intersects the ground truth bounding box. This metric only returns a value if there is a intersection, whereas the generalized version takes the proximity of the two boxes into account. This is especially useful for small objects, as the bounding boxes of small objects are easily 'missed' and thereby returning a value of 0. In these cases the GIoU would still return a value, indicating if the boxes were close to each other [Rezatofighi et al., 2019]. The GIoU is calculated as:

$$GIoU = \frac{|A \cap B|}{|A \cup B|} - \frac{|C \backslash (A \cup B|}{|C|} = IoU - \frac{|C \backslash (A \cup B)|}{|C|}$$

A and B represent the predicted and ground truth bounding boxes, and C is the bounding box containing both of these. $\cap$, and $\cup$ are representing area of overlap and union respectively.

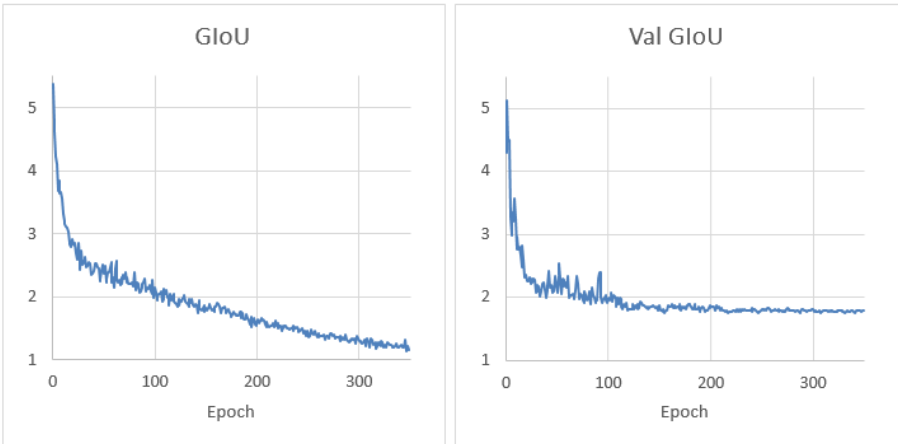

Figure 4.2: Training GIoU loss and validation GIoU loss.

The graphs in figure 4.2 show the loss of the metrics, the loss is indicating how wrong the model is in localising objects. In general, the loss of these two metrics should be as low as possible and the training should follow a steady decreasing curve that flattens at the end. The $GIoU$ is calculated from the training data and the *val* $GIoU$ is calculated from the validation data. Both graphs are indicating that overall training went well, though the $GIoU$ seems to be still in a decrease at the end of the training, indicating that the model could have trained for longer.

4.1.3 Mean Average Precision (mAP)

mAP (mean Average Precision) is a measure for overall model accuracy, derived by calculating the area under a *Precision-Recall* curve. The *Precision* and *Recall* metrics are good assessments for model accuracy but are each sensitive to false negatives and false positives, meaning that these graphs alone can sometimes be misleading. By plotting a curve of these two metrics and calculating the area under the curve, a non bias metric for the overall model accuracy is found [Manning et al., 2008]. The mAP-0.5 is stating that only objects with a IoU threshold above 0.5 (having 50% overlap) were used in this metric.

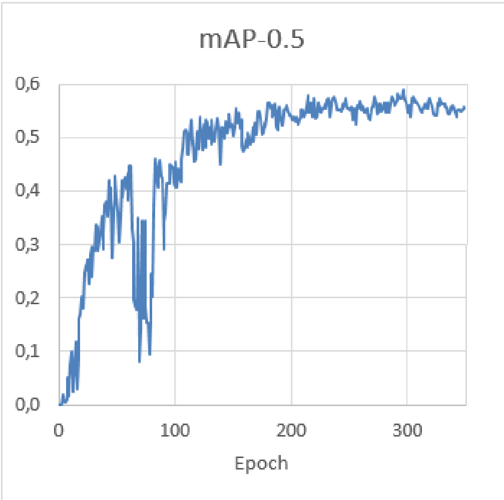

Figure 4.3: mAP curve, measuring with IoU=0.5. At the last training epoch mAP = 0,557 = 55,7% accuracy

Ideally, the mAP curve should have a steady increase through most of the training and then flatten out at the end. The curve follows the expected pattern most of the time, except for some major deviations around epoch 75. Given how the curve is calculated and by looking at the Precision curve in figure 4.1, the cause for this is most likely found in the factors causing spikes in the Precision curve.

4.1.4 Validation Objectness & Validation Classification

The last two relevant metrics are *Validation Objectness* and *Validation Classification*. These metrics measure the models ability to decide if a bounding box contains an object, and if so, what the object is. The model will internally create a large number of bounding boxes, and the objectness score is measuring how well the model picks which boxes contain an object, this is utilizing a prediction score and the IoU metric. If the box is predicted to contain an object, the *Validation Classification* is then measuring how well the model can

determine the object class [Alexe et al., 2012].
The most importance use of these two metrics is their ability to show overfitting. The output values should be relatively stable during training, meaning that if they start to rise the model is overfitting.

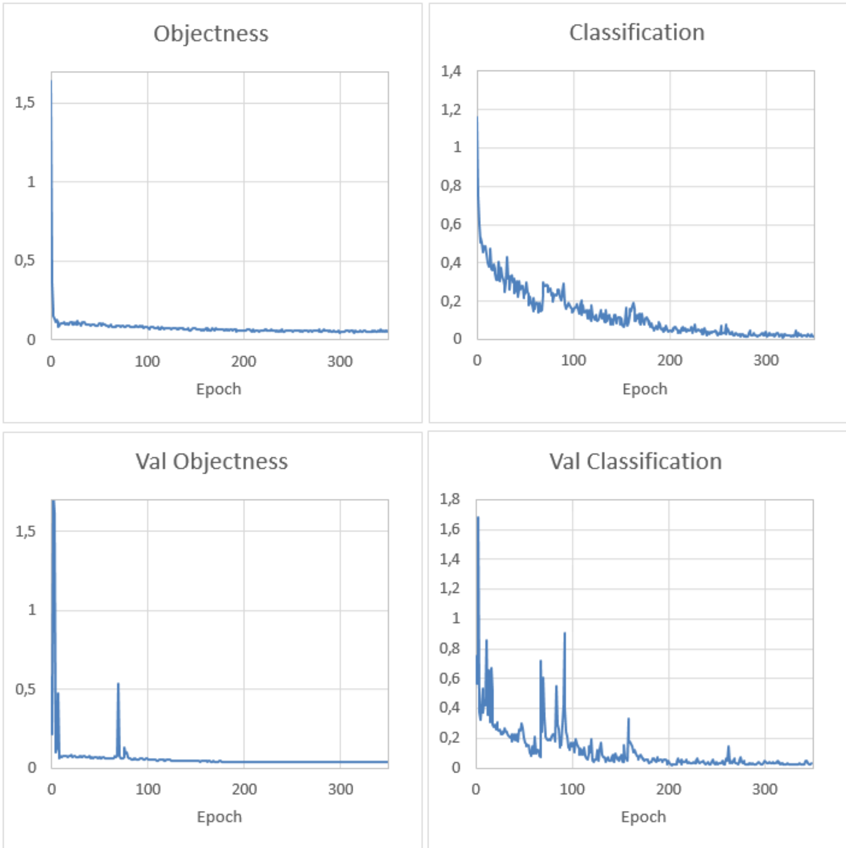

Figure 4.4: Training and Validation loss for Objectness and classification

The graphs in figure 4.4 are again a measure of loss of the two values, meaning that they should decrease throughout the training. The very flat *Objectsness* graph indicates that the model is not learning a lot throughout the training, and the validation is again spiking around the same time seen in the mAP and Precision curve. The *Classification* however indicates a more steady training, with the model having continuous improvements. With the objectsness measuring localising objects and classification determining what the object is, the graphs are indicating that the challenge mostly lies in classifying objects in the assigned categories. The graphs shows no indication of overfitting.

4.1.5 Summary

The training graphs gives an overall indication that the training went well, but also shows room for improvement. The biggest issue with the training is the reoccurring sudden spikes

around epoch 75. As these spikes only happen in validation graphs and the precision, which is also utilising validation data, the cause is most likely an issue in the validation data.
Secondly, the fact that the metrics measuring accuracy all stall around 55% is not a direct issue, but given the usual performance of deep learning applications, it is not great either. This could have been increased by lowering the IoU threshold, that would however, not improve the model but only make it more generous in validation. In order to achieve a higher accuracy, the only real solution is to use more data in training. The small amount of data also means that the metrics should not singly be used as overall model validation as it is simply not enough data to do proper statistics on. Instead, a test against new ground truth data is the best way to get actual validation measures of the model.

4.2 Test area validation

The test area validation, or the ground truth test, is a full scale detection test carried out at the Disko Bay validation area (see figure 3.1) in two individual Sentinel-1 scenes. For each scene, a full detection is carried out with a confidence level of 0.5, this means that the model will only return objects that have a 50% certainty of being either an iceberg or a ship.

The two dates selected for validation are **April 18, 2020** and **April 25, 2020**, in below table the satellite information for two dates are presented.

Table 4.1: Validation data Satellite information

Satellite	**Acquisition Time**	**Path**	**Incidence Angle**
Sentinel-1A	18/04/2020 20:47:31 - 20:47:56	Ascending	38.5720°
Sentinel-1B	25/04/2020 10:07:57 - 10:08:22	Descending	38.4363°

Validation icebergs are detections made by DMI for the Copernicus *Sea Ice Berg Concentration* product[1], this data is represented as polygons outlining each detected iceberg. The polygon data set is not publicly available, but have been provided for this project, a low resolution overview of the data is also published at DMIs PolarPortal[2].

The ship AIS data for the Greenland study areas has been provided by the Danish company Gatehouse, this data is not publicly available. Because the data is satellite AIS, it has a lower accuracy than shore-based AIS systems, such as the system in Denmark. The

[1] Copernicus Marine Service. Arctic Ocean - SAR Sea Ice Berg Concentration
[2] DMI Polar Portal, Isberge.

AIS data is sorted according to the timestamp of the Sentinel-1 acquisition, this does not guarantee an exact match but only points close to detected objects.

4.2.1 April 18, 2020

In the Sentinel-1 scene of April 18, 2020, a total number of 18317 icebergs and 226 ships were detected.

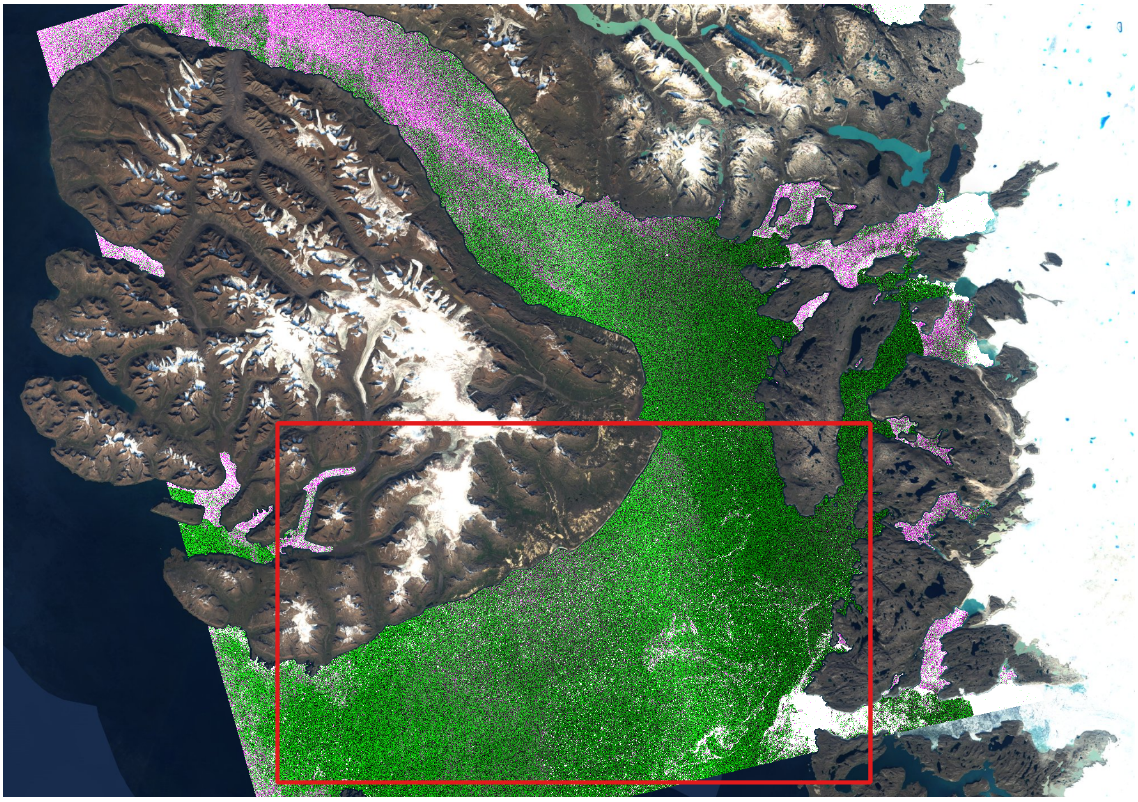

Figure 4.5: April 18, 2020, Disko Bay Sentinel-1 RGB Composite

In figure 4.5, the full scene RGB Composite can be observed. The composite is made up of HH-HV-HH, which means that open water is represented by the green color and strong reflective objects (icebergs and ships) appear as white with some purple reflection as well. The vast amount of purple seen in top of the image is a mix of icebergs and floating sea ice.

The model has not been trained in dense ice situations so no validation will take place at such areas. The mostly green area in the lower half of the image, marked by the red square, is open water with a large number of objects. This area is used for the primary validation.

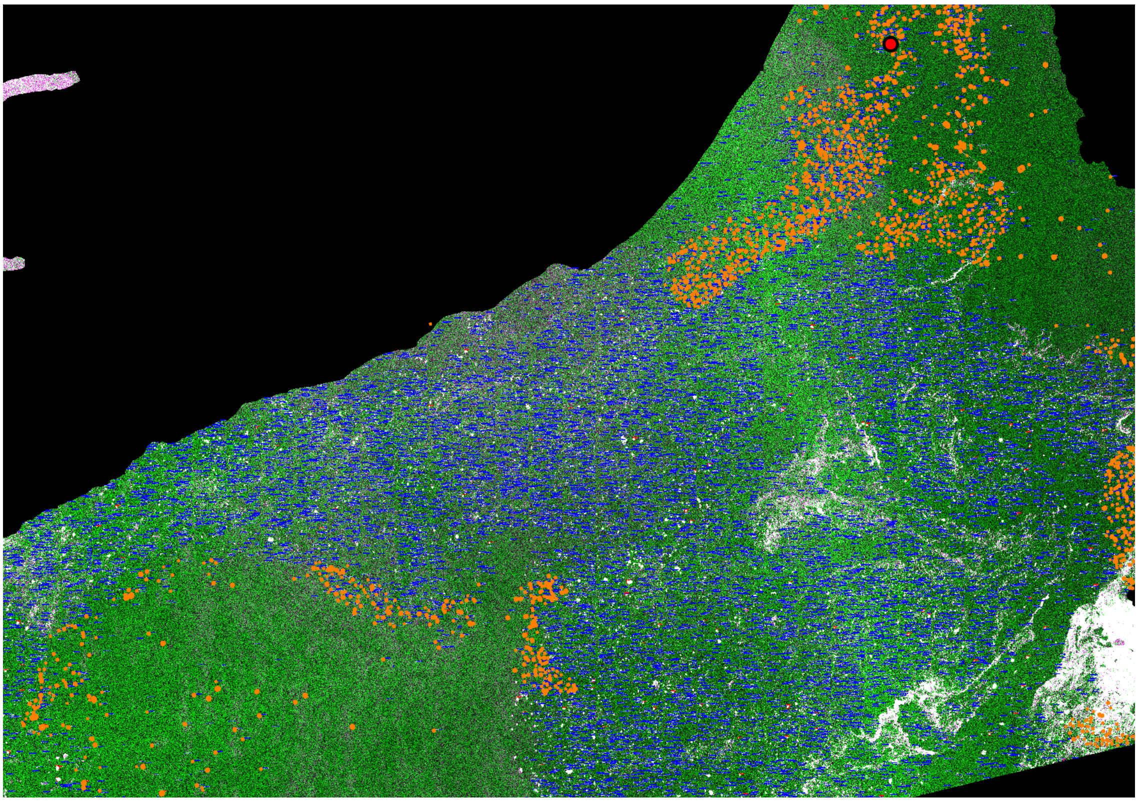

Figure 4.6: April 18, 2020, Validation area detection output.

In figure 4.6, the detection output for the validation area is seen, detected iceberg are represented with a blue color and detected ships are represented in red. Due to a relatively small amount of ship detections they are difficult to see in the image. The orange polygons are icebergs detected by DMI in the same Sentinel-1 scene, and the red point at the top of the image is the only ship present at the data acquisition time .

Figure 4.6 clearly shows that icebergs detected by the project model, and icebergs detected by DMI do not follow the same geographic extent. The reason for this, is the fact that DMI only detects icebergs in open waters, and assessing the DMI sea ice-chart of the date[3] shows that the large area without DMI detections is in fact classified as sea-ice.

[3] DMI: Ice chart central West Greenland, April 18, 2020

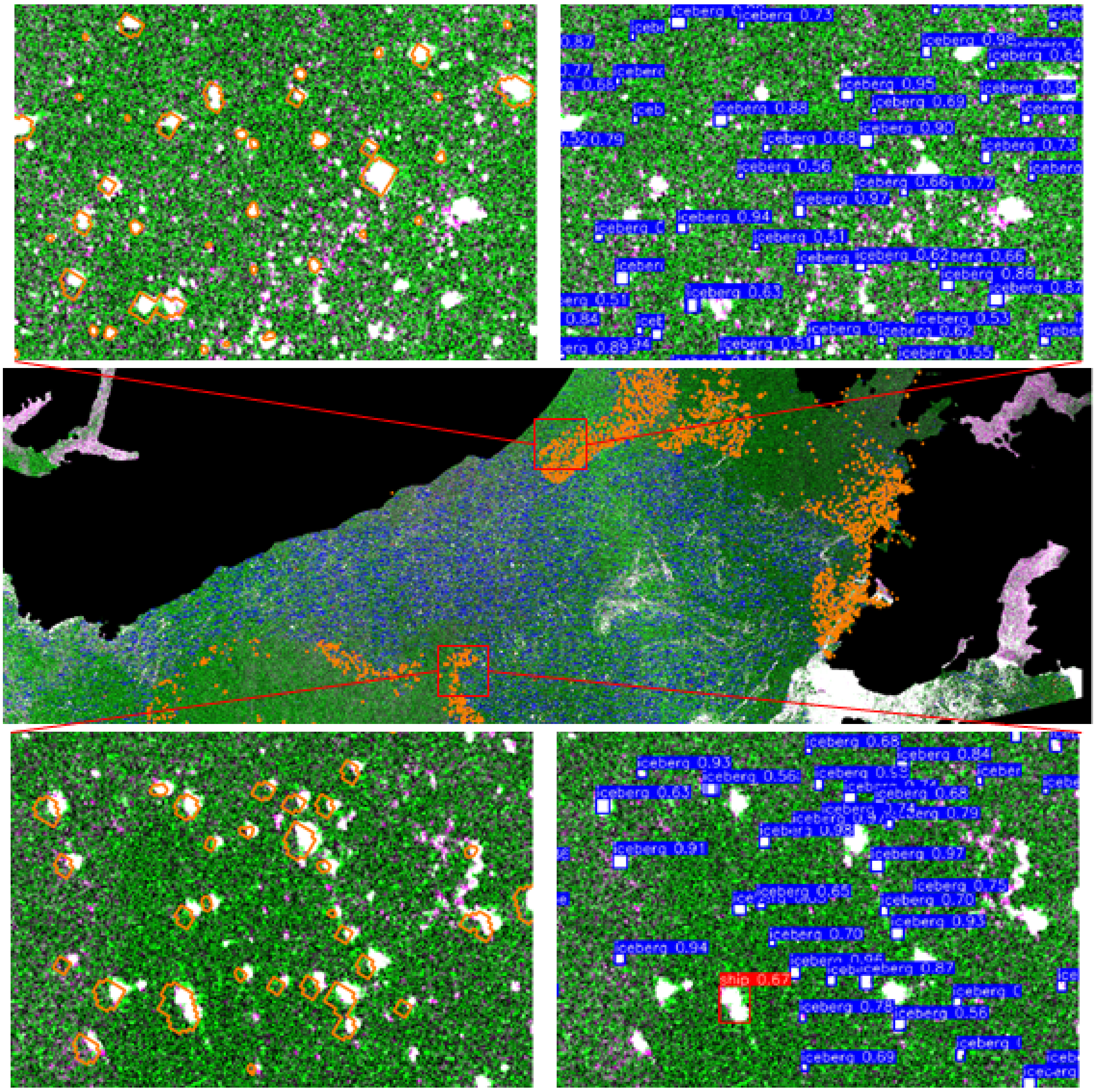

Figure 4.7: April 18, 2020. Closeup examples detection, left images show iceberg polygons from DMI and right images show model detections.

In the validation area, DMI detected a total of 2340 icebergs, these are the orange polygons in figure 4.7. The polygons appear to have a small offset towards left, this is due to differences in Sentinel-1 pre-processing. With the use of DMI iceberg polygons, the detections made by the project are validated against these. The polygons are used as ground truth and the model is validatet by measuring how many of the these where detected. In the areas without iceberg ground truth, validation is not possible.

Each iceberg polygon is validated with three possible outcomes: Detected as iceberg, Detected as ship, or Not detected. A validation overview is seen in table 4.2 below.

The validation shows that out of the 2340 icebergs detected by DMI, 54.95% of them where detected by the model. A fraction of these where correctly detected but classified wrongly

Table 4.2: April 18, 2020. Validation overview

DMI Icebergs	**Detected as iceberg**	**Detected as ship**	**Not detected**
2340	1243	42	1055

as ships. This overall accuracy of 54.95% corresponds very well to the models predicted accuracy of 55.7% (see mAP, figure 4.3).

In the validation area, there was only 1 known ship present and this has not been detected.

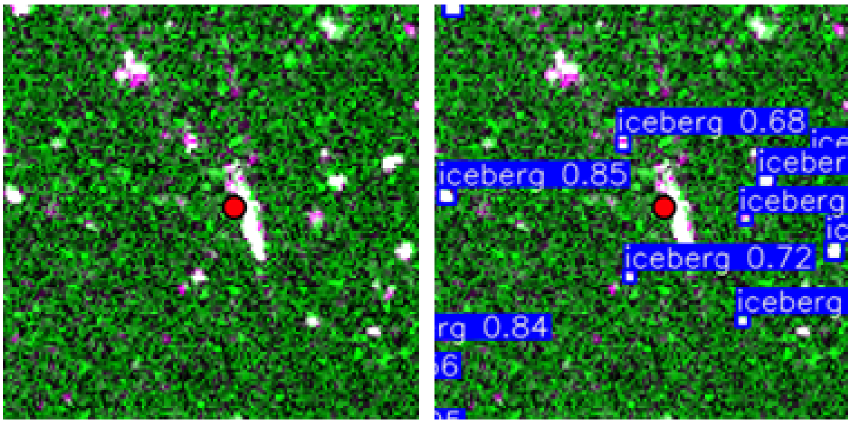

Figure 4.8: The only AIS point in the validation area, left image is the Sentinel-1 RGB composite and right is the model detection output

With only one validation point, no sensible accuracy assessment can be made. With the result from the iceberg validation along with the fact that 226 ships where detected even though only one were present, this validation does not indicate that the model is capable of detecting and correctly classify the ship class.

4.2.2 April 25, 2020

In the Sentinel-1 scene of April 25, 2020, a total number of 23576 icebergs and 207 ships were detected. In figure 4.9 below, the full scene is seen in RGB composite with a smaller validation area. The validation of this scene followed the same procedure as the scene of April 18.

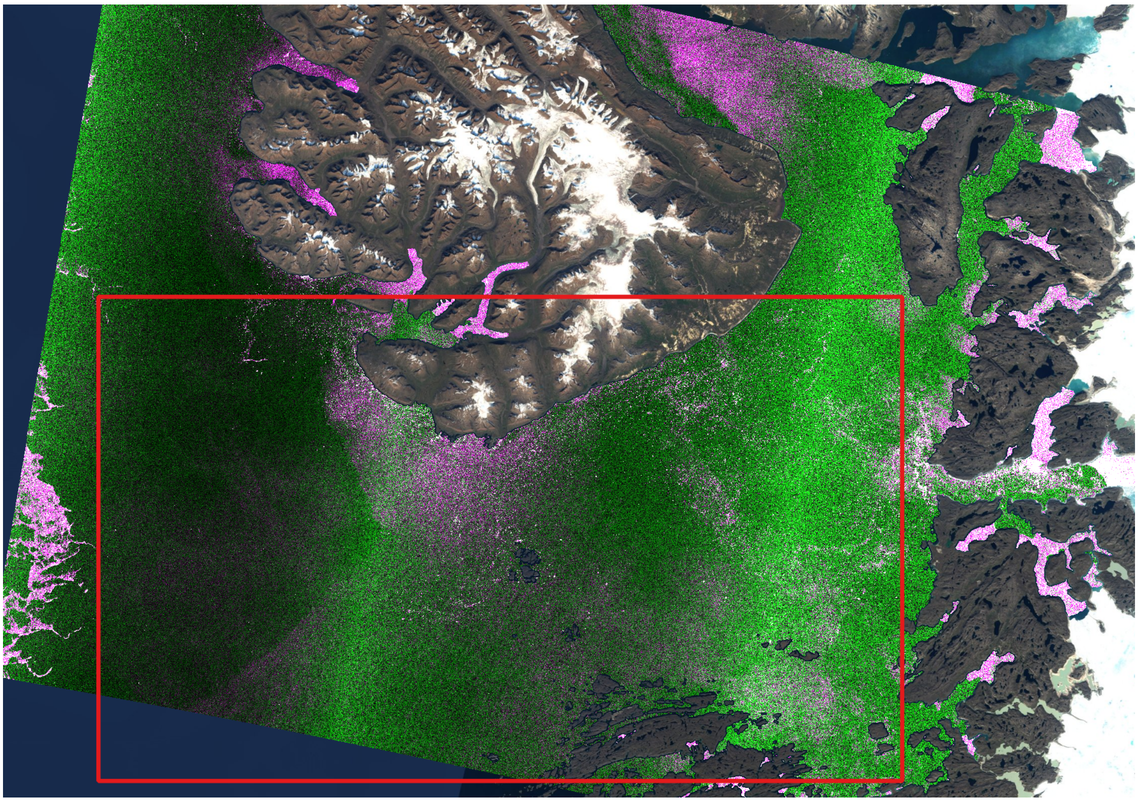

Figure 4.9: April 25, 2020, Disko Bay Sentinel-1 RGB Composite

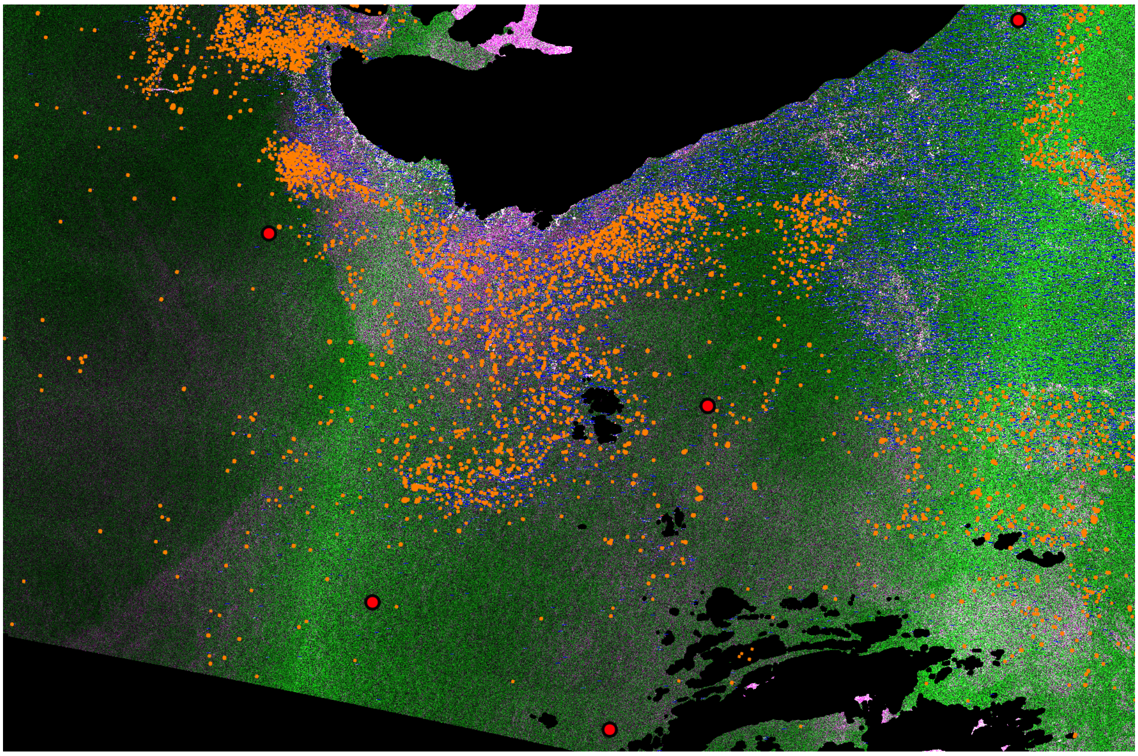

Figure 4.10: April 25, 2020, Validation area detection output and validation data.

The extent of the icebergs detected by DMI is again dependent on the estimated extent of sea-ice. For the date of April 25 there is no ice-chart available but the ice-chart for April 24[4] shows the empty polygon areas to be classified as sea-ice.

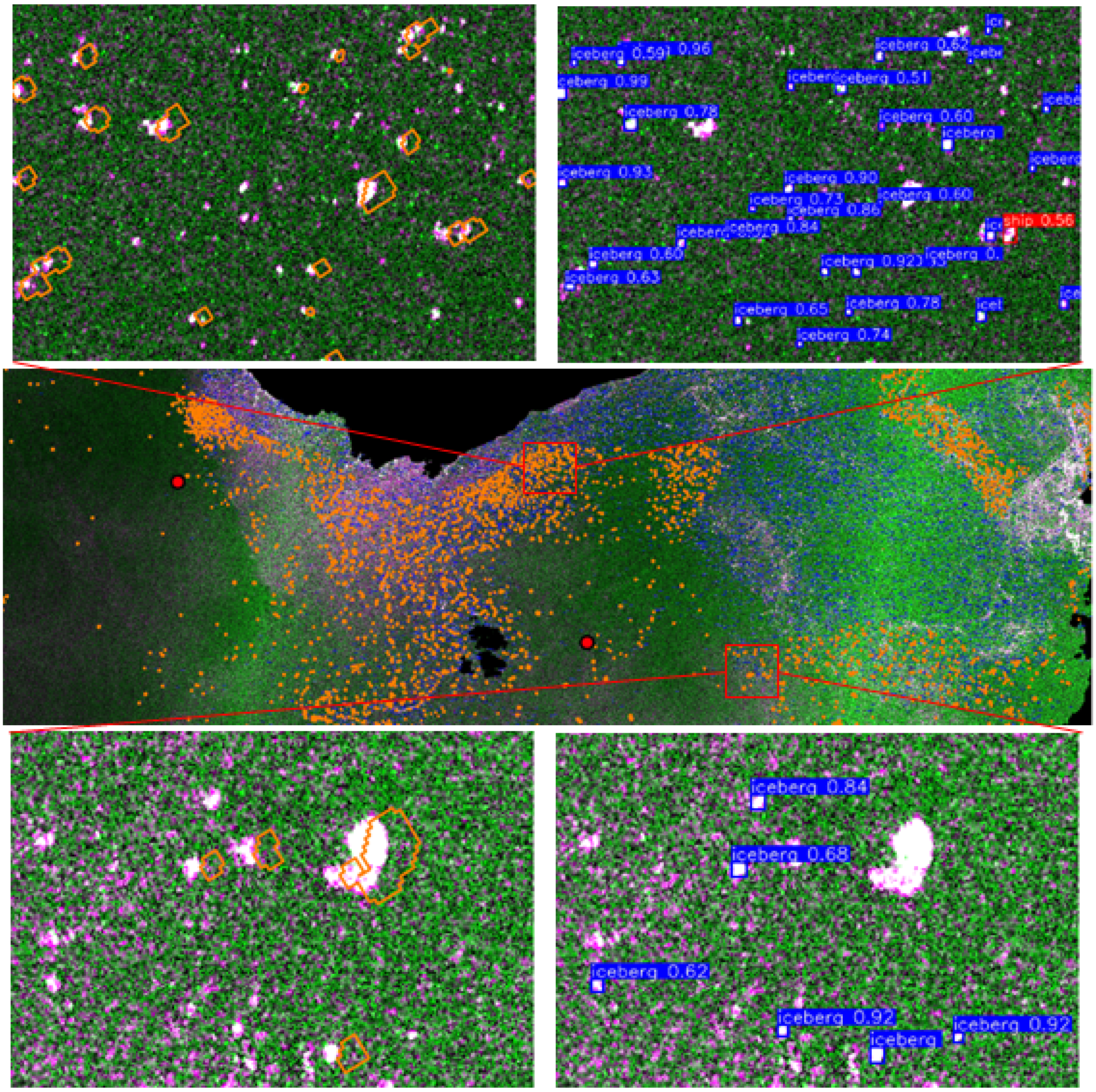

Figure 4.11: April 25, 2020. Closeup examples detection, left images show iceberg polygons from DMI and right images show model detections.

Within the validation area DMI detected 4601 icebergs, these were again validated with the outcomes of: Detected as iceberg, Detected as ship, or Not detected. An overview of the validation is seen in table 4.3 below.

Table 4.3: April 25, 2020. Validation overview

DMI Icebergs	**Detected as iceberg**	**Detected as ship**	**Not detected**
4601	2285	69	2247

[4]DMI: Ice chart central West Greenland, April 24, 2020

Out of the 4601 icebergs detected by DMI, 51.16% were detected by the model, again with a small amount wrongly classified as ships. The validated accuracy of 51.16% is a bit lower than the 54.95% achieved in the scene of April 18, but not a significant difference.

In the validation area there were 5 ships present at the satellite acquisition time, the AIS data point from these are shown as the red dots in figure 4.12. Out of these 5 ships, 3 have been detected as icebergs and 1 has correctly been detected as a ship. An actual accuracy assessment can not be made from only 5 data points, but the findings confirm the figure 4.4 statement of the models being well able to detect objects but struggling to classify these. Figure 4.12 shows the model predictions at the AIS data points.

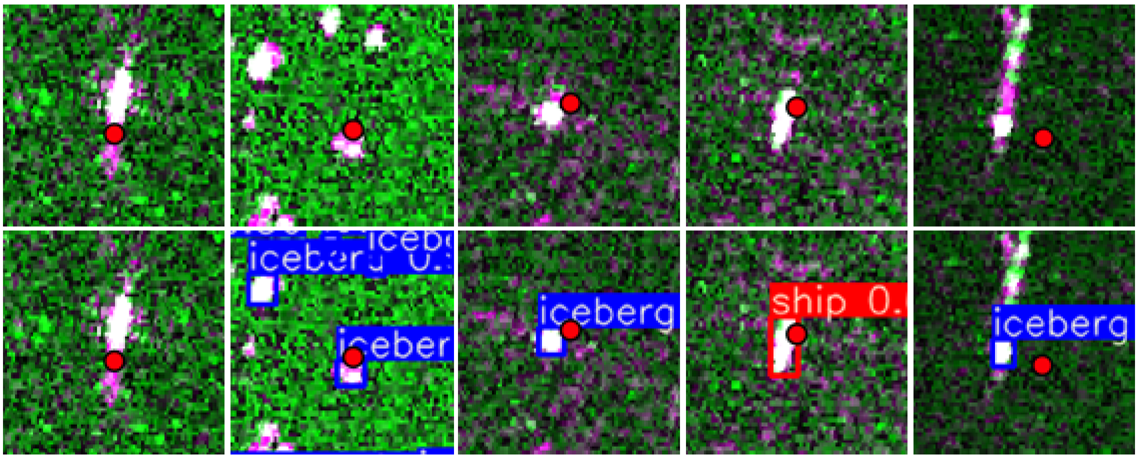

Figure 4.12: The 5 AIS data points in the validation area, top images are Sentinel-1 RGB composites and bottom images are the model detection output

4.3 Summary

Based on the icebergs detected by DMI and ship AIS data, the validation shows an overall iceberg detection accuracy of 51-55%, and a ship detection accuracy significantly lower. For each scene, the model predicted around 20.000 objects, and a very small amount of these where ships. This indicates that even though the model has issues with the ship class, the vast majority of objects where still classified correctly. Overall, the validation results correspond well with the estimated accuracy measures from the model training, the average precision is about the same and the issue regarding object classification has clearly shown.

The chosen validation procedure is the only feasible way to validate the model, but it still carries some issues. The issues regarding validation and difficulties facing the creation and training of a model with the desired capabilities, are elaborated on in the following chapter.

Discussion 5

The accuracies presented in chapter 4 are the results of a training and validation process with a multiple of different factors affecting the end result. The biggest of the issues facing the whole modelling process is the availability, quality and quantity of the required data for the model. These issues, and a general discussion about the project research questions, are elaborated in this chapter.

5.1 Training Issues

A deep learning model, of any kind and for any purpose, requires a very large amount of highly accurate data to train and detect properly. The model used in this project had the goal of detecting two object types, ships and icebergs, but the creation of data-sets for the two object classes raised a number of issues.

5.1.1 Icebergs

The main issues faced when creating a data set covering icebergs, are the lack of completeness in current automatic detection methods together with a missing definition of exactly what features are of interest to the user.
Firstly, while there are definitions and categorizations of what exactly is an iceberg and different types of icebergs (such as the definition by NOAA[1]), these are not used in the Greenlandic iceberg charting by Copernicus and DMI. With so much ice coming out of the Greenlandic glacier outlets, deciding upon what is of interest to the application and what is not, is very dependent on the given application and end user purpose.

When creating iceberg training data this lack of categorization imposes the issue of what objects to label and how to label them.

[1] NOAA. National Ocean Services, What is an iceberg?

Figure 5.1: Image from a camera station placed in the bottom of the fjord Nuup Kangerlua. Photo taken the 13th of August 2018

Figure 5.1 above shows an example of the real world situation at the glacier outflows. The picture highlights the complexity in labeling icebergs in the Greenlandic fjord. Some icebergs are clearly seen, but most of the ice are smaller pieces and patches of drift ice, which is difficult to exactly categorize.
When looking at such scenes from satellite radar, the complexity and decision making is up to the creator of the training data.

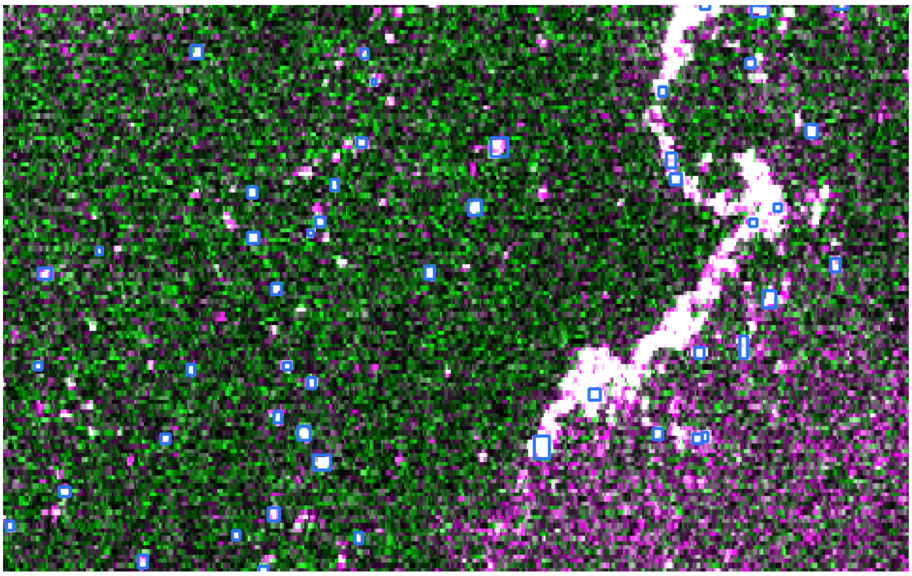

Figure 5.2: Example of Sentinel-1 iceberg training data

Figure 5.2 shows an example of iceberg training data used in the model and is a good indicator for a complex situation of data to be labeled. It could well be argued that too many objects are missing labeling, leaving them out of training, but in opposition to this one could say that too many smaller objects are included and that these are not of great importance. In figure 5.2 training objects are also seen to be located within the large piece of floating ice in the right side of the image, this also raises the question of 'icebergs' being present in other types of ice, or if such large pieces of floating should be in a class of its own or not be included at all.
Besides the issue of inconsistent training data causing inconsistent detections, the inconsistency is also confusing the model. When data is labeled as the example in figure 5.2 the labeled objects is passed on as icebergs and everything not labeled, is known to model as not being an iceberg, which in some cases will result in counter-productive training.
All training labels used in the project have been detected automatically with a CFAR algorithm, leaving these questions still to be decided upon. A manual labeling process would be much more time consuming but likely also output a more coherent data set.

5.1.2 Ships

Given that ships are well defined objects under constant monitoring, creating a training data set for this did not face the same issues as the iceberg data. The main issues in working with ship positioning data are the availability and quantity of ships in Arctic waters.
In Greenlandic waters, there are usually around 100 ships transmitting AIS data. A substantial amount of these are laying moored at the busy harbours of Nuuk, Sisimiut and Illulisat, leaving an even smaller amount of ships actually sailing. Having the rest of these ships sailing around different places of Greenland, the traffic of any given area is very sparse. This case was clearly proven in chapter 4.2, with only a handful or less ships being present in each Sentinel-1 scene. In comparison, hundreds of ships are sailing through Danish waters and these are covered by only a few Sentinel-1 scenes.

In the model validation this was clearly an issue, but the low amount of ships also poses a great issue in creating model-training data. In general, access to ship AIS data is costly and free services only provide limited access, this is the case for Greenland as well. Even if Greenland AIS data is acquired, the processing of Sentinel-1 data for all waters around Greenland is still a significant task.
The bottom line of this is that creating a training data set with ships around Greenland is not an impossible task, but given the major geographical extent and limited amount of ships, a very difficult task. The ship data used in project model was from Denmark where

historical AIS data is free and the processing is limited to a few Sentinel-1 scenes. Denmark and Greenland are however, covered by different polarizations, raising the question of the impact of training and detecting in different polarizations.

5.2 Research Question 1

How does the different polarizations used in the Sentinel-1 data acquisition affect the model results?

Due to the nature of the geographic distribution of the two object classes, it was inevitable to create training data in two different polarizations. As most of the world is covered with the same polarization, this is not an issue for most cases and likely the reason why it is not very well covered in relevant literature.

The training of the two classes in the model is based on different polarizations, HH+HV for the Greenland areas and VH+VV for the Danish areas.
To which degree this factor has impacted the detection results is difficult to say, but it certainly has an impact as the model has simply never 'seen' a ship in the HH+HV polarization. To test whether the model has been successfully trained but impacted by the polarization, a full detection of a Sentinel-1 scene covering Denmark is carried out. This detection should not find any icebergs, and the greater amount of ships will give an indication of the models ability to detect these.
The Sentinel-1 scene being used was acquired on May 17, 2020.

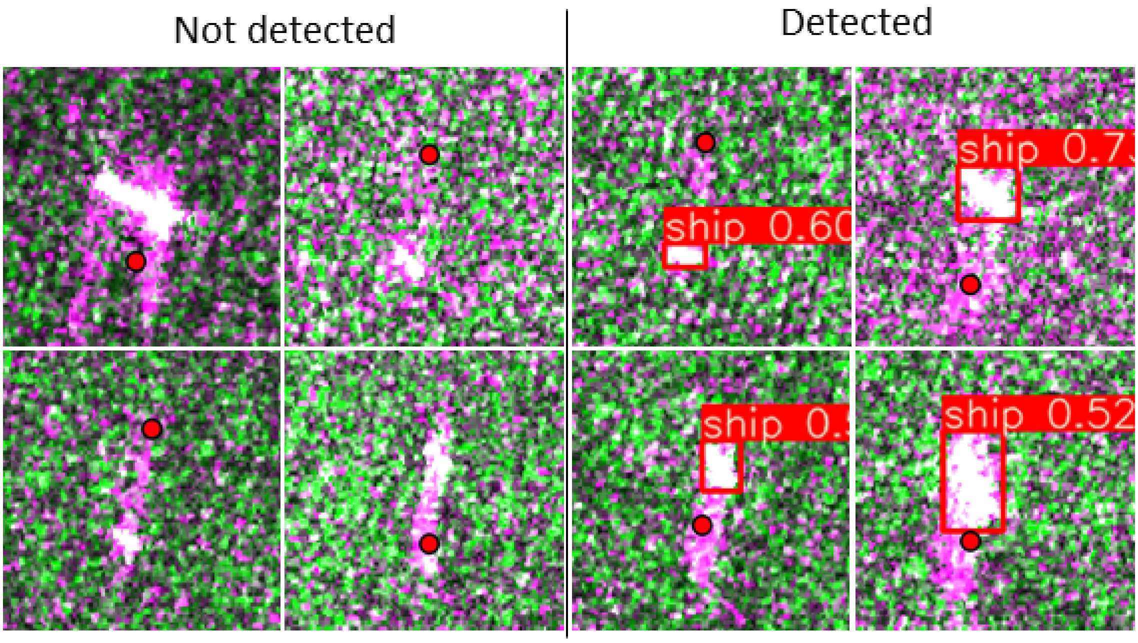

Figure 5.3: Example of ship detections in Denmark

For the whole scene, the detection returned 1 iceberg and 108 ships. Though the iceberg obviously is a wrong detection, the test proves that the model is capable of differentiating between the two object classes.
The 108 ships detections are validated with AIS data, the scene contains 110 AIS data points. The points are validated as either detected or not detected.

Table 5.1: May 17, 2020. Ship validation overview

AIS data point	**Detected**	**Not Detected**
110	63	46

This validation returns a ships class accuracy of 57.27%, an accuracy that well resembles the estimated 55.7% training accuracy (see mAP in section 4.1.3).
The ships detected here are in the same polarization as those used in the training, this gives an indication that the polarization does have an effect in what the model is able to predict. When training and detection is done within the same polarization, the iceberg and ship classes are each capable of reaching accuracies around the models 55%, thus proving that the model is functioning but is affected by the cross-polarization.

5.3 Research Question 2

With the use of deep learning, what degree of accuracy can be achieved in detecting and classifying floating ocean objects?

Research question two is directly aimed at the overall project purpose of discriminating between icebergs and ships in Arctic waters. As stated in the introduction in section 1, the current detection issues is discriminating the between the two objects when both are present within the same geographical extent.
What the project has proved is that a model can be trained to recognize and detect the two object types, but as far as this project goes, not within the same geographical extent.

Within the same detection scene, it has not been possible to discriminate between the two objects and no exact measure for accuracy in discrimination can be derived from the findings of this project. Though the quest to answer this research question has not been fulfilled, the project shows major promise in detection capabilities and therefore the possibility of solving what the project set out to do.

All measures of model accuracy have shown around 55%, and though this number is not impressive in itself, it must be considered how much data were used to achieve that number. Training on 519 ships and 1760 icebergs, is a very small data set in a deep learning context. The creator of the Yolo object detection algorithms states that the minimum number of objects that should be used in training is 2000 per class [2]. As proved in this study, decent results can be achieved with less data, but with the amount of data used, high accuracies are unlikely.

The accuracy measures highlighted in section 4.1 are all good for monitoring model performance and to get indications of training issues, but the bottom line of any well performing deep learning model is the amount of training data. The very basis in image recognition techniques is that a model is learning how objects are looking by seeing as many of them as possible, so the best way of increasing accuracy will usually be to use more data.

In future works, the issues regarding icebergs and difficulty in labeling and validating them would have to be addressed. DMI is currently the only large-scale provider of iceberg data for Greenland, and the uncertainties in this data makes it questionable if it can be used as 'ground truth'. The data from DMI is sufficient for detection comparisons, but does not completely reflect the real world scenarios.
A future approach could be to focus on building a large ship data set and only detecting ships. This would of course be an application not capable of detecting any sort of ice, but it could be argued that if ships can be detected among icebergs then the model is capable of discriminating the objects.

[2] GitHub - AlexeyAB/darknet

6 Conclusion

The project problem statement highlighted the need for a strong detection algorithm being able to discriminate between icebergs and ships, and sought out to solve this issue by applying state of the art deep learning detection algorithms. Based on the results achieved in the project, the overall problem remains unsolved.

Despite this, the project stands as a very good proof of concept of the possibilities within deep learning and remote sensing. With still increasing documentation and solutions for implementing and using deep learning applications, the project states how such an application can be used to fit a very specific purpose. The workflow carried out, has shown how data from the traditional fields of GIS and Remote Sensing can be transformed and used in an object detector designed to detect objects that are completely unrelated to the project goals. The field of deep learning has come so far that advanced knowledge in neural networks is no longer required to utilize the algorithms, allowing researchers within GIS to focus even more on creating and maintaining the data needed to achieve satisfactory results.
The implemented model has been proved to work in different geographic regions, and besides that, has proved a gap in the initial project expectations and what is actually achievable. Within the different polarizations the results where satisfactory, but working between different polarizations is causing issues. The cross polarization is only an issue working in Arctic and Antarctic areas, so most of the relevant literature does cover the issue. However, with still increasing interest in the Arctic, this is something that must, and most likely will, be investigated further.

The project model has proven to function well and achieve results that correspond well to what can be expected. To reach a point where the project goal can be achieved, the main task is to work on creating a far larger ship data set, one that also covers Arctic waters, and setting up more specific goals for how to map ice and icebergs. Lastly, implementing actual ground truth data, such as images from a stationary camera station, could very well benefit the creation and validation of model data.

Bibliography

Alexe et al., 2012. Bogdan Alexe, Thomas Deselaers and Vittorio Ferrari. *Measuring the objectness of image windows.* IEEE transactions on pattern analysis and machine intelligence, 34(11), 2189–2202.

Ammar et al., 2019. Adel Ammar, Anis Koubaa, Mohanned Ahmed and Abdulrahman Saad. *Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3.* arXiv preprint arXiv:1910.07234.

Bilel Benjdira, Taha Khursheed, Anis Koubaa, Adel Ammar and Kais Ouni, *2019*. Bilel Benjdira, Taha Khursheed, Anis Koubaa, Adel Ammar and Kais Ouni. Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pages 1–6. IEEE, 2019.

Bourbigot et al., 2016. M Bourbigot, H Johnsen, R Piantanida and G Hajduch. *Sentinel-1 product definition.* MDA Document Number: SEN-RS-52-7440.

Chang et al., 2019. Yang-Lang Chang, Amare Anagaw, Lena Chang, Yi Chun Wang, Chih-Yu Hsiao and Wei-Hong Lee. *Ship detection based on YOLOv2 for SAR imagery.* Remote Sensing, 11(7), 786.

Danish Ministry of Defence, 2016. Danish Ministry of Defence. *Forsvarsministeriets fremtidige opgaveløsning i Arktis.*

Dechesne et al., 2019. Clément Dechesne, Sébastien Lefèvre, Rodolphe Vadaine, Guillaume Hajduch and Ronan Fablet. *Ship identification and characterization in Sentinel-1 SAR images with multi-task deep learning.* Remote Sensing, 11(24), 2997.

DNV GL, 2015. DNV GL. *Marine Environmental Risk Assesment - Greenland.* (2014-0951, Rev. E).

Eastwood et al., 2019. Steiner Eastwood, Juha Karvonen, Frode Dinessen, Andrew Fleming, Leif Toudal Pedersen, Roberto Saldo, Jørgen Buus-Hinkler, Bruce Hackett, Fanny Ardhuin, Matilde Brandt Kreiner and C. Wettre. *Quality Information Document for OSI TAC Sea Ice products.* Copernicus Marine Environment Monitoring Service, (2.7).

Everingham et al. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.* http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Filipponi, 2019. Federico Filipponi. *Sentinel-1 GRD Preprocessing Workflow.* 18(1), 11.

Heiselberg and Heiselberg, 2017. Peder Heiselberg and Henning Heiselberg. *Ship-Iceberg Discrimination in Sentinel-2 Multispectral Imagery by Supervised Classification.* Remote Sensing, 9(11), 1156.

Hijmans et al., 2018. R Hijmans, N Garcia and J Wieczorek. *Global Administrative Areas Database (GADM) Version 3.6.* UN: New York, NY, USA.

Hill, 2010. Brian Hill. *Ship Collision with Iceberg Database.*

Jocher et al., April 2019. Glenn Jocher, guigarfr, perry0418, Ttayu, Josh Veitch-Michaelis, Gabriel Bianconi, Fatih Baltac, Daniel Suess, WannaSeaU and IlyaOvodov. *ultralytics/yolov3: Rectangular Inference, Conv2d + Batchnorm2d Layer Fusion*, 2019. [ONLINE] Available at: https://doi.org/10.5281/zenodo.2672652.

Kuznetsova et al., 2020. Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig and Vittorio Ferrari. *The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale.* IJCV.

Le Traon et al., 2019. Pierre Yves Le Traon, Antonio Reppucci, Enrique Alvarez Fanjul, Lotfi Aouf, Arno Behrens, Maria Belmonte, Abderrahim Bentamy, Laurent Bertino, Vittorio Ernesto Brando, Matilde Kreiner et al. *From observation to information and users: the Copernicus Marine Service perspective.* Frontiers in Marine Science, 6, 234.

Lin et al., 2014. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C Lawrence Zitnick. *Microsoft coco: Common objects in context.* pages 740–755.

Liu et al., 2016. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C Berg. *Ssd: Single shot multibox detector.* pages 21–37.

Long et al., 2017. Yang Long, Yiping Gong, Zhifeng Xiao and Qing Liu. *Accurate object localization in remote sensing images based on convolutional neural networks.* IEEE Transactions on Geoscience and Remote Sensing, 55(5), 2486–2498.

Manning et al., 2008. Christopher D Manning, Prabhakar Raghavan and Hinrich Schütze. *Introduction to information retrieval.* Cambridge university press, 2008.

Mundhenk et al., 2016. T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla and Kofi Boakye. *A large contextual dataset for classification, detection and counting of cars with deep learning.* pages 785–800.

Ocean Conservancy, 2017. Ocean Conservancy. *Navigating the North: An Assessment of the Environmental Risks of Arctic Vessel Traffic.* Ocean Conservancy, 2017.

Ouchi, 2016. Kazuo Ouchi. *Current status on vessel detection and classification by synthetic aperture radar for maritime security and safety.* pages 3–5.

Pelich et al., 2019. Ramona Pelich, Marco Chini, Renaud Hostache, Patrick Matgen, Carlos Lopez-Martinez, Miguel Nuevo, Philippe Ries and Gerd Eiden. *Large-Scale Automatic Vessel Monitoring Based on Dual-Polarization Sentinel-1 and AIS Data.* Remote Sensing, 11(9), 1078.

Porter et al., 2018. Claire Porter, Paul Morin, Ian Howat, Myoung-Jon Noh, Brian Bates, Kenneth Peterman, Scott Keesey, Matthew Schlenk, Judith Gardiner, Karen Tomko, Michael Willis, Cole Kelleher, Michael Cloutier, Eric Husby, Steven Foga, Hitomi Nakamura, Melisa Platson, Jr. Wethington, Michael, Cathleen Williamson, Gregory Bauer, Jeremy Enos, Galen Arnold, William Kramer, Peter Becker, Abhijit Doshi, Cristelle D'Souza, Pat Cummens, Fabien Laurier and Mikkel Bojesen. *ArcticDEM*, 2018. [ONLINE] Available at: https://doi.org/10.7910/DVN/OHHUKH.

Desmond Power, Carl Howell, Kelley Dodge, Francesco Scibilia, Jan Richard Sagli, Richard Hall et al., 2018. Desmond Power, Carl Howell, Kelley Dodge, Francesco Scibilia, Jan Richard Sagli, Richard Hall et al. Towards Automation of Satellite-Based Radar Imagery for Iceberg Surveillance-Machine Learning of Ship and Iceberg Discrimination. In *OTC Arctic Technology Conference*. Offshore Technology Conference, 2018.

Raihan et al., 2019. Md Aamir Raihan, Negar Goli and Tor M Aamodt. *Modeling deep learning accelerator enabled GPUs.* pages 79–92.

Redmon and Farhadi, 2018. Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement.* arXiv.

Ren et al., 2015. Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks.* pages 91–99.

Ressel et al., 2015. Rudolf Ressel, Anja Frost and Susanne Lehner. *NAVIGATION ASSISTANCE FOR ICE-INFESTED WATERS THROUGH AUTOMATIC ICEBERG DETECTION AND ICE CLASSIFICATION BASED ON TERRASAR-X IMAGERY.* International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences.

Rezatofighi et al., June 2019. Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid and Silvio Savarese. *Generalized Intersection over Union.*

Rodriguez et al., 2005. Ernesto Rodriguez, CS Morris, JE Belz, EC Chapin, JM Martin, W Daffer and S Hensley. *An assessment of the SRTM topographic products.*

Russakovsky et al., 2015. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein et al. *Imagenet large scale visual recognition challenge.* International journal of computer vision, 115(3), 211–252.

Carlos Santamaria, Harm Greidanus, Mélanie Fournier, Torkild Eriksen, Michele Vespe, Marlene Alvarez, Virginia Fernandez Arguedas, Conor Delaney and Pietro Argentieri, 2016. Carlos Santamaria, Harm Greidanus, Mélanie Fournier, Torkild Eriksen, Michele Vespe, Marlene Alvarez, Virginia Fernandez Arguedas, Conor Delaney and Pietro Argentieri. Sentinel-1 contribution to monitoring maritime activity in the arctic. In *Proceedings of the ESA Living Planet Symposium, Prague, Czech Republic*, pages 9–13, 2016.

Simonyan and Zisserman, 2014. Karen Simonyan and Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556.

Szegedy et al., 2017. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke and Alexander A Alemi. *Inception-v4, inception-resnet and the impact of residual connections on learning.*

Vakalopoulou et al., 2015. Maria Vakalopoulou, Konstantinos Karantzalos, Nikos Komodakis and Nikos Paragios. *Building detection in very high resolution multispectral data with deep learning features.* pages 1873–1876.

Van Etten, 2019. Adam Van Etten. *Satellite imagery multiscale rapid detection with windowed networks.* pages 735–743.

Van Etten et al., 2018. Adam Van Etten, Dave Lindenbaum and Todd M Bacastow. *Spacenet: A remote sensing dataset and challenge series.* arXiv preprint arXiv:1807.01232.

Wang et al., 2019. Yuanyuan Wang, Chao Wang, Hong Zhang, Yingbo Dong and Sisi Wei. *A SAR dataset of ship detection for deep learning under complex backgrounds.* Remote Sensing, 11(7), 765.

Wu et al., 2015. Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang and Gang Sun. *Deep image: Scaling up image recognition.* arXiv preprint arXiv:1501.02876, 7(8).

Zhang et al., 2018. Zhengxin Zhang, Qingjie Liu and Yunhong Wang. *Road extraction by deep residual u-net.* IEEE Geoscience and Remote Sensing Letters, 15(5), 749–753.

Zhao et al., 2019. Jiafei Zhao, Rongkun Jiang, Xuetian Wang and Hongmin Gao. *Robust CFAR Detection for Multiple Targets in K-Distributed Sea Clutter Based on Machine Learning.* Symmetry, 11(12), 1482.