Deep Learning for Detecting and Classifying Ocean Objects

Application of YoloV3 for Iceberg–Ship Discrimination

Hass, Frederik Seerup; Arsanjani, Jamal Jokar

Published in:
ISPRS International Journal of Geo-Information

DOI (link to publication from Publisher):
10.3390/ijgi9120758

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Hass, F. S., & Arsanjani, J. J. (2020). Deep Learning for Detecting and Classifying Ocean Objects: Application of YoloV3 for Iceberg–Ship Discrimination. *ISPRS International Journal of Geo-Information*, *9*(12), Article 758. https://doi.org/10.3390/ijgi9120758

International Journal of
Geo-Information

MDPI

Article

Deep Learning for Detecting and Classifying Ocean Objects: Application of YoloV3 for Iceberg–Ship Discrimination

Frederik Seerup Hass and Jamal Jokar Arsanjani *

Department of Planning, Geography and Surveying, Aalborg University Copenhagen, A.C. Meyers Vænge 15, 2450 Copenhagen, Denmark; frederiksh@plan.aau.dk

* Correspondence: jja@plan.aau.dk

Received: 31 October 2020; Accepted: 18 December 2020; Published: 19 December 2020

Abstract: Synthetic aperture radar (SAR) plays a remarkable role in ocean surveillance, with capabilities of detecting oil spills, icebergs, and marine traffic both at daytime and at night, regardless of clouds and extreme weather conditions. The detection of ocean objects using SAR relies on well-established methods, mostly adaptive thresholding algorithms. In most waters, the dominant ocean objects are ships, whereas in arctic waters the vast majority of objects are icebergs drifting in the ocean and can be mistaken for ships in terms of navigation and ocean surveillance. Since these objects can look very much alike in SAR images, the determination of what objects actually are still relies on manual detection and human interpretation. With the increasing interest in the arctic regions for marine transportation, it is crucial to develop novel approaches for automatic monitoring of the traffic in these waters with satellite data. Hence, this study aims at proposing a deep learning model based on YoloV3 for discriminating icebergs and ships, which could be used for mapping ocean objects ahead of a journey. Using dual-polarization Sentinel-1 data, we pilot-tested our approach on a case study in Greenland. Our findings reveal that our approach is capable of training a deep learning model with reliable detection accuracy. Our methodical approach along with the choice of data and classifiers can be of great importance to climate change researchers, shipping industries and biodiversity analysts. The main difficulties were faced in the creation of training data in the Arctic waters and we concluded that future work must focus on issues regarding training data.

Keywords: deep learning; object detection; ocean objects; synthetic aperture radar; classification; YoloV3

1. Introduction

Synthetic aperture radar (SAR) is a very capable tool for ocean monitoring, especially in regard to detecting oil spills, mapping ice, and locating unidentified ships. Since SAR is based on active remote sensing, it has capabilities to function both during the day and night through any weather conditions, and spaceborne SAR products therefore allow for constant and seamless monitoring of vast areas. Mapping and detecting objects through SAR data are based on measurements of the surface texture properties of different object types. Depending on the application, different methods are applied in SAR-based mapping, sea ice charting is mostly based on backscatter values measured from observations [1] where iceberg detection relies on adaptive threshold algorithms that detect sudden increases in backscatter values between an object and the ocean. [2]. The same underlying methodology algorithm is used in ship traffic monitoring [3], allowing authorities to monitor and detect vessels that are not traceable with the automatic identification system (AIS) or other reporting signals [4].

While object detection in SAR data has served a variety of applications and has solid ground on well-established methodologies, basing the detection on adaptive thresholding algorithms can be challenging and the determination of an object is often still up to human interpretation. In areas where there are both a great number of icebergs and ships, it is challenging, even for a human interpreter, to tell them apart. A recent study on automatic detection of ships in 2000 Sentinel-1 SAR images covering Arctic waters using the adaptive threshold CFAR algorithm reached the following conclusion:

> *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 SAR based 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.* [5]

The capabilities and possibilities within deep learning object detection are increasing at a fast pace. Today, there is a significant amount of deep learning frameworks to choose from, with some of the most notable being Faster R-CNN [6], SSD [7], YOLO [8] and Resnet [9]. Most of these are being utilized in image object detection, locating a vast number of objects in everyday photos. The algorithms are being utilized for object detection in aerial, satellite photography and SAR data at an increasing degree.

In the creation of a training dataset for SAR ship detection, [10] utilised and evaluated different deep learning models such as SSD, Faster R-CNN and RetinaNet. All of these models achieved accuracies between 88–91%, with RetinaNet achieving the highest accuracy but at the cost of the longest training time. The study [11] managed to classify types of ships in Sentinel-1 images, by training OpenSAR data on a multi-task neural network. The research studies achieved accuracies of 96–97% on small image tiles and 85% on larger Sentinel-1 scene patches. The creators of OpenSAR [10] did not test the Yolo object detector as it has been proven to be fast and accurate [8,12]. A study on ship detection [13] compared training and detection between Faster-RCNN and YoloV2 and managed to achieve 90% accuracy with the Yolov2 detector, 20% higher than Faster-R-CNN, and proved significantly better training and detection times. The Yolo framework has gained increased traction over the last few years, proving good detection capabilities and great detection speeds and training times. In remote sensing applications, it has outperformed the field of established algorithms [14], and thus making the YoloV3 algorithm the ideal choice of algorithm for the purpose of this study.

Given the recent advances and results produced by deep learning image recognition and object detection algorithms, it is the inevitable way to proceed for the future of satellite ocean monitoring. The main objective of this study was set to implement YoloV3, which has become a popular and reliable object detector, in order to investigate its usefulness and what challenges arise in the task of iceberg–ship discrimination in SAR data.

2. Materials and Methods

2.1. Data

Training a deep neural network for object detection and classification purposes requires a large portion of labelled images serving as training and validation data. These data were generated through a combination of an automatic detection algorithm (CFAR) and manual digitization of objects in a number of Sentinel-1 interferometric wide (IW) swath images over various locations. The automatic detection was set to locate and outline as many objects as possible in the size range of 20–480 m, this helped to quicken the labelling process and ease the assessment of the large image scenes. All detected objects were manually inspected for precise outlining and removal of false objects; if missing objects were found in the images, these were outlined manually. Labelling the object as two object classes, ship, and iceberg, was achieved based on AIS data for the selected areas. The ship class was mainly gained from the Danish study areas, where the AIS data were sorted accordingly to the Sentinel-1 acquisition timestamp, and all objects were manually correlated to nearby AIS data points. The iceberg class was created from the Greenland study areas, where the AIS data were used to ensure that the outlined objects were not correlating to any nearby AIS data points. There is, to our knowledge, no complete and accurate coastline data set for Greenland, and with some areas prone to high tides, near-surface rocks can cause false objects to appear. Sentinel-2 optical imagery was used to assist quality assurance of the data e.g., removal of surface rocks. It was emphasized to use Sentinel-1 data of the same areas but captured from both satellites and with different orbits, paths, and directions, this ensures that training objects are seen at different angles and from different sides. The study areas and Sentinel-1 scenes can be seen in Figure 1. Over the different areas, a total number of 2279 objects were digitized in 7 different Sentinel-1 scenes. See details on the labelled data and satellite information in Table 1.

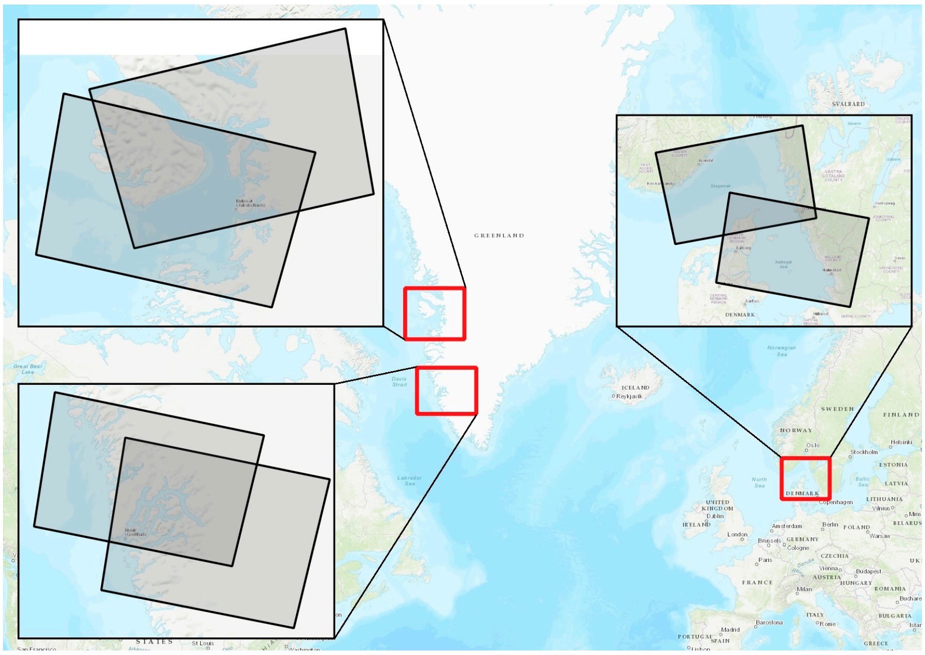

Figure 1. Study areas outlined in red and, Sentinel-1 scenes used within the study areas outlined in black.

Table 1. Information on acquired satellite data for annotation of ship and icebergs.

Location	Satellite	Acquisition Time	Polarization	Path & Angle [1]	Object Class	No. of Objects
Greenland	Sentinel-1A	24/11/2019 (09:45:23–09:45:52)	HH+HV	Descending X34°	Iceberg	1150
Greenland	Sentinel-1B	30/11/2019 (09:44:41–09:45:10)	HH+HV	Descending X34°	Iceberg	613
Denmark	Sentinel-1A	08/10/2019 (05:32:31–05:32:56)	VH + VV	Descending X34°	Ship	78
Denmark	Sentinel-1B	19/11/2019 (05:31:39–05:32:04)	VH + VV	Descending X34°	Ship	102
Denmark	Sentinel-1A	23/11/2019 (17:02:05–17:02:30)	VH + VV	Ascending X34°	Ship	118
Denmark	Sentinel-1B	16/01/2020 (17:01:20–17:01:45)	VH + VV	Ascending X34°	Ship	112
Denmark	Sentinel-1B	23/02/2020 (05:31:35–05:32:00)	VH + VV	Descending X34°	Ship	108

[1] Incidence angle measured at mid swath.

The two Greenland study areas (Disko Bay and Nuup Kangerlua) were selected based on the expected amount and density of icebergs. While icebergs are common sights all over the coast of Greenland and Eastern Canada as well, the selected areas have glacier outlets from the icesheet flowing directly into them ensuring a stable flow of icebergs during the warmer months of the year. The Danish study area covers the ocean Kattegat, this is a busy shipping route due to fact that all cargo from the Baltic areas travel through here. Choosing a waterway within was however mainly chosen based on the fact that AIS data are made free by the Danish Maritime Authority.

The acquired satellite data for the study is seen in Table 1, it is important to note the different polarizations at the Greenland and Denmark locations and associated object classes. Given the geographical extent of the polarizations and the nature of the locations of the objects, as the Arctic has a lack of ships but a great number of icebergs and vice versa, it is not feasible for a study of this scale to produce a data set with all objects in the same polarization.

The Sentinel-1 data are converted into RGB composites with the individual polarization used as image bands. Earlier studies have indicated better ship detection capabilities in the dual-polarization (VV or HH) modes [10,15], but since the goal is not solely ship detection, it was decided to also include the cross-polarization in composite. Thus, making the Sentinel-1 RGB colour composite structured as follows:

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

The initial objects are outlined as vectors in the given Sentinel-1 images (see Figure 2), and they are converted into the darknet annotation format with text files containing information on the label class of objects and positions in the image. The Sentinel-1 images are cropped into image tiles in the sizes of 640 × 640 pixels, resulting in a total of 1609 images with corresponding label files. Code for darknet conversion is available at GitHub (see Supplementary Materials). Out of these images, 20% (322) are selected as validation data, and the remaining 80% (1288) for model training.

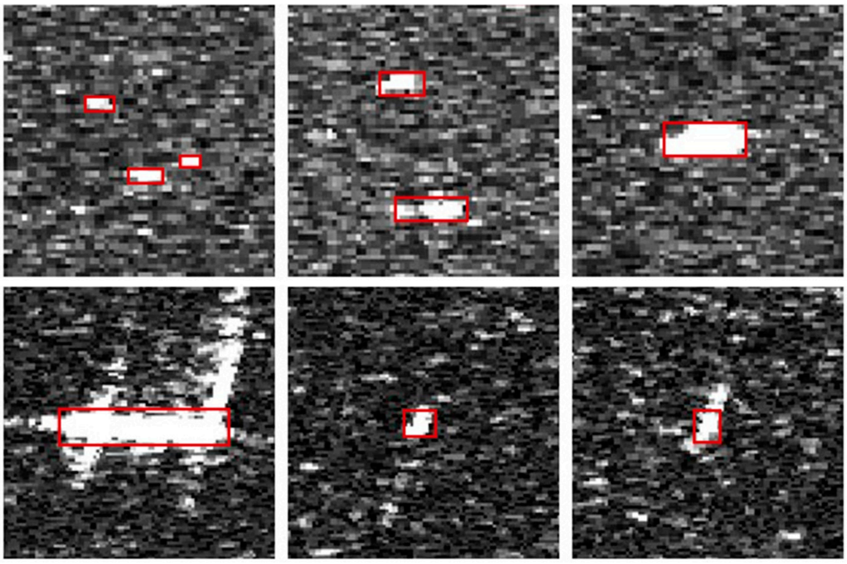

Figure 2. Samples of training objects' bounding boxes, icebergs in top row and ships in bottom row. The targets are shown in single-polarization greyscale, HH (**top**) and VH (**bottom**).

2.2. *YoloV3 Model Architecture*

The YoloV3 algorithm implemented in this study is based on Darknet-53, a 53-layer deep convolutional neural network (CNN) with residual connections. The traditional CNN object detectors function as two stage detectors that first have to identify individual regions of interest in the image and then carry out bounding box detection within these regions. The two-stage detection performs at competitive levels but at slow speeds, and while the improvements to speed have been made in further developments such as the Fast R-CNN and Faster R-CNN, they cannot achieve equal training time to the single stage detection. Yolo is a single stage detector that does not need to divide the image into separate regions, but instead handles the full image at once and hence the name: You Only Look Once. This is achieved by creating feature maps consisting of grid cells through a 3 level pyramid-like resampling of the image [8]. The resampling of the image is completed at the levels of 32, 16 and 8 in the individual feature maps, with a 1 × 1 detection kernel for each layer that contains the underlying image cells. The kernel is a 3-dimensional array with the shape of 1 × 1(Bx(5C)), with B being the number of bounding boxes per grid cell (3 as standard) and C, more importantly, being the number of classes for the model to predict. The shape of the kernel is an important factor when considering the size of the input images for the model to train and predict. With images of 640 × 640 pixels, the finest scale prediction boxes have the size of 20 × 20 pixels. The predictions made on each feature map are passed through the upsampling layers and residual connections to perform detections at the original scale without loss of information from the finer scales. The biggest advantage of YoloV3 over its predecessor YoloV2 is the scaling of the image and predictions made on each level and hereby its ability to detect smaller objects. Furthermore, the scaling causes the number of bounding boxes for prediction to increase by a great magnitude. With the case of size 640 × 640 images, the number of prediction boxes is 25,920 per image.

2.3. Training

Model training was carried out on a NVIDIA Quadro M4000 GPU with 8 GB of memory, with a training time of 4.9 min per epoch. With the adaptive learning rate algorithms usually yielding higher model accuracies than static ones [16], the Adam learning rate optimizer was chosen over the static stochastic gradient descent (SGD). Both the basic SGD and its further adaptive developments are popular in neural network applications [17], but given the findings of [18,19], which proved Adam's usefulness on relatively small datasets (less than 1000 images), the Adam optimizer is chosen for the model of this study. Hyperparameter setting was completed based on studies who successfully implemented Yolo in remote sensing cases. The author of the Yolo-based "Yolt" model [14] suggests implementing the same hyperparameters as the Yolo model. The default parameters of YoloV3 are 0.001, 0.9 and 0.0005 for the learning rate momentum and weight decay, respectively, and we decided to keep these values. There is some variance in proposed learning rate settings, but studies applying Yolo to aerial imagery have succeeded with a learning rate of 0.001 [12] and have proven that this value produces higher precision [20]. The model was trained for a total of 350 epochs, resulting in a total training time of 27 h. Due to limitations in computer memory, the batch size was set at 4.

2.4. Evaluation Metrics

The model is evaluated using the following metrics: Precision, Recall and F1, which are used to measure the model detection performance [21]. 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:

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

$$Recal = \frac{TP}{TP + FN}$$

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

The precision and recall metrics both measure the model detection performance, but each account for different factors in the detection process. 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 often move in opposite directions in relation to each other; high precision causes low recall and vice versa. The F1 score accounts for these measure biases and is thereby perceived as an overall accuracy measure [21]. 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.

Generalized intersection over union (GIoU) measures how well the model predicts object bounding boxes. The GIoU is developed from the standard intersection over union (IoU), a metric that measures how well a predicted bounding box intersects the ground truth bounding box. This metric only returns a value if there is an 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 return a value of 0. In these cases, the GIoU would still return a value, indicating if the boxes were close to each other [22]. The GIoU is calculated as:

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

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

Mean average precision (mAP) 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 both are 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 [21]. The mAP of 0.5 states that only objects with an IoU threshold above 0.5 (having 50% overlap) were used in this metric.

3. Results

3.1. Training Evaluation

The model was trained for 350 epochs and evaluated using the scores *precision, recall, F1, GIoU, and mAP*, as seen in Table 2. The scores are calculated from model validation data, consisting of 321 image tiles. The F1 score and mAP score are seen to follow each other closely, given indications on the training at different stages, the model quickly reached F1 and mAP scores of ~0.4, whereafter they continue to increase but at a slower rate. At the end of the training, the model achieved the accuracy scores of F1 = 0.530 and mAP = 0.557. The GIoU is steadily decreasing, indicating that the model is becoming better at correctly locating targets.

Table 2. Training metrics at different iteration stages.

Epoch	*Precision*	*Recall*	*F1 Score*	*GIoU*	*mAP*
100	0.656	0.321	0.430	2.14	0.407
200	0.583	0.549	0.541	1.58	0.542
300	0.493	0.610	0.534	1.29	0.548
350	0.476	0.600	0.530	1.16	0.557

Comparisons between input images and model predictions are shown in Figure 3. Visually inspecting the predictions, it can be seen that the detector struggles to detect the largest of the objects while having good detection capabilities for the smaller objects, though also showing weakness in dense object situations. Most of the ships detected are false positives, as these are icebergs wrongly classified as ships.

Figure 3 shows the prediction carried out in Western Greenland, where there is an abundance of icebergs but very few ships. The vast majority of objects here are icebergs, and the model correctly detects them; however, the low number of ships makes it difficult to evaluate exactly how well the ship detection is performing. Therefore, we tested the model at the Danish study site, while expecting that it should only detect ships.

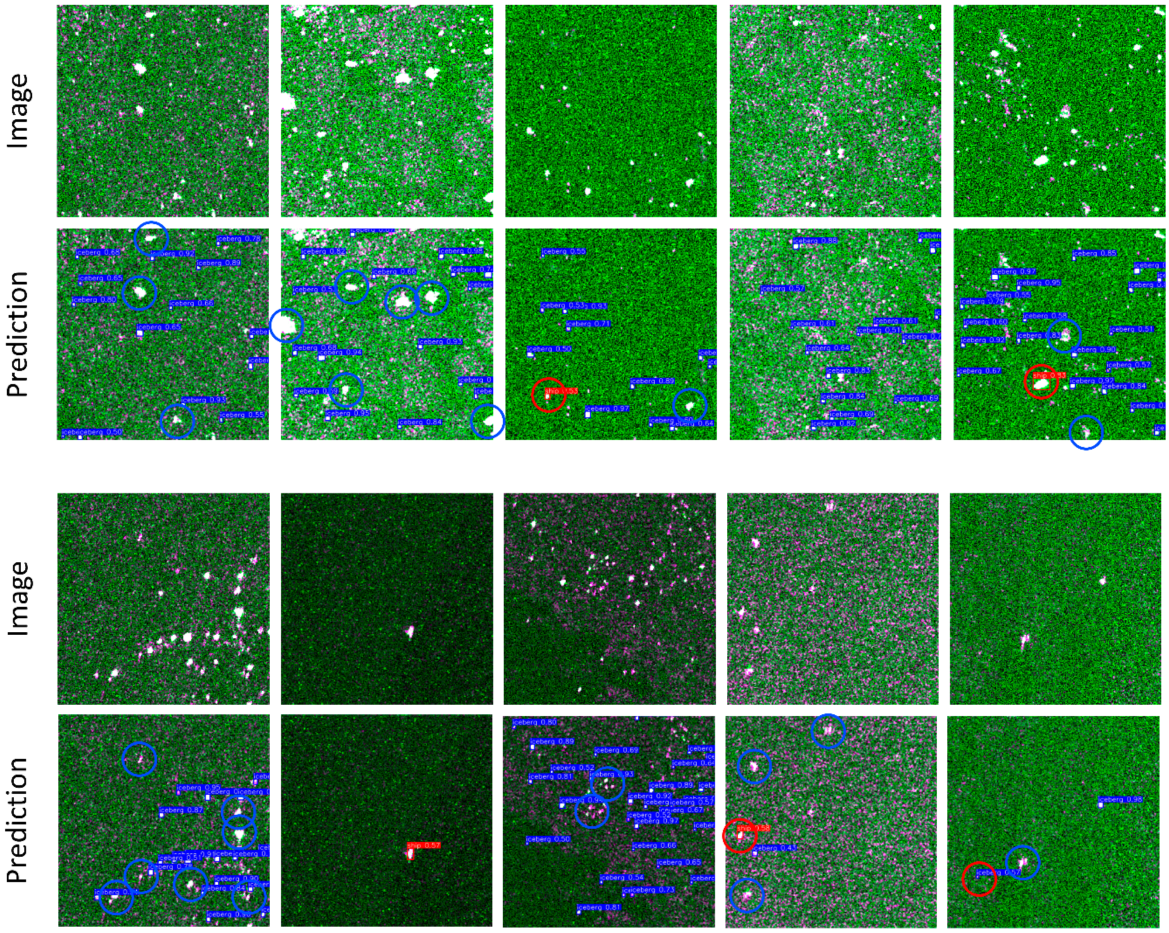

Figure 3. Input Sentinel-1 RGB composite images and output predictions, blue labels are predicted icebergs and red labels are predicted ships. Blue circles are false negatives and red circles are false positives.

The prediction shown in Figure 4 proves that the model is indeed capable of detecting ships, with only a few ships going by undetected. The detection setting here is simpler though, with the biggest difference being the number of objects and their proximity to each other.

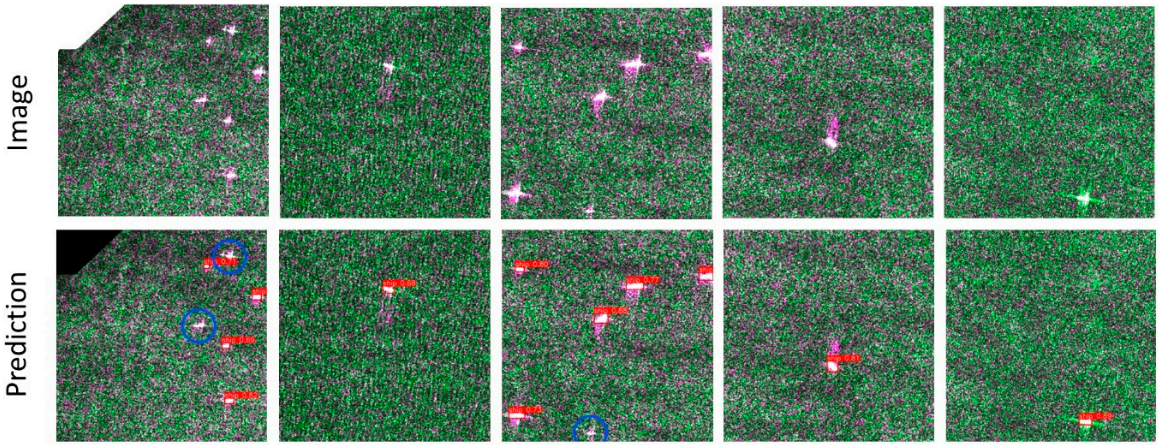

Figure 4. Input Sentinel-1 RGB composite images and output predictions, red labels are predicted ships. Blue circles are false negatives.

3.2. Testing the Model against Existing Iceberg Detections

To test the model in a full-scale setting, we carried out a prediction for a full Sentinel-1 scene covering the Disko Bay in Western Greenland. The predictions made here are compared to iceberg detection obtained from the Danish Meteorological Institute (DMI) on the same Sentinel-1 data. The prediction is carried out with the confidence set to 0.5, meaning that the model will only return objects that have a 50% certainty of being either an iceberg or a ship. The date selected for validation was April 25, 2020.

The icebergs used for validation are detections made by DMI for the Copernicus *Sea IceBerg Concentration* product (Copernicus Marine Service. Sentinel-1 Sea Ice Berg Concentration), these data are 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 DMI's PolarPortal (DMI Polar Portal, Isberge).

The ship AIS data for the Greenland study areas have been provided by the Danish company Gatehouse, these data are not publicly available. As the data are satellite AIS, it has a lower accuracy than shore-based AIS systems. The AIS data are sorted according to the timestamp of the Sentinel-1 acquisition, this does not guarantee an exact match but only points close to detected objects.

In the Sentinel-1 scene of April 25, 2020, a total number of 23,576 icebergs and 207 ships were detected.

In Figure 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 colour and strong reflective objects (icebergs and ships) appear as white with some purple reflection as well. The vast amount of purple seen in the 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 took place in such areas. The green dominant area at the bottom of the image, marked by the red square, is open water with a large number of objects, which was used for this validation.

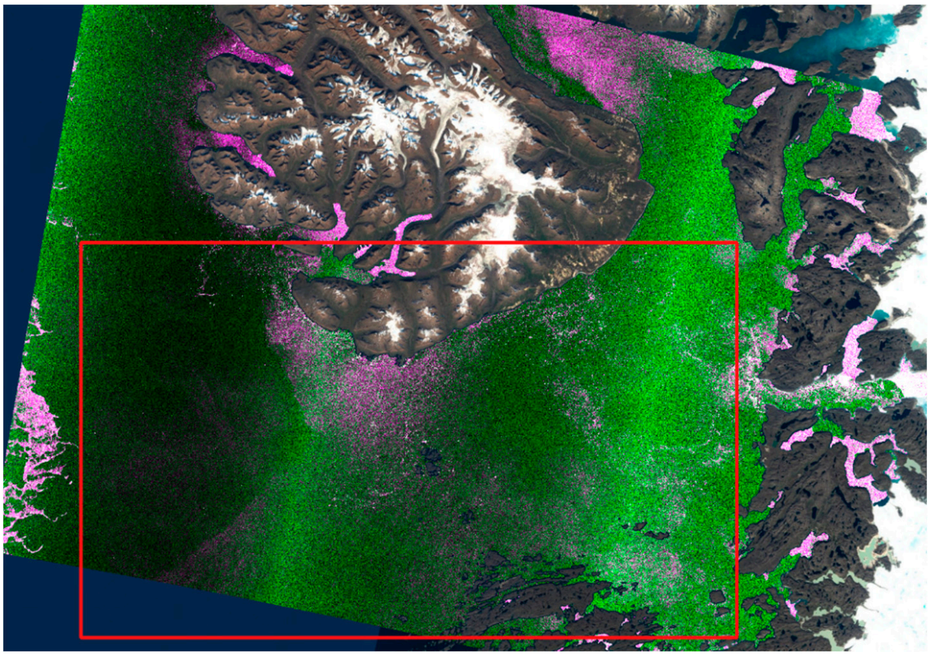

Figure 5. 18 April 2020, Disko Bay Sentinel-1 RGB composite cropped with a coastline and overlaid on a Sentinel-2 image mosaic.

In Figure 6, the detection output for the validation area is seen, detected icebergs are represented with a blue colour 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 points are AIS points with ship positions. Figure 7 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 day before shows that the large area without DMI detections is classified as sea ice (DMI ice chart, 24 April 2020).

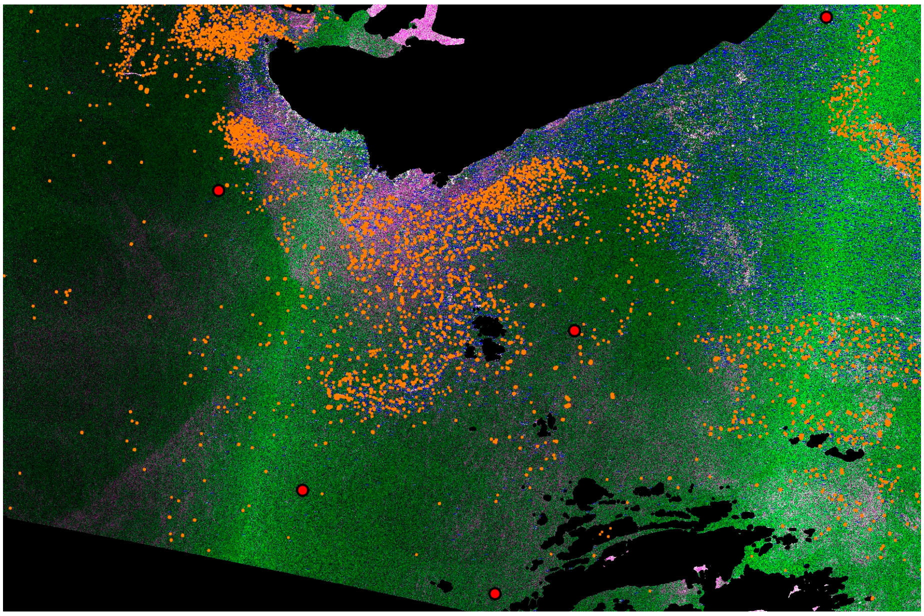

Figure 6. 20 April 2020. Validation area detection output, orange polygons and red dots are validation objects. Blue marks are iceberg detections, and red marks are ship detection (too few to see in the figure).

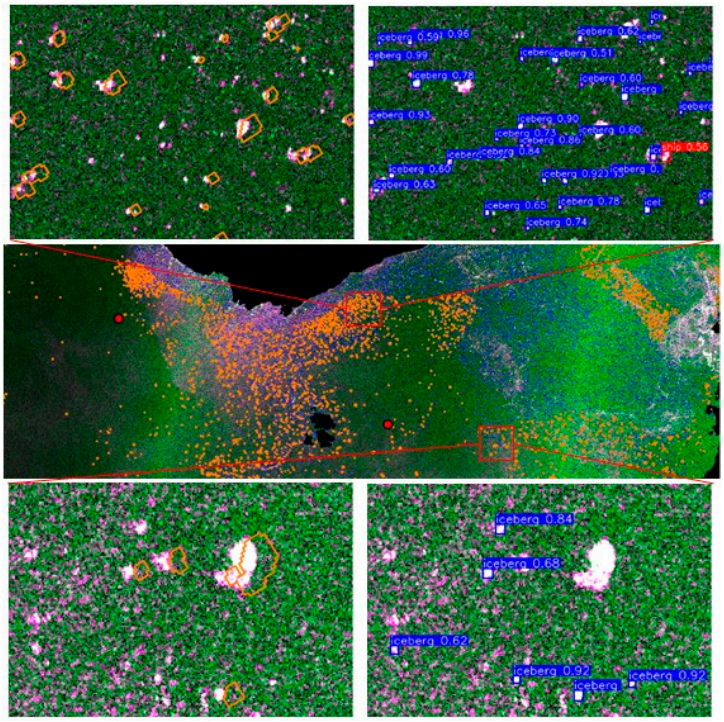

Figure 7. 20 April 2020. Close-up detection examples, left images show validation iceberg polygons from DMI and right images show model detections. Blue marks are iceberg detections, and red marks are ship detections.

In the validation area, DMI detected a total of 4601 icebergs, these are the orange polygons in Figure 6. The polygons appear to have a small offset towards the left, this is due to differences in Sentinel-1 pre-processing. With the use of DMI iceberg polygons, the detections made by the study are validated against these. The polygons are used as ground truth and the model is validated by measuring how many of these were 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 shown in Table 3. The validation shows that out of the 2340 icebergs detected by DMI, 54.95% of them were detected by the model. A fraction of these were correctly detected but classified wrongly as ships.

Table 3. 18 April 2020. Validation overview.

DMI Icebergs	Detected as Iceberg	Detected as Ship	Not deteCted
4601	2285	69	2247

This gives an overall accuracy of 51.16%, which corresponds very well to the models predicted accuracy of 55.7% (see mAP, Table 2). In the validation area, there were five ships present at the satellite acquisition time, the AIS data points from these are shown as the red dots in Figure 8. Out of these five ships, three were detected as icebergs and one was correctly detected as a ship. With the result from the iceberg validation along with the fact that 69 ships were detected even though only five were present, this validation indicates the model is to a large extent capable of detecting and classifying objects, but still struggles to detect and correctly classify the ships. The prediction results for the five ships are shown in Figure 8.

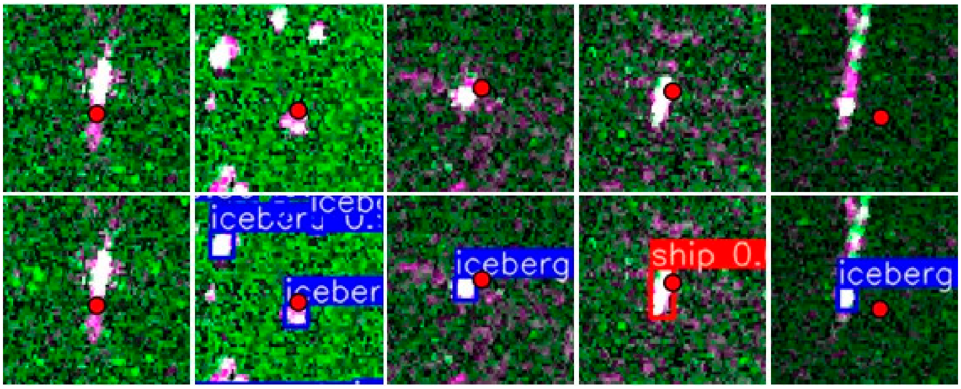

Figure 8. The 5 AIS validation data points in the validation area, top images are the input Sentinel-1 RGB composites and bottom images are the model prediction.

3.3. Results Summary

Based on the icebergs detected by DMI and ship AIS data, the validation shows an overall iceberg detection accuracy of about 51%. The ship detection carried out in Denmark had about 70% accuracy, and too few predictions were made in Greenland to estimate an actual accuracy, but the results indicate an accuracy lower than 50%. It should be noted though, that out the 23,000 objects predicted by the model, only a very few of these were ships. This indicates that even though the model has issues with the ship class, the vast majority of objects were still classified correctly.

4. Discussion

We set out to implement the Yolo detection algorithm for iceberg–ship discrimination, a difficult classification task performed in complex environments. In the following discussion, we will cover the biggest issues facing the modelling process: availability, quality and quantity of the input data for the model.

While some SAR datasets exist on ships, there is not a sufficient large-scale data set on icebergs. The only thing that comes close is the dataset provided for the 2017 Kaggle competition (Statoil/C-CORE Iceberg Classifier Challenge) on the topic in question, but the data provided are in very small image tiles and are not under a complex situation, such as the one presented in this study. Given the lack of completeness in current automatic detection methods, a great amount of manual work must be spent on creating an iceberg dataset. However, labelling icebergs in SAR data around Greenland is a complicated process, and a task that seems to remain unsolved by the Earth observation scientific community. We therefore faced the task of creating an iceberg dataset through a mix of automatic detection and manual labelling. This left the difficult question of how to label icebergs up to our interpretation. While there are definitions and categorizations of what exactly an iceberg is and different types of icebergs (such as the definition by the National Oceanic and Atmospheric Administration, NOAA), these are not used in Greenlandic iceberg charting by Copernicus and DMI (the polygons used in Section 3.2).

Figure 9 shows an example of the real-world situation at the glacier outflows. The picture highlights the complexity of labelling icebergs in the Greenlandic fjord. Some icebergs are clearly seen, but most of the ice are smaller pieces and patches of drift ice, which are difficult to categorize exactly. When looking at such scenes from satellite radar, the complexity in labelling remains the same.

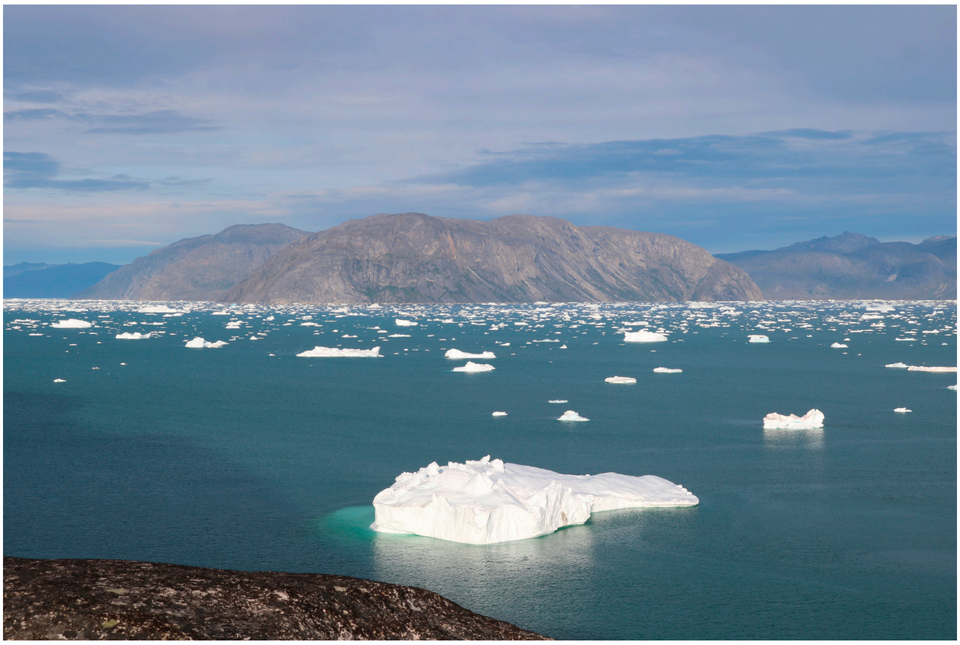

Figure 9. Image from a camera station placed in the bottom of the fjord Nuup Kangerlua. Photo taken on 13 August 2018.

Figure 10 shows an example of iceberg training data used in the model and is a good indicator of a complex situation where data are to be labelled. It could be argued that too many objects are not labelled, leaving them out of training, but in opposition to this, one could say that too many small objects are included, and these are not of great importance. In Figure 11, 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, such as in this study [23], or if such large pieces of floating ice should be in a class of their own or maybe not be included at all.

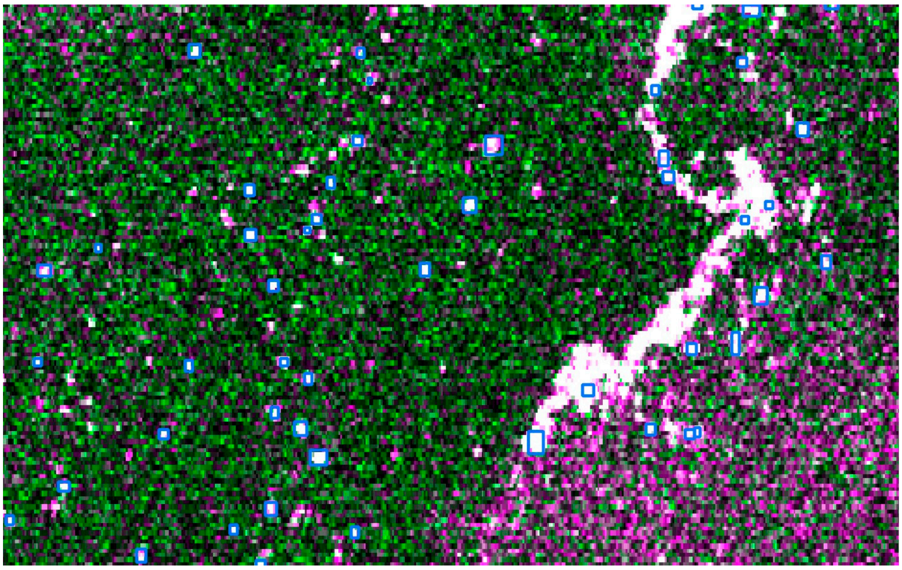

Figure 10. Example of icebergs detected with an adaptive thresholding algorithm in Sentinel-1 data.

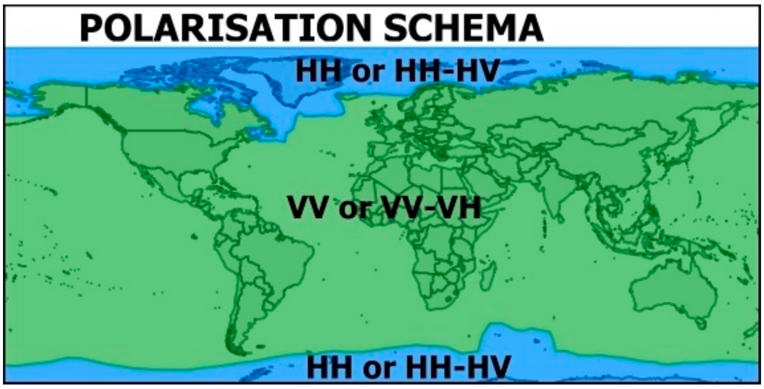

Figure 11. Sentinel-1 acquisition polarisation schema. (Source: European Space Agency "Sentinel High Level Operations Plan (HLOP)").

Since ships are clearly defined objects, populating a SAR dataset with these does not face the same issues as the icebergs. The quantity of ships sailing in Arctic and iceberg-infested waters are however very low, making it challenging to create a comprehensive data set.

Only a few dozen ships are usually sailing in Greenlandic waters at a time and given the very large geographical extent of these waters, the traffic of any given area is very sparse. This poses a challenge in validating any given model, but even more so in creating model training data. To avoid acquiring AIS data over a very long timeframe and processing equal large amounts of Sentinel-1 data, it was decided to populate the dataset with ships from more busy waters, and hence the need for the Danish study area (see Figure 1). Denmark and Greenland are, however, covered by different Sentinel-1 polarizations (see Figure 11), raising the question of the impact of training and detecting in different polarizations.

Due to the nature of the geographic distribution of the two object classes and the two polarization types, locating areas with great quantities of both ships and icebergs is a major challenge. Given that each of the object classes are by far most abundant in their separate polarization regions, it was inevitable to create training data in two different polarizations. As shown in Figure 11, most of the world is covered with the same polarization, so this is not an issue for the majority of the studies regarding object detection in SAR data, and likely the reason why it is not very well covered in the literature. Thus, 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 is not trained on ships appearances in the HH+HV polarization. To which degree, and exactly what effect the cross-polarization training and detection have on overall accuracy, are certainly subjects for further investigation.

5. Conclusions

In this paper, we proposed implementing the YoloV3 object detection algorithm for Sentinel-1 iceberg and ship detection in Arctic waters, a long-lasting issue in remote sensing of the arctic regions. Our study shows the capabilities of the state-of-the-art deep learning framework, while also highlighting the issues facing implementation of such models.

The choice of the Yolo framework was based on documented performance, training time and inference speed. With the model showing such good performance with a bare amount of training data, we confirm the choice of Yolo for this purpose. At the time of this study, YoloV3 was state-of-the-art and we chose this detector based on documented performance under various settings, new improvements have since arrived and we encourage future research to implement the later versions such as the YoloV4 or V5 model. While we still believe that good results can be achieved with other single-stage detectors, such as the SSD, the purpose of this study has not been to compare deep learning frameworks, but to highlight the difficulties in implementation and validation.

Due to the lack of existing quality data, we set out to create our own data set for the purpose of the project. The data set created is, in a deep learning context, still at a relatively small size. However, testing the model under very difficult circumstances and complex backgrounds still yielded good detection capabilities, paving the way for future work.

In this specific case, the capabilities of any object detection framework are far beyond the quality and quantity of existing data sets, stating that the creation of training data is currently of greater importance than comparing model frameworks. The cross-polarization scenario is a challenging large-scale annotation of ship data, with only a few ships sailing in Arctic waters, while setting up specific goals for annotating icebergs is a necessity as well. Future research should keep implementing state-of-the-art algorithms, but our conclusion remains that real improvements to end results come from continuous work in annotating large-scale data sets for the research community to use.

Supplementary Materials: The respective developed application is accessible at GitHub-Frhass: https://github.com/frhass/yolo_dataprep.

Author Contributions: Conceptualization, Jamal Jokar Arsanjani; formal analysis, Frederik Seerup Hass; methodology, Frederik Seerup Hass; supervision, Jamal Jokar Arsanjani; validation, Frederik Seerup Hass; writing—original draft, Frederik Seerup Hass; writing—review and editing, Frederik Seerup Hass and Jamal Jokar Arsanjani. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Le Traon, P.Y.; Reppucci, A.; Alvarez Fanjul, E.; Aouf, L.; Behrens, A.; Belmonte, M.; Benkiran, M. From observation to information and users: The Copernicus Marine Service perspective. *Front. Mar. Sci.* **2019**, *6*, 234. [CrossRef]
2. Tonboe, R.T.; Eastwood, S.; Lavergne, T.; Sørensen, A.M.; Rathmann, N.; Dybkjær, G.; Kern, S. The EUMETSAT sea ice concentration climate data record. *Cryosphere* **2016**, *10*, 2275–2290. [CrossRef]
3. Ouchi, K. Current Status on Vessel Detection and Classification by Synthetic Aperture Radar for Maritime Security and Safety. In Proceedings of the 38th Symposium on Remote Sensing for Environmental Sciences, Gamagori, Aichi, Japan, 3–5 September 2016.
4. Danish Ministry of Defence. *Forsvarsministeriets Fremtidige Opgaveløsning i Arktis*; Danish Ministry of Defence: Copenhagen, Denmark, 2016.

5. Santamaria, C.; Greidanus, H.; Fournier, M.; Eriksen, T.; Vespe, M.; Alvarez, M.; Argentieri, P. Sentinel-1 Contribution to Monitoring Maritime Activity in the Arctic. In Proceedings of the ESA Living Planet Symposium, Prague, Czech Republic, 9–13 May 2016.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 91–99. [CrossRef] [PubMed]
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
8. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
9. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv* **2016**, arXiv:1602.07261.
10. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. A SAR dataset of ship detection for deep learning under complex backgrounds. *Remote Sens.* **2019**, *11*, 765. [CrossRef]
11. Dechesne, C.; Lefèvre, S.; Vadaine, R.; Hajduch, G.; Fablet, R. Ship identification and characterization in Sentinel-1 SAR images with multi-task deep learning. *Remote Sens.* **2019**, *11*, 2997. [CrossRef]
12. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car Detection Using Unmanned Aerial Vehicles: Comparison between Faster R-Cnn and Yolov3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; pp. 1–6.
13. Chang, Y.L.; Anagaw, A.; Chang, L.; Wang, Y.C.; Hsiao, C.Y.; Lee, W.H. Ship detection based on YOLOv2 for SAR imagery. *Remote Sens.* **2019**, *11*, 786. [CrossRef]
14. Van Etten, A. Satellite Imagery Multiscale Rapid Detection with Windowed Networks. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 735–743.
15. Pelich, R.; Chini, M.; Hostache, R.; Matgen, P.; Lopez-Martinez, C.; Nuevo, M.; Eiden, G. Large-Scale Automatic Vessel Monitoring Based on Dual-Polarization Sentinel-1 and AIS Data. *Remote Sens.* **2019**, *11*, 1078. [CrossRef]
16. Choi, D.; Shallue, C.J.; Nado, Z.; Lee, J.; Maddison, C.J.; Dahl, G.E. On empirical comparisons of optimizers for deep learning. *arXiv* **2019**, arXiv:1910.05446.
17. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
18. Taqi, A.M.; Awad, A.; Al-Azzo, F.; Milanova, M. The impact of multi-optimizers and data augmentation on TensorFlow convolutional neural network performance. In Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 10–12 April 2018; pp. 140–145.
19. Olson, M.; Wyner, A.; Berk, R. Modern neural networks generalize on small data sets. In Proceedings of the Advances in Neural Information, Neural Information Processing Systems 31 (NeurIPS 2018), Montreal, BC, Canada, 3–8 December 2018; pp. 3619–3628.
20. Ammar, A.; Koubaa, A.; Ahmed, M.; Saad, A. Aerial images processing for car detection using convolutional neural networks: Comparison between faster r-cnn and yolov3. *arXiv* **2019**, arXiv:1910.07234.
21. Manning, C.D.; Schütze, H.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
22. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
23. Soldal, I.H.; Dierking, W.; Korosov, A.; Marino, A. Automatic Detection of Small Icebergs in Fast Ice Using Satellite Wide-Swath SAR Images. *Remote Sens.* **2019**, *11*, 806. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.