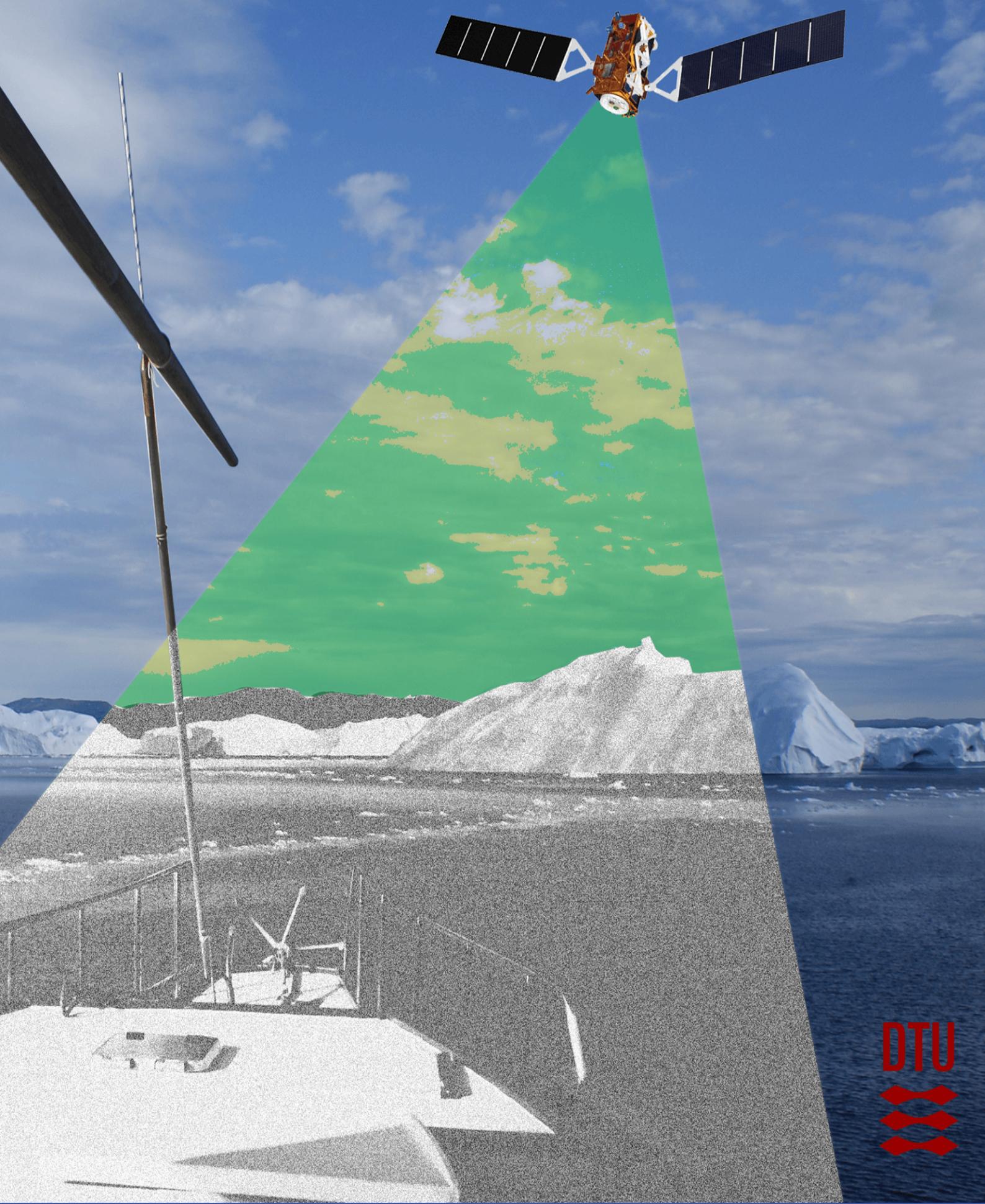
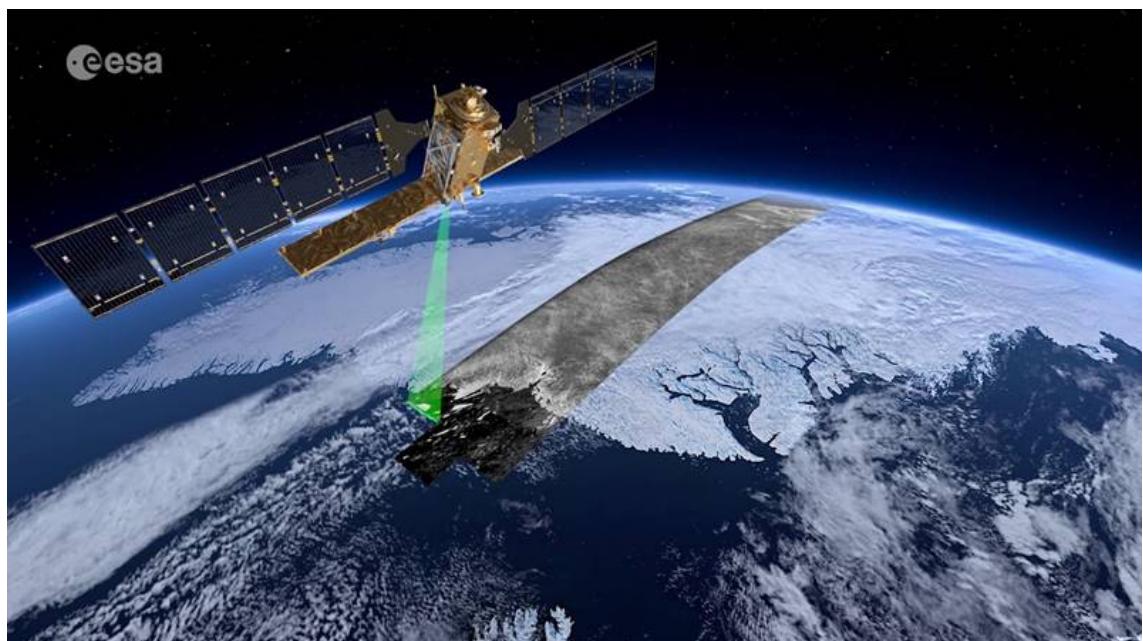


Ship-Iceberg classification in satellite images using deep neural nets

Master Thesis by Danjal Longfors Berg





Ship-Iceberg classification in satellite images using deep neural nets

Master Thesis
January, 2021

By
Danjal Longfors Berg

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.

Cover photo: Lasse Klint, 2021

Published by: DTU , Akademivej, Building 356, 2800 Kgs. Lyngby Denmark

Approval

This thesis has been prepared over five months at the Section for Center for Security DTU, Department of Earth and Space Physics and Engineering, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of statistics.

Danjal Longfors Berg - s143277

.....
Signature

.....
Date

Abstract

With the retraction of sea ice in the Arctic, the northeast and northwest passages become more available to ship traffic. Real-time or close to real-time surveillance is desired to locate ships that are involved in illegal activities. Such ships are likely to turn off their Automatic Identification System (AIS) transponder and are referred to as dark ships. These ships can be located in a Synthetic Aperture Radar (SAR) images. In the Arctic, icebergs can have a similar signature as ships in a SAR image.

In this master thesis, the discrimination and classification between ships and icebergs in SAR satellite images have been investigated. A public available Kaggle dataset of ships and icebergs has been analysed. Multiple Convolutional Neural Networks (CNNs) have been used, such as a modified Unet, VGG, Resnet and Densenet. The best networks were combined to an ensemble network. Additional parameters such as the estimated target pixel area and incidence angle were included in the network. A custom dataset of SAR ships and icebergs from Greenland was created to test if the networks trained on the public available dataset could predict other data. The preprocessing steps used to generate similar images of ships and icebergs were analysed. It was shown that the public dataset was not calibrated for incidence angle and Sentinel 1 subswath thermal noise well enough to eliminate the noise in the SAR HV polarisation band. Of the 14 different CNNs architectures trained, no individual network was superior. The 8 best models had similar test loss on the Kaggle dataset. The best network could predict ships from the custom prepared dataset with an accuracy of 95% but icebergs with an accuracy of only 71%. A more extensive custom dataset should be created so better control over the preprocessing steps can be accomplished.

Acknowledgements

Danjal Longfors Berg, MSc Earth and Space Physics and Engineering, DTU
The Creator of this thesis. *Veni, vidi, vici*

Henning Heiselberg, Head of Center, Security DTU.

A special thanks to my main supervisor Henning Heiselberg who suggested- and has shown great interest in the project. Your dedication and eagerness to learn new and adapt to new methods in image analysis and deep learning are inspiring.

Ole Baltazar Andersen, Professor, Geodesy and Earth observation.

Thanks to Ole Baltazar Andersen my other supervisor who has shown immense enthusiasm upon my work process and results.

Sune Hørlück, Specialist, Innovation Lab, Smart Products, Technology & Innovation, Terma A/S.

Thanks to Sune Hørlück who could give a unique perspective on the applications and methods applied in this master thesis.

Kristian Sørensen, MSc Earth and Space Physics and Engineering, DTU.

One of my best decisions was to change student office to where you were. It has been a blast of a ride to make this thesis in your presence. We have helped each other through hard times and when not everything went as planned. I hope we will share office again at some point in the future.

Contents

Preface	ii
Abstract	iii
Acknowledgements	iv
1 Introduction	1
2 Deep Learning Theory	3
2.1 Simple neural network	3
2.2 Convolutional Neural Network	7
2.3 Regularisation	10
2.4 Optimizer	12
2.5 Known CNN Models	12
3 Synthetic Aperture Radar Theory	15
3.1 SAR Background	15
3.2 Sentinel-1 Specification	18
3.3 Scattering model of the ocean	21
3.4 SAR detection of moving targets	22
3.5 Backscatter Normalisation	22
4 Dataset of ships and icebergs	23
4.1 Prelabelled data	23
4.2 Data prepared in this study	23
5 Method	25
5.1 Prepare backscatter values for Neural Network	25
5.2 Preparing Sentinel-1 Scenes	30
5.3 Sentinel-1 GRD preprocessing	30
5.4 Land Masking	32
5.5 Constant False Alarm Rate	34
5.6 Automatic Identification System	35
5.7 Generating Ship Iceberg Images	36
6 Results	39
6.1 Backscatter dependence on incidence angle	39
6.2 Transfer Models	45
6.3 Custom data	51
6.4 Pseudo labelling and Size	52
6.5 Extended model and custom data prediction	53
7 Discussion	59
7.1 Preprocessing steps	59
7.2 Transfer models	60
7.3 Custom data	62
7.4 Pseudo labelled data and size	62
7.5 Extended model	66
7.6 Comparison to the Kaggle Competition	67

8 Future Work	69
9 Conclusion	71
Bibliography	73
A Modified Unet	75

1 Introduction

In this master thesis, ship and iceberg discrimination were investigated. In the Arctic region ships are tracked by satellite with Automatic Identification System (AIS). AIS monitors ships position and other information about the ships. Ships that were supposed to send AIS information but for whatever reason does not, are called dark ships. Dark ships can be found in Synthetic Aperture Radar (SAR) satellite images.

In the Arctic region there is an additional obstacle when locating ships in SAR images. Icebergs and ships can look very similar. This master thesis investigates a subcategory in machine learning called deep learning for classifying and distinguish between ships and icebergs.

Ship iceberg classification in SAR satellite images have been attempted in the past. In recent years many commercial companies and space agencies have launched SAR satellites with increased temporal and spatial resolution. With technology improvements in both SAR satellites and in deep learning, the Arctic ocean could be monitored in real time or close to real time.

A problem in supervised deep learning is in order to train a Neural Network the dataset has to be prelabelled. Labeling data are very time consuming. A prelabelled dataset was used from Statoil/C-CORE which was used in a Kaggle competition. This dataset has no information about how the images were generated and preprocessed. This dataset was extended with custom data prepared in this study. The preprocessing steps of European Space Agency (ESA) Sentinel-1 SAR images were investigated to obtain the best possible result. The custom data prepared in this study was predicted with the Statoil/C-CORE trained network. The custom data was further used to extend the original training set from Kaggle.

With control over the processing chain and with a good classification between icebergs and ships, monitoring of the Arctic oceans could be used for many different purposes. Such purposes are:

- Illegal fishing
- climate monitoring
- search and rescue
- Illegal trespassing, upkeep and maintain sovereignty over waters

The Arctic ocean gets more accessible for marine traffic due to retraction of sea ice. The focus for authorities to maintain law and order increases with more traffic. Ideally it is desirable to have real-time or close to real-time surveillance over the Arctic ocean. With control over the data preprocessing steps and a good neural network for classification can this be archived.

2 Deep Learning Theory

Neural network is a subcategory within machine learning. Neural Networks are not a new concept but due to computational limitations Neural networks have not been feasible to work with. Until 2006 neural networks could not surpass more traditional machine learning methods, [Nielsen 2019]. Neural networks can give accurate predictions for problems in computer vision, time series analysis, speech recognition, natural language processing and many other areas.

2.1 Simple neural network

A fundamental part of many neural networks are feed forward neural networks or multi-layer perceptrons (MLP). The goal for the network in the case of a classification problem is to map an input x to a category y . The mapping of x to y can be defined as:

$$y = f(x; \theta) \quad (2.1)$$

Where x is an input image in the case of image analysis, as is assumed throughout this section. θ are a range of parameters the network learns to find the optimal result that gives the best approximation to the function (f). The input image (x) is provided to the feed forward neural network and is reshaped into a vector. For a classification problem the output layer is given as class probabilities. The output y have the same dimensions as number of k classes and is given as a vector. The probabilities are in the range of $[0, 1]$ after the output \hat{y} is normalised and the image is assigned to the class with highest probability. Any intermediate steps between input and output are preformed in the hidden layers. Hidden layers means that it is neither a input or an output layer.

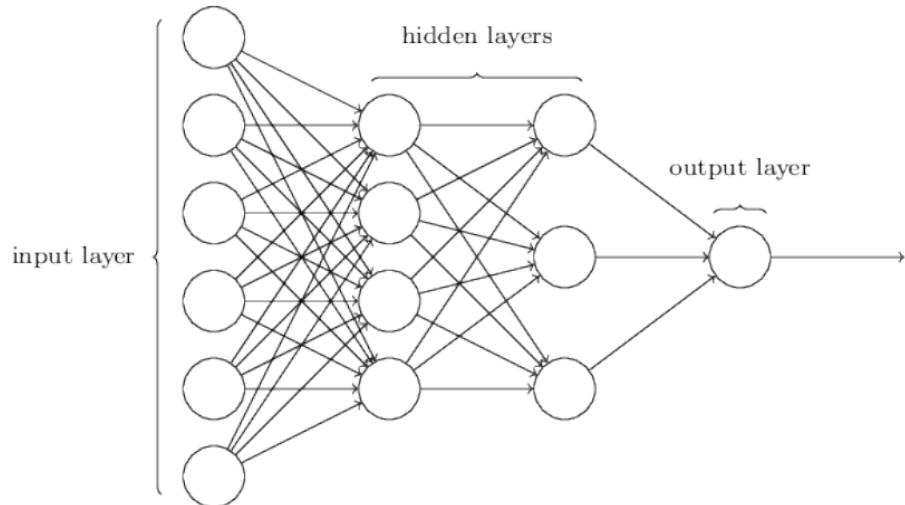


Figure 2.1: Image taken from Nielsen 2019. A simple fully connected feed forward neural network with an input layer, 2 hidden layers and an output layer.

2.1.1 Forward model

The input layer x is given as $[x_1, x_2, \dots, x_n]$. As an additional node b is introduced. b is characterised as a bias node and will help the network to better fit the data. The input layer is multiplied with weights w_i as is shown as arrows in figure 2.1. The summation to each node at the hidden layers is given as z_i and defined as:

$$z_i = \sum_{d=0}^D w_{id}^{(1)} x_d + b \quad (2.2)$$

The input to a node z_i in the hidden layer is followed by an activation function a . Many different activation functions exists but the most common is called rectified linear units function (ReLU) and is given as:

$$h_i = a(z_i) = \max\{0, z_i\} \quad (2.3)$$

Each hidden layer has a bias node b as well. The bias node for the hidden layers are not connected with previous layers. The same summation with activation functions will occur throughout the network to the last hidden layer:

$$h_i = a\left(\sum_{d=0}^D w_{id}^{(n)} h_d + b\right) \quad (2.4)$$

The output layer is calculated in the same way as the hidden layers:

$$\hat{y} = \sum_{m=0}^M w_{jm}^{(N)} h_m + b \quad (2.5)$$

The output should be seen as probabilities and therefore the output \hat{y} is transformed by a softmax function for a multiclass classification problem:

$$y_i = \frac{\exp \hat{y}_i}{\sum_{k=1}^K \exp \hat{y}_k} \quad (2.6)$$

2.1.2 Backpropagation, Gradient Decent and Batches

The objective is to adjust the weights in the network to better predict the correct target. In the beginning the weights are pseudo random initialised where the weights are given an uniform distribution in Pytorch¹ with a standard deviation dependent on the number of weights. The bias in the input and hidden layers are set to zero. A random initialised network can predict correct in $\frac{100\%}{K}$ of the cases where K are the number of classes. To improve the classification the weights are optimised with stochastic gradient decent. The optimisation is done by minimising a loss function. A typical loss function for multiclass classification are the cross entropy loss function:

$$L = - \sum_{k=1}^K t_k \log y_k \quad (2.7)$$

¹<https://pytorch.org/>. Pytorch is a deep learning library for Python that is efficient and automates many steps required for deep learning.

Where t_k is the target value given as:

$$t_k = \begin{cases} 1 & \text{if class label is } k \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

The objective is to update the weights (and biases) such that the network minimises the given loss or cost function. In the forward propagation of the network an image is given a probability for each class. In backpropagation the weights (and biases) updates from the output layer and updates backwards in the network. All weights to a neuron are updated in such a way that all nodes that have an influence to the target are increased proportionally. Next time the image or a similar image is present the network is more likely to predict correct. To optimize the weights, its derivatives are taken with respect to the loss function. The weights and biases are updated by:

$$w_{ij}^{(l)new} = w_{ij}^{(l)} - \eta \frac{\partial L}{\partial w_{ij}^{(l)}} \quad (2.9)$$

η is called the learning rate and defines how much the weights should be updated for each iteration.

By the chain rule the derivative can be split up:

$$\frac{\partial L}{\partial w_{ij}^{(l)}} = \frac{\partial L}{\partial z_{ij}^{(l)}} \frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}} \quad (2.10)$$

[Nielsen 2019, p. 48] shows how this can be further derived. An important thing to note in equation (2.10) is that the derivatives for all layers are connected.

Stochastic gradient decent takes a batch of the training data to approximate the correct step that minimises the loss function. More than a 1000 images are often required for each class as training data for the network to be able to generalise and properly discriminate between different categories. It is computational expensive to use all available training images in each iteration where the weights and biases are updated by backpropagation. A small batch of the images are instead used to approximate the correct direction in the multidimensional space that minimises the loss. The batch size for stochastic gradient decent are often in the range [8, 128]. The higher the batch size the better approximation to the gradient decent but the trade-off is that the computational cost is greater.

In general is there a relation between the learning rate η and batch size. For a greater batch size the learning rate can be increased as well due to a better approximation of the stochastic gradient decent. The size of the learning rate η depends on the problem but are normally in the range $[1 \times 10^{-4}, 1 \times 10^{-1}]$.

Tuneable parameters as the learning rate and batch size are referred to as hyperparameters. Other such parameters can be the number of neurons in a layer. Hyperparameters are user defined.

The loss and hence the neural network converges to a minimum after a given number of iterations. The network will converge to a local minimum but ideally a global minimum. The global minimum indicates the optimal solution that the networks architecture can achieve. There are many local minima the loss function can converge to and therefore the model should be trained multiple times to find the best model that fits the data.

2.1.3 Binary Cross Entropy Loss

In the case where only two classes exist, as for Ship / Iceberg classification, binary cross entropy loss are used and it is defined as:

$$L = \frac{1}{N} \sum_i y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \quad (2.11)$$

The change in the loss function also makes a change to the activation function in the last layer $a(\hat{y}_i) = y_i$ to a sigmoid function instead of softmax given as:

$$y_i = \frac{1}{1 + \exp(-z)} \quad (2.12)$$

2.1.4 Limitation of Feed Forward Neural Network

In the data presented later for ship / iceberg detection the images are of size 75×75 pixels with two bands but lets assume in this section that only one of these bands and see it as a grayscale image. A similar architecture as figure 2.1 is used with two hidden layers where the first hidden layer has 1024 neurons and the second hidden layer has 512 hidden neurons. The output layer should classify the image as either ship or iceberg and therefore the output layer has 2 neurons. In this simple example $75 \times 75 = 5625$ neurons are in the input layer connected to the first hidden layer. When all these neurons are connected there will be a total of $5625 \times 1024 + 1024 \times 512 + 512 \times 2 + 3 = 6,285,315$ weights to update through backpropagation where 3 are the biases from the input and 2 hidden layers. As it is shown with this very small simple neural network the number of weights will quickly explode out of proportion and therefore take longer time to train.

2.1.5 Vanishing- / Exploding Gradients

It could be argued that instead of using a few hidden layers with many neurons could the number of weights to train be limited by adding more hidden layers but with fewer neurons. This gives rise to an other problem in deep learning called Vanishing- / Exploding Gradients. Vanishing gradients occur due to how the backpropagation is calculated and the weights are updated. When the derivatives with respect to the loss function is taken it will, by the chain rule, be connected to all previous layers. With the chain rule the derivatives of all layers are multiplied with each other, so many small gradients less than one can be multiplied together.

Without stating it explicitly two things has already been added to the network to limit the influence of the vanishing / exploding gradient problem.

- Kaiming He weight initialisation, [K. He et al. 2015]
- Use ReLU as activation function

Kaiming weight initialisation ensures that the variance of initialised layers is dependent on the size of the input to a fully connected layer l . This ensures that the variance of the summation $z_i = \sum_{d=0}^D w_{id}^{(l)} x_d + b$ is $\text{Var}(w_i) = \frac{1}{n}$.

Previously it was common to use sigmoid as activation function before ReLU became the standard. The derivative used in backpropagation for the sigmoid is small and therefore squeezes the gradients. This is a problem in deep neural nets because it leads to vanishing gradients the further back in the network. With vanishing gradients the weights are not updated or only by a negligible amount and the network will therefore stagnate.

Exploding gradients are not as large a problem as vanishing gradients. Exploding gradients results in very large gradients and the network becomes unstable. Exploding gradients can be mitigated by have fewer layers in the network or use gradient clipping which limits the size of the gradients.

2.2 Convolutional Neural Network

To improve the learning capacity of the neural network Convolutional layers are introduced. A convolution is a kernel looped over an image to extract features. A Feed Forward Neural network does not use spatial structures within the image. A Feed Forward Neural network treats pixels that are close to each other the same way as pixels that are far away. Convolutional neural network takes advantage of the spatial structure in the image and can learn important structures. Examples of these structures the convolutional layers can learn are edge detection, smoothing filters, highpass or lowpass filters and many more types of filters.

2.2.1 Convolving Layers

A filter / kernel of size $N \times M$ is placed over the image and a new value is calculated for a given location in the image.

And one pixel in the convolved image is calculated by:

$$S(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (2.13)$$

Where I is the input and K is the kernel.

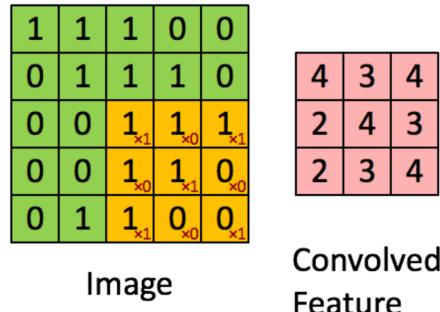


Figure 2.2: A 3×3 convolution over a 5×5 image. The convolved feature is 3×3 .

Besides the size of the kernel can change, the stride or how far the kernel moves over the image can also change. The stride is usually set to 1 in the x and y direction of the image. A stride of 1 and a kernel size of 3×3 are used on the image in figure 2.2. Here it shows that the output size is smaller than the input size. To keep the same dimensions in the convolved feature as well as the input image, the image can be padded. A usual padding for convolutions in deep learning is to add zeros around the edge of the image. The convolved feature size is given by:

$$output_{dim} = \frac{input_{dim} - filter_{dim} + 2padding}{stride} + 1 \quad (2.14)$$

For a 3×3 kernel are there 9 weights the neural network has to train. These 9 weights are shared throughout the image. This limits the number of weights the network has to

train and helps generalise structures in the image. A convolutional neural network can train many different filters to the same image. Often these are chained together such that an input image of size $H \times W$ with 16 filters can be trained on so the output is of size $H \times W \times C$. Where C is the number of channels or number of filters. After the convolutional layer have been applied to the image an activation function is used such as ReLU. This new representation of the data is then used as the input to a new convolution block of size C .

2.2.2 Fully Connected Layers

For categorical classification these convolutional layers will be input to a feed forward neural network. The new representation of the image where the network has found and highlighted important features of size $H \times W \times C$ are flattened into a vector. Each pixel is a node in the fully connected input layer.

In the previous example with grayscale images of size 75×75 would there be 5625 neurons to the input to the Feed Forward part of the neural network. With the additional convolved layers there are now $75 \times 75 \times 16 = 90,000$ neurons. These additional layers adds additional weights to train in the fully connected part of the network. To limit the input size to the Feed Forward Neural Network Pooling layers are introduced.

2.2.3 Pooling layers

A pooling layer is often used immediately after a convolutional layer. Pooling layer returns a lower level representation of the data. A widely used pooling layer is max-pooling with a size of 2×2 and a stride of 2.

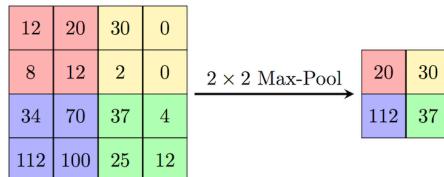


Figure 2.3: A 2×2 max-pooling layer used on a 4×4 image. The result in an 2×2 image.

As illustrated in figure 2.3 the max-pooling kernel returns the maximum value within the area over the image where the kernel is applied. A 2×2 kernel with stride 2 reduces the image size by a factor of 4. Max-pooling will highlight features within the image but not their exact relative positional information. Some information will be lost due to the pooling layer but the benefits of reducing the parameters the network needs to learn outweighs the reduced information in the image. Max-pooling is only one type of pooling other types of pooling that are used are, minimum pooling, average pooling and L2 pooling.

2.2.4 Activation functions

There are many different activation functions to choose from. Two activation functions that have already been mentioned are sigmoid and ReLU. A few activation functions can be seen in figure 2.4.

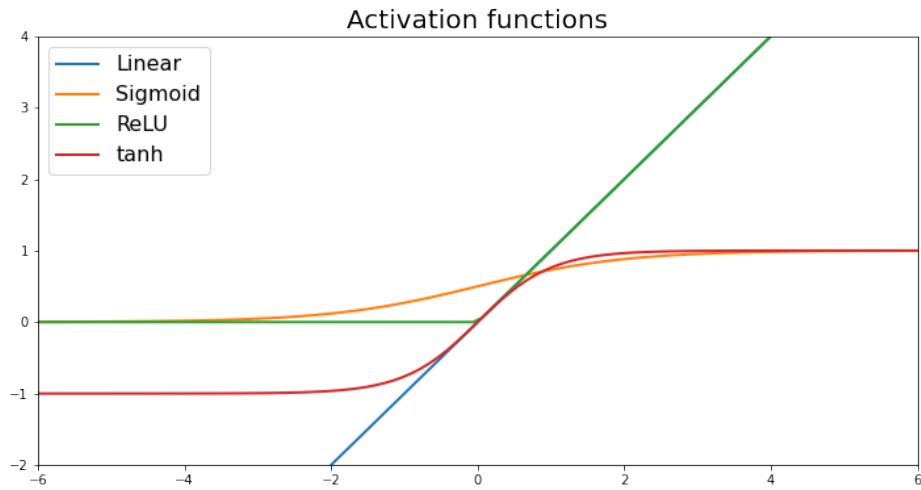


Figure 2.4: Four different activation functions. Linear, Sigmoid, ReLU and tanh.

ReLU was an improvement from sigmoid due to how it avoided some of the problems with vanishing gradients. ReLU has some special properties because the function is non-linear. Non-linear activation functions are capable of learning more complex data. Further is ReLU less computational expensive. ReLU has become the most common activation function does it have some drawbacks. In the region where $x < 0$ will the gradient be zero. This can lead to dead neurons meaning that some neurons can get stuck and never update.

To prevent dead neurons some different modifications to ReLU are introduced. Leaky Rectified Linear Unit (LReLU) adds a small negative slope to $x \leq 0$. The negative slope α is by default 1×10^{-2} .

$$R(z) = \begin{cases} z & z > 0 \\ \alpha z & z \leq 0 \end{cases} \quad (2.15)$$

A further improvement to LReLU is Parametric ReLU (PReLU). Instead of treating α in LReLU as a constant negative slope it is in PReLU a learnable parameter in the network. These activation functions are the most common but more can be found at Pytorch².

²<https://pytorch.org/docs/stable/nn.html#non-linear-activations-weighted-sum-nonlinearity>

2.2.5 Simple Convolutional Neural Network

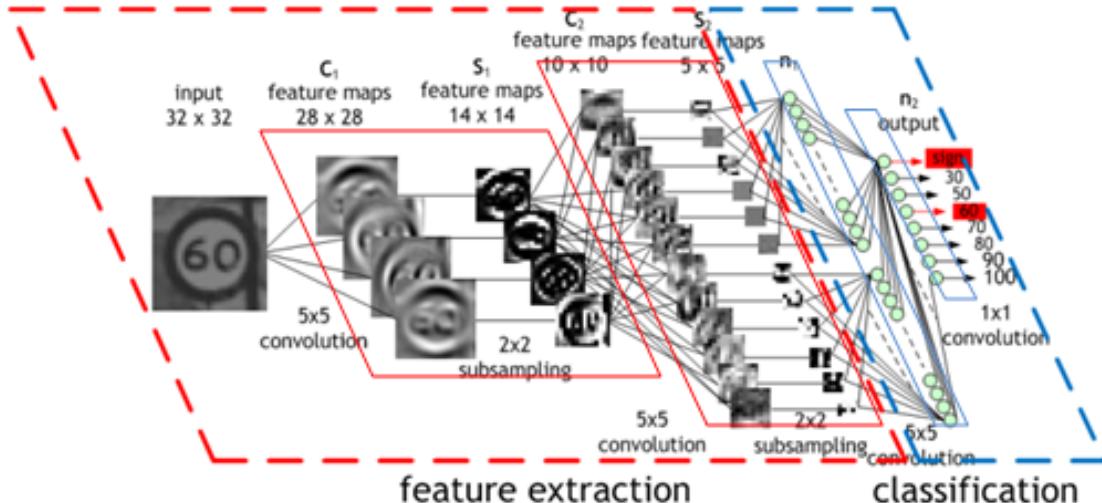


Figure 2.5: Image taken from <https://developer.nvidia.com/discover/convolutional-neural-network>. Important features are extracted from a speed sign. The convolved features are subsampled (max-pooled) to give a lower representation of the image. The speed limit is classified in the fully connected classification part of the network.

In figure 2.5 a simple convolutional neural network can be seen. The network consist of two parts, a feature extraction and a classification part. The feature extraction part have convolutional layers followed by a pooling layer or in figure 2.5 referred to as subsampling. After feature extraction the features are flattened into a vector as input to the classification part. The classification have fully connected layers and in the case of multi classification the probability is calculated by a softmax function.

Even with this relative simple network architecture there is a high degree of freedom to change different hyperparameters to find the optimal configuration. Inside the network these following areas can be tweaked:

- Number of convolutional layers, the kernel size, stride and padding.
- Type of activation functions
- Type of pooling layer
- Size of the fully connected layers

There are also different ways to regularise the network which locate the local minimum of the network faster and avoid overfitting the data so the network is more robust.

2.3 Regularisation

There are a number of ways to regularise the network. This is mainly done to prevent overfitting and to make sure the network is better to generalise.

2.3.1 L2 regularisation

L2 regularisation adds an extra term to the cost function

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \quad (2.16)$$

Where C_0 is the original cost function and could either be cross entropy, binary cross entropy or some other loss function. λ is the regularisation parameter and n is usually the size of the training set. w are the weights in the network. The L2 regularisation term have an effect on the network which makes it prefer to learn small weights to minimise the term. This creates a balance between the original cost function and the L2 regularisation term. Large weights are only possible if the original cost function is minimised more than the regularisation term. λ is a user defined value which is dependent on the data.

2.3.2 Dropout

Dropout helps regularise the weights in the network indirectly, [Srivastava et al. 2014]. Dropout turns off a given percentage of the neurons for each iteration throughout the network. The neurons selected to be turned off are picked randomly, see figure 2.6. For each iteration is a new random subset of the neurons turned off. Dropout is applied on a layer basis so the data analyst can choose where dropout should be applied. With Dropout the neural network learns to not rely on only a few neurons for a given class instead the network learns how more neurons can contribute to the classification problem. Dropout makes the network more robust and reduces overfitting. Dropout can be seen as training the network many times and do a majority voting to get the best result. When it is time to evaluate the model all dropout layers turned are off so they do not influence the model.

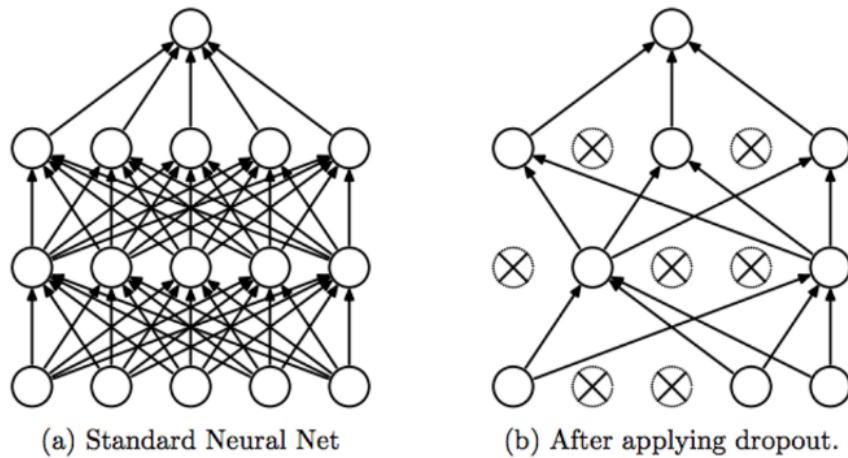


Figure 2.6: Dropout applied on fully connected layers. a) no dropout have been applied. b) dropout has been applied and some nodes are turned off.

2.3.3 Data augmentation

It is important in supervised deep learning to get a large enough dataset to train on. With more data the network can better generalise and predict correct. As a rule of thumb there should be more than at least a 1000 unique images for training each class. It can be very time consuming to find and label new data and therefore data is often augmented to enhance the dataset. It is assumed that the object is already centered in the image as a preprocessing step. For images there are some standard augmentation techniques:

- Flip image over its horizontal- / vertical axis
- resize image and chop
- Rotate image
- Other augmentations can be found at Pytorch transforms³

³<https://pytorch.org/docs/stable/torchvision/transforms.html>

The type of augmentation used requires domain knowledge or some common sense. Flip data horizontal and vertical would not be a good idea for number or letter classification. If the number 6 is both flipped horizontal and vertical it would look like a 9 and therefore be classified wrong. Other letters and numbers would not make sense if flipped. For other images like cat and dog classification horizontal flip could be a good idea but vertical would not be as probable or realistic.

Images are often resized and chopped so the center or the target in the image is shifted. This can either be done by resizing the image by interpolation or by padding the edge of the image. For letter and number classification it could be a good idea rotate the images in $\pm 20^\circ$ to artificially get more data.

All these augmentations ensures that the network learns a more general representation of the data and avoid overfitting the limited amount of training data.

2.3.4 Batch normalization

The input data to the neural network are usually normalised and standardised so the data is between $[-1, 1]$ with a unit standard deviation if the data is assumed Gaussian distributed. The network is updated by stochastic gradient decent where the direction of the step is based on a batch size. These batches have varying distributions even if the entire dataset is standardised. The change between batches results in a internal covariate shift, [Ioffe and Szegedy 2015].

$$\hat{x} = \frac{x - E[x]}{\sqrt{Var[x]}} \quad (2.17)$$

Each batch is therefore normalised to avoid this problem. Batch normalization can be applied after each convolutional layer since the convolutional layers can change the mean and variance after each operation.

2.4 Optimizer

So far the classic stochastic gradient decent has been mentioned. There are many alternatives to stochastic gradient decent. A widely used optimizer is ADAM which uses momentum and RMS prop, [Kingma and Ba 2017].

The learning rate specifies how much the weights are updated for each iteration. In the beginning of training a neural network the loss will decrease fast but slows down later in the training. The networks ability to learn new important features slows down as the loss converges to a global or local minimum. Close to the minimum big steps are not needed as in the beginning of the learning where the loss improves significant for each iteration. A learning rate scheduler can adapt the learning rate. A simple and useful scheduler is Reduce Learning Rate (LR) On Plateau (ReduceLROnPlateau). ReduceLROnPlateau updates the learning rate when the network no longer improves its loss and stagnates. The learning rate is updated by a factor (default 0.1) after a set number of epochs or update cycles called patience (default 10). If the loss of the network has not improved in the last 10 epoch the learning rate are updated $LR_{new} = LR_{old} \times 0.1$.

2.5 Known CNN Models

Many variants of convolutional neural networks have been proposed. Some often applied CNN models are, VGG [Simonyan and Zisserman 2015], ResNet, [Kaiming He et al. 2015], and DenseNet, [Huang et al. 2018] or variations of these models.

VGG is a highly influential network due to good results and it is fairly easy to implement. The network performs well with many different classification problems.

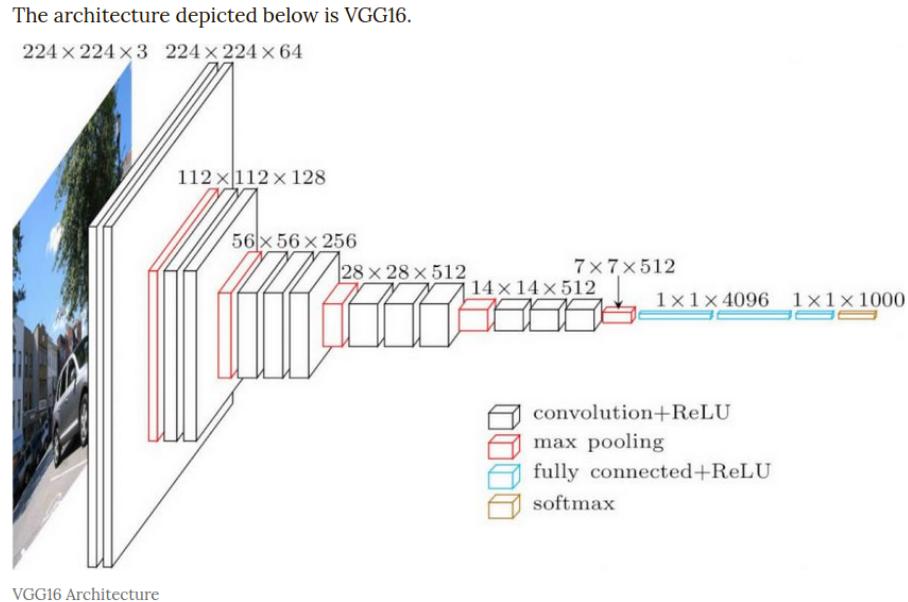


Figure 2.7: Example of a VGG architecture used to predict 1000 different classes. VGG16 has 16 weight layers (convolution and fully connected layers). VGG16 has 3 convolutional layers followed by a maxpool layer.

The VGG network is called a very deep convolutional network. It was shown that convolutional networks with smaller kernels (3×3) could make the network deeper which improves its accuracy on classification problems. Very deep CNN inherits the vanishing / exploding gradient problem described in 2.1.5 due to the depth of the network. The two network architectures ResNet and DenseNet tries to solve this problem. ResNet is a Residual Network and the title of the paper is called Deep Residual Learning for Image Recognition. ResNet solves the problem with diminishing results for even deeper networks than VGG by adding the previous convolved layers before a new convolutional block to the result after the convolutional block. When the network is backpropagated, the layers are subtracted, hence the name Residual Network.

An other way to solve the problem with vanishing / exploding gradients are with DenseNet or Densely Connected Convolutional Network. As for ResNet there are connections before and after a convolutional block. Instead of adding the layers they are concatenated in DenseNet. These connections are also referred to as skip connections. Skip connections in DenseNet are concatenated up to 5 convolutional blocks ahead. Skip connections serve two purposes, the distance between the classification layer and the input layer are minimised through the skip connections and additionally with the concatenation of layers the network will better remember previous layers and therefore not learn the same feature multiple times.

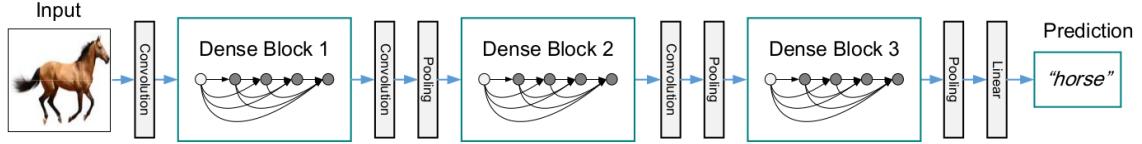


Figure 2.8: Example of DenseNet architecture. Previous convolutional layers are concatenated with other layers indicated with arrows.

2.5.1 Recalibrate Convolutional Layers

The main purpose of a CNN is to extract features that can be classified by the fully connected part of the network. [Hu et al. 2019] investigated a recalibration of the convolutional block to adaptively highlight important features. The recalibration block is called Squeeze and Excitation (SE). SE can be used in many other types of networks and has improved existing types of network. In figure 2.9 the SE block has been used on ResNet.

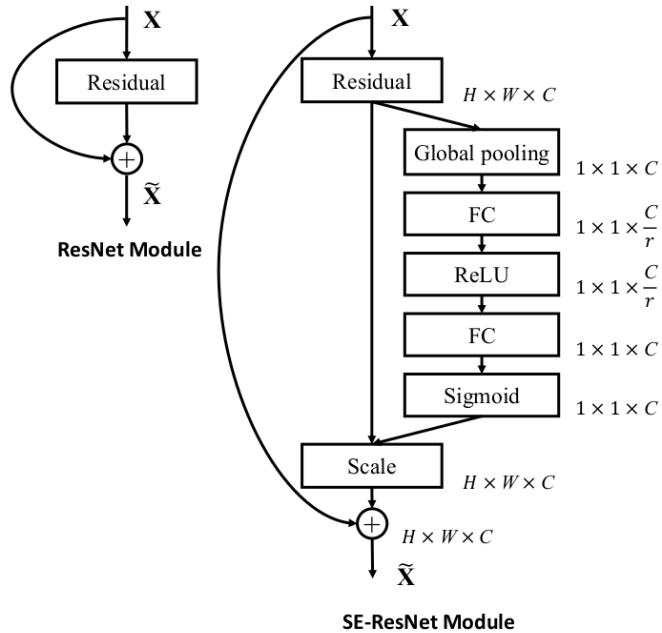


Figure 2.9: Squeeze and Excitation (SE) block used on Resnet. SE blocks recalibrate the convolutional layers and highlight important features.

The SE block squeezes the convolution layers into a lower representation by global pooling and further by fully connected layer with a reduction ratio r . The reduction is a trade-off between model complexity and accuracy for the network. The SE block adds little additional computational cost.

3 Synthetic Aperture Radar Theory

Synthetic Aperture Radar (SAR) emits electromagnetic waves. SAR is widely used for various remote sensing applications for earth observation. Compared to optical remote sensing the main benefits of SAR is that the signal can penetrate clouds and the target does not have to be illuminated by the sun. A SAR has an optical window through the atmosphere.

In the arctic region which is studied in this thesis and close to the polar circle there are darkness in the winter months and often clouded. A SAR has to be operated on a moving platform, which typical is on a satellite or aircraft. In the harsh climate and vast area in the arctic it is not possible to cover the entire area regularly with aircraft surveillance. Satellites have a good temporal resolution due to the frequent revisit over the same area. The lifetime of a Satellite is often guarantied for many years. With knowledge that new data will be available in the foreseeable future SAR satellites makes makes an ideal candidate for monitoring the arctic region.

3.1 SAR Background

A number of different wavelengths or bands are used dependent on the application. The wavelength of radar systems varies between 0.75 cm to 100 m or 40 GHz to 3 MHz. The naming scheme for radar bands was invented for military purposes. The European Space Agencies launched two SAR satellites Sentinel-1A and Sentinel-1B in 2014 and 2015. Sentinel-1A and Sentinel-1B uses C-band (4 GHz to 8 GHz) with a central frequency of 5.405 GHz. Other widely used bands are L- and X-band.

The SAR is side-looking and scans with a minimum to maximum incidence angle. The direction or path of the moving platform is called the azimuth direction and the scan direction is called ground range.

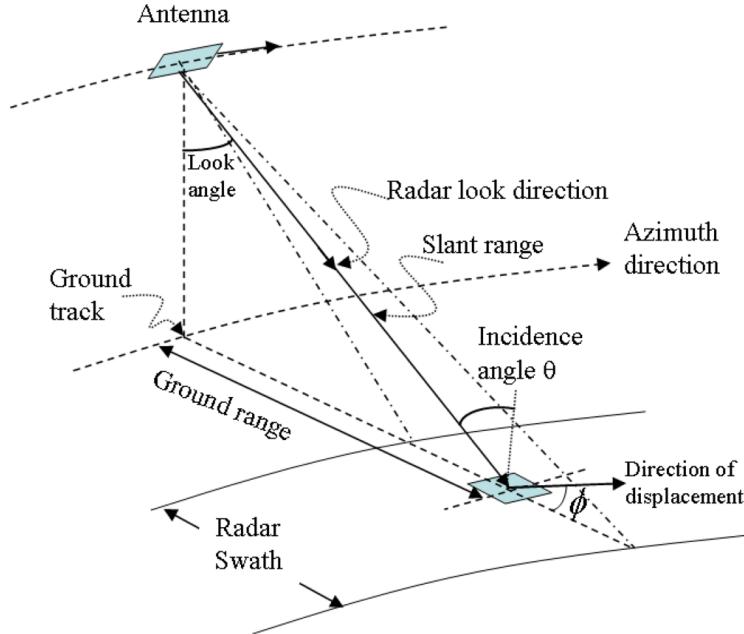


Figure 3.1: Side looking SAR antenna. The SAR scans from a minimum to maximum incidence angle called the radar swath.

The electromagnetic wave transmitted from the SAR hits a target in a given coordinate (x, y) which consists of an amplitude $A(x, y)$ and a phase $\phi(x, y)$. This gives a complex image of complex number pairs $A \cos \phi, A \sin \phi$. For each resolution cell many different targets can get hit. Each resolution cell is therefore a sum of the scatterers illuminated by the SAR.

$$A \exp i\phi = \sum_{k=1}^N A_k \exp i\phi_k \quad (3.1)$$

The summation of complex numbers in a resolution cell gives a noiselike feature to the image called speckle. To reduce the affect of speckle images are usually multi-looked. The speckle is reduced by averaging over the range and the azimuth resolution cells.

The intensity of radar backscatter depends on the target. In a SAR image the darker areas represent low backscatter where brighter areas represent high backscatter. The type of backscatter can be decomposed into three types of scatterers, surface, volume and double bounce scatter. A flat and smooth surface like water or a barren field will consist of primarily surface scattering. Vegetation is primarily volume scattering and tree chunks, man made object like buildings or ships is primarily double bounce. The type of scattering is also dependent on the wavelength of the SAR and the dielectric properties of the target. Water and metal has high dielectric constant and therefore the SAR cannot penetrate the object.

The geometry of a SAR image is very different from an optical image. A SAR emits pulses that are reflected by the surface. The placement of objects in a SAR image depends on the wave that are returned first. There is no difference in a flat landscape but distortions can be seen in a mountainous area or an area with tall objects.

Coastal areas in Greenland can have a steep terrain. The coastline can be misplaced in SAR images. Such distortions can also occur for ships and icebergs. This depends on the incidence angle, spatial resolution of the SAR and size of the object.

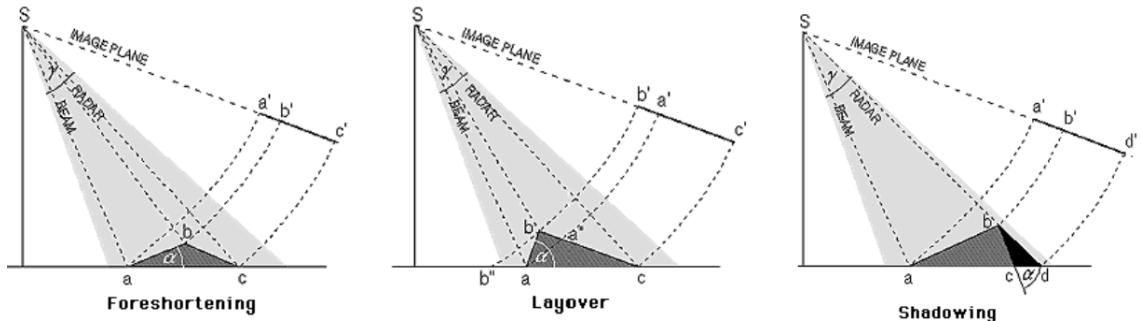


Figure 3.2: Three types of distortions that can occur in SAR images. Pixels for the three effects foreshortening, layover and shadowing will be misplaced in a SAR image.

Three types of such effects can be seen in figure 3.2 called foreshortening, layover and shadowing. The intensity of the backscatter varies with the incidence angle. More of the signal will be reflected back at a lower incidence angle than higher incidence angle and therefore is more bright in the image.

To make the backscatter independent on the resolution of radar system the backscatter is normalised with respect to the area. The backscatter can be projected into three different surfaces as can be seen in figure 3.3. The chosen projection depends on the application. σ^0 is the backscatter normalised with respect to the ground.

Backscattering coefficients

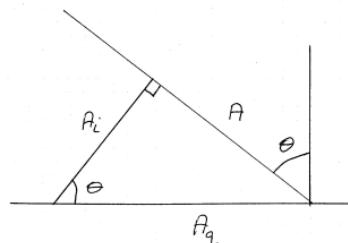


$$\sigma^0 = \frac{\sigma}{A_g} \quad \gamma^0 = \frac{\sigma}{A_i} \quad \beta^0 = \frac{\sigma}{A}$$

$$\boxed{\gamma^0 = \frac{A_g}{A_i} \sigma^0 = \frac{\sigma^0}{\cos \theta}}$$

$$\boxed{\beta^0 = \frac{A_g}{A} \sigma^0 = \frac{\sigma^0}{\sin \theta}}$$

$$\boxed{\beta^0 = \frac{A_i}{A} \gamma^0 = \frac{\gamma^0}{\tan \theta}}$$



$$\begin{aligned} \sin \theta &= \frac{A}{A_g} & \tan \theta &= \frac{A}{A_i} \\ &\Rightarrow \frac{A_g}{A_i} & &= \frac{1}{\cos \theta} \end{aligned}$$

Figure 3.3: Image from DTU remote sensing course. The SAR backscatter are projected into 3 planes and normalised with the area.

The sensor can have different polarisation's. The electromagnetic wave that propagates in a direction can have different signatures. The most general polarisation is elliptical but the majority of sensors uses linear polarisation and combinations hereof. The linear scattering matrix can have the following form:

$$S = \begin{bmatrix} S_{vv} & S_{vh} \\ S_{hv} & S_{hh} \end{bmatrix} \quad (3.2)$$

Where

- S_{vv} - Vertical transmitted, Vertical received
- S_{vh} - Vertical transmitted, Horizontal received
- S_{hv} - Horizontal transmitted, Vertical received
- S_{hh} - Horizontal transmitted, Horizontal received

3.2 Sentinel-1 Specification

Many processing steps have already been taken for the product that can be downloaded from ESA Sentinel-1. These products are of three types: level-0, level-1 and level-2 product. In this thesis only level-1 products will be used. The level-1 product is either given as a Single Look Complex (SLC) with both amplitude and phase or Ground Range Detected (GRD) that consist of multi-looked intensity. Different scan modes can be used dependent on the application. These are stripmap (SM), Interferometric Wide swath (IW), Extra Wide swath (EW) or Wave (WV). The data used in this thesis are level-1 GRD IW product. In IW mode the swath and azimuth distance of a SAR image is 250×200 km with a resolution of 22×20 m subsampled to 10×10 m. The two main modes used are IW and EW.

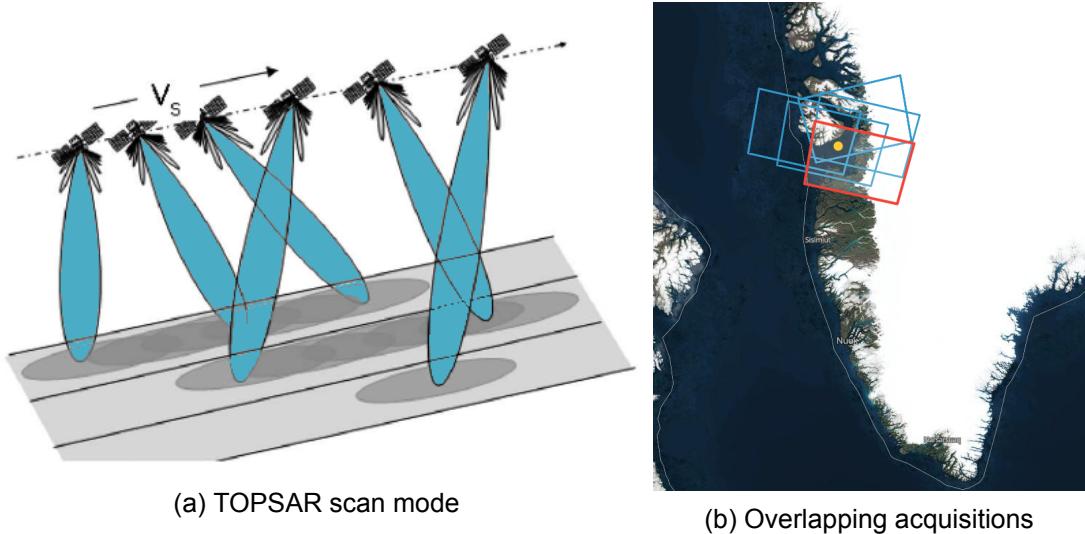


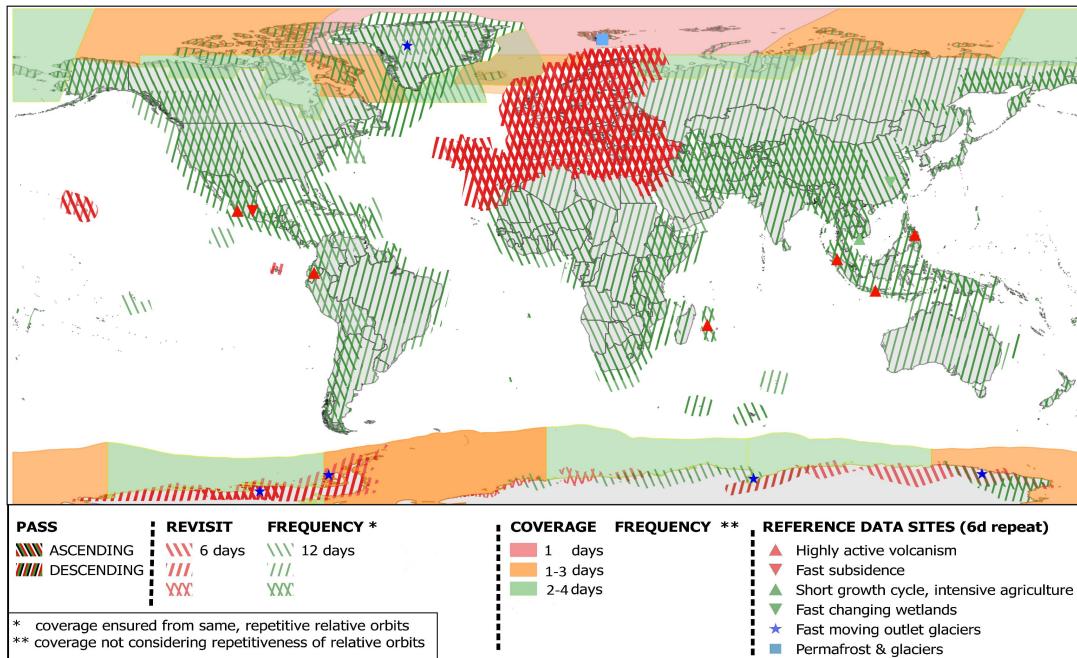
Figure 3.4: TOPSAR scan mode and a Sentinel IW scene is on a map. Multiple scenes are taken from the same location. The scenes are shown as overlapping acquisitions.

Characteristics IW	Value
Swath width	250 km
Incidence angle range	$29.1^\circ - 46.0^\circ$
sub-swaths	3
Polarisation options	Dual HH+HV, VV+VH, Single HH, VV

The IW scan mode is called TOPSAR and can be seen in figure 3.4a. The satellite scans in three subbands to cover a wider area. IW is primarily used over land and in coastal regions. EW is used over the ocean in the arctic region for sea ice detection to create sea ice maps. For sea ice detection the wider footprint is more important than the spatial resolution. The modes used in different parts of the world can be seen in figure 3.5b.

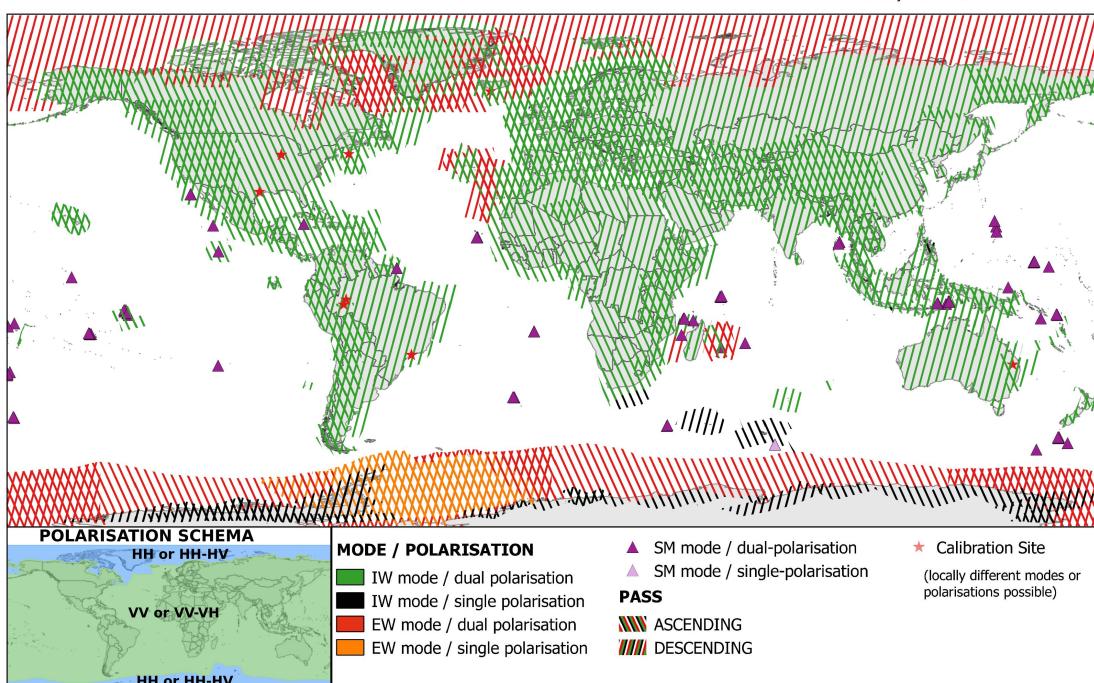
The two satellites Sentinel-1 A and B revisits the same exact spot every 12 day. The 2 satellites are in a constellation and share the same orbit such that the same spot is visited every 6th day. The Sentinel-1 satellites are near-polar orbiting with an inclination angle of 98.18° , so the frequency of revisit is occurring more often in polar regions. The satellites tracks are not the exact same in polar regions but consists of overlapping tracks. A few tracks over Disko bay area in Greenland can be seen in figure 3.4b from August 2020. The tracks are shifted in both longitude and latitude. The majority of acquisitions in figure 3.4b are descending in its orbit. In the transition area from ocean to land there is a shift in the scan mode. Ascending orbits in western Greenland starts from the ocean and because of this the sensor have to switch acquisition mode from EW to IW. Sentinel-1 does not store all data from the arctic. The satellites have to recharge and images in the arctic are in less demand. The revisit frequency can be seen in figure 3.5a.

Sentinel-1 Constellation Observation Scenario: Revisit & Coverage Frequency



(a) Revisit

Sentinel-1 Constellation Observation Scenario: Mode - Polarisation - Observation Geometry



(b) Mode

Figure 3.5

3.3 Scattering model of the ocean

Due to waters high dielectric constant the SAR will not penetrate the water but only record information about the surface. Observations over the ocean for satellites such as Sentinel-1 Bragg resonance is predominant. Bragg scattering is predominant for incidence angles above 20°. The formula for Bragg resonance is given by:

$$\lambda_s = \frac{n\lambda_r}{2 \sin \theta_i} \quad (3.3)$$

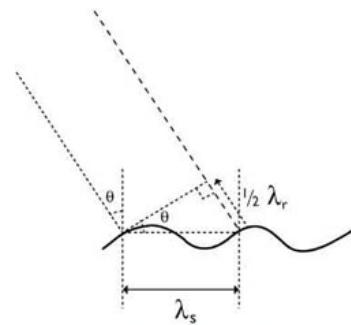


Figure 3.6: Bragg scattering.

Bragg wavelength λ_s , n order of resonance, λ_r radar wavelength and θ_i incidence angle, [Crisp 2004]. If the length between the ocean waves matches the radar wavelength the backscatter will increase. As described previously for speckle is each resolution cell a summation of individual contributions. In case of Bragg scattering there is constructive resonance and the phase of the individual waves that contributes to the resolution cell are aligned. The backscatter of the ocean is strongly dependent on the wind speed and direction. If there is no wind and hence no waves in the ocean, little to no information are returned to the SAR. The surface of the ocean will be dominant specular reflection. The wind and incidence angle have the following effect on the backscatter σ^0 .

- σ^0 increases with increasing wind speed
- σ^0 is largest when the wind is blowing towards the radar
- σ^0 is smallest when the wind is blowing across the radar look direction
- σ^0 for C-band VV is larger than σ^0 for C-band HH for all wind speeds and directions
- σ^0 decreases with increasing incidence angle
- σ^0 HH for C-band decreases more rapidly with increasing incidence angle than σ^0 for C-band VV

This list is taken from [Crisp 2004]. The roughness of the ocean and amplitude of the waves increases with wind speed. The backscatter changes is dependent on the wind direction. The shape of the waves seen from the SAR waves curve concavely when wind is blowing towards the radar and therefore increases the backscatter. Smooth surfaces where little backscatter is returned as for lakes, ocean and other water surfaces acts like a mirror. Increased incidence angle will therefore decrease the backscatter.

3.4 SAR detection of moving targets

It is assumed in SAR images that all objects within a scene are stationary. Moving targets within a SAR image will be exposed to some effects due to movement in azimuth and range. The effect with the greatest importance for ship detection are azimuth image shift. A moving target will appear misplaced in the SAR image in the azimuth direction if the target is moving with a range velocity component. For satellite SAR the displacement can be corrected for by:

$$\delta x = -u_R \frac{RV_p}{V^2} = -u_R \frac{R}{V} \frac{r+h}{r} \quad (3.4)$$

Where δx is the displacement, u_R is the range velocity component of the target, R is the range to the target, V_p is the satellite speed and V is the ground speed from the satellites view. For Sentinel 1 are these values from equation 3.4:

$$\delta x = -u_R \frac{693 \text{ km} / \cos(37.55^\circ)}{7.4 \text{ km/s}} \frac{6371 \text{ km} + 693 \text{ km}}{6371 \text{ km}} = -u_R \times 129.2 \text{ s} \quad (3.5)$$

Where $r = R_{\text{earth}} = 6371 \text{ km}$, the satellite altitude $h = 693 \text{ km}$ and the satellite speed $V = 7.4 \text{ km/s}$.

A fishing trawler sails at a speed around 8 knot or 4.12 m/s if the ship sails in the SAR azimuth direction the displacement will be 532.3 m or 53 pixels in the SAR image.

3.5 Backscatter Normalisation

The backscatter intensities from the illuminated area are also sensitive to the incidence angle and not only the terrain. This has the effect of higher intensities at lower incidence angles compared to higher incidence angles. Backscatter Normalisation is important when using scenes from multiple acquisition because the orbit tracks can change. In polar regions many overlapping acquisitions happens and the incidence angle to a give object can therefore vary from scene to scene.

To minimise backscatter variations a cosine correction based on Lambert's law for optics are given as:

$$\sigma_{\theta_i}^0 = \sigma_0^0 \cos^n(\theta_i) \quad (3.6)$$

Where $\sigma_{\theta_i}^0$ is the cosine dependent measured backscatter. σ_0^0 is the backscatter independent on the incidence angle. n is a roughness dependent weighting factor where default is 2 (1 and 2 is often applied). This law is under the assumption that re-radiated power in the upper hemisphere follows a cosine law.

The backscatter is normalised to a reference angle θ_{ref} :

$$\sigma_{ref}^0 = \frac{\sigma_{\theta_i}^0 \cos^n(\theta_{ref})}{\cos^n(\theta_i)} \quad (3.7)$$

The reference angle θ_{ref} is set to 37.55° for Sentinel-1 GRD IW products. Mladenova et al. 2013 notes that the cosine based normalisation is suitable if applied over areas with "simple structures". Other methods for backscatter normalisation is proposed in the article.

4 Dataset of ships and icebergs

In supervised deep learning it can be problematic to get a large enough dataset. For this reason a prelabelled dataset from Statoil/C-CORE has been used. There was no information about how the Kaggle dataset was preprocessed. The dataset is extended with more ship and icebergs from Sentinel-1 SAR images. As part of the study, Sentinel 1 images have been prepared and processed. The preprocessing steps have been investigated to generate the best possible images. The data prepared in this study is matched to the same format as the dataset from Statoil/C-CORE.

4.1 Prelabelled data

The Kaggle dataset consists of 10,000 ships and icebergs, [Statoil/C-CORE Iceberg Classifier Challenge 2017]. These 10,000 images consist of 1600 images split into a training set that are labelled and 8400 test images that are not labelled. The data is labelled with a column called *is_iceberg* where iceberg is labelled 1 and ships are labelled 0. These images are of size $75 \times 75 = 5625$ pixels. The images has two polarisation's

- Horizontal transmitted - horizontal received (HH)
- Horizontal transmitted - vertical received (HV)

Further is the incidence angle from the Sentinel-1 satellite to the target a parameter provided. The first 5 rows of the dataframe can be seen below:

id	band_1	band_2	inc_angle	is_iceberg
dfd5f913	[-27.878, -27.154, ...	[-27.154, -29.538, ...	43.924	0
e25388fd	[-12.242, -14.920, ...	[-31.506, -27.985, ...	38.156	0
58b2aaa0	[-24.604, -24.604, ...	[-24.871, -24.093, ...	45.286	1
4fcfc3a18	[-22.455, -23.083, ...	[-27.889, -27.520, ...	43.831	0
271f93f4	[-26.007, -23.165, ...	[-27.207, -30.259, ...	35.626	0

Band_1 is HH polarisation and Band_2 is HV polarisation. The backscatter is given in a dB scale.

$$\sigma_{dB}^0 = 10 \log_{10}(\sigma^0) \quad (4.1)$$

An unique id has been given to each image for both the training and test dataset. The images in the test set is assigned a probability within 0 (ship) and 1 (iceberg) with the trained neural network. The probabilities are uploaded to the Kaggle competition and the user is assigned a score. The unique id ensures that the correct score for the test set can be calculated.

4.2 Data prepared in this study

To extend the Kaggle Statoil/C-CORE dataset was Sentinel-1 scenes used from Greenland in the Nuuk and Disko bay / Ilulissat area. In the arctic region HH and HV polarisation are used due to better iceberg detection in these polarisation's. Disko bay was chosen due to Ilulissat glacier and many iceberg flows in this area. This glacier produces around 10% of Greenland's icebergs. To ensure a dataset with both icebergs and marine traffic, the Sentinel-1 SAR images from August 2020 were used. It was difficult to find an area with frequent ship traffic in this region where the Sentinel-1 polarisation's were HH and HV with IW scan mode. Besides the Disko bay area where the ships are mainly fishing

boats, Nuuk was chosen to include military vessels and other types of ships to get a more versatile dataset. The ships were found with Automatic Identification System (AIS) data from Gatehouse. The timestamp from AIS was matched with the Sentinel-1 images. The data were saved with the same naming scheme and variables as used for the Kaggle Statoil/C-CORE dataset. Furthermore the backscatter was found to be dependent on the incidence angle. Therefore a correction degree was saved as well for each band. The data stored about the ships were: shiptype, length, width and speed over ground (SOG) saved too. This additional information were available in the AIS data.

5 Method

The Kaggle Statoil/C-CORE dataset has already been preprocessed from Sentinel 1 scenes to images of ships and icebergs. The Kaggle dataset was explored further to find patterns that might be exploited in the neural network to discriminate between ships and icebergs.

The custom data used in this study has been processed from Sentinel 1 images. A method for preprocessing the Sentinel 1 scenes are proposed in section 5.3.

5.1 Prepare backscatter values for Neural Network

Many features implemented in neural networks assume that the data are Gaussian distributed. The backscatter amplitude σ^0 is multi-looked Gamma distributed. The backscatter was converted to a dB scale so the data could be assumed Gaussian. The distribution of 1600 ships and icebergs from the Kaggle Statoil/C-CORE dataset is shown in figure 5.1.

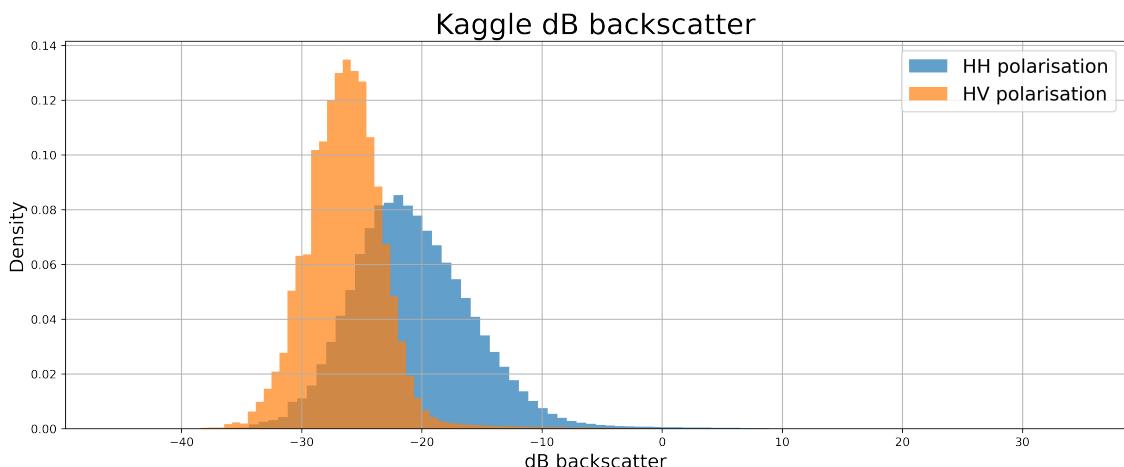


Figure 5.1: Density distribution of σ_{dB}^0 for HH polarisation (blue) and HV polarisation (orange)

The statistics for the training data can be seen below:

	Minimum	Maximum	Mean	Standard deviation
band_1 HH	-45.594	34.575	-20.697	5.218
band_2 HV	-45.655	20.154	-26.347	3.403

The backscatter from ships and icebergs are several magnitudes greater than the ocean in the background. From figure 5.1 and the table it shows that the distribution is very elongated for positive values.

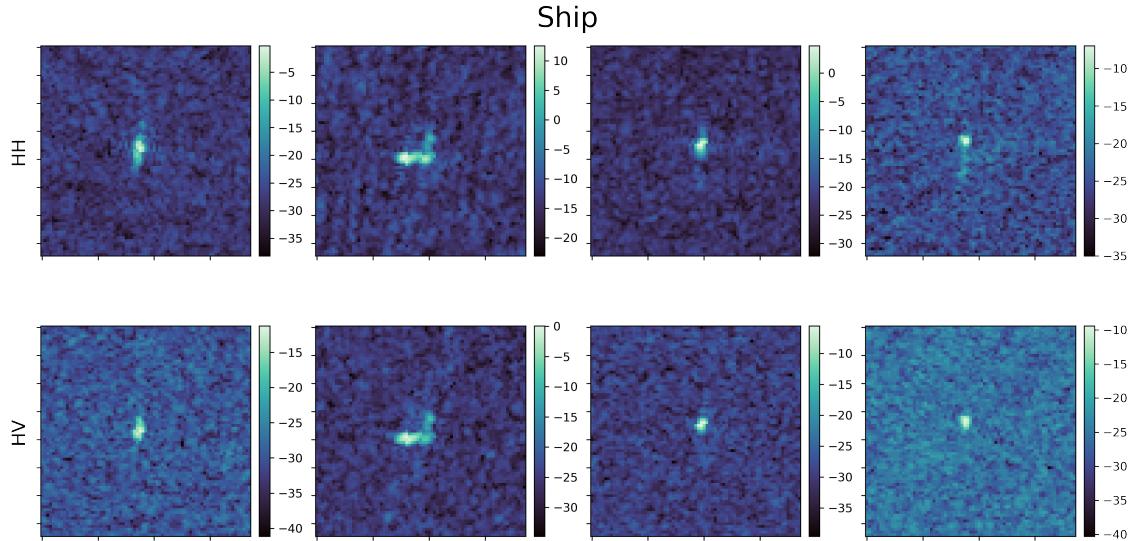


Figure 5.2: Four different ships in HH and HV polarisation.

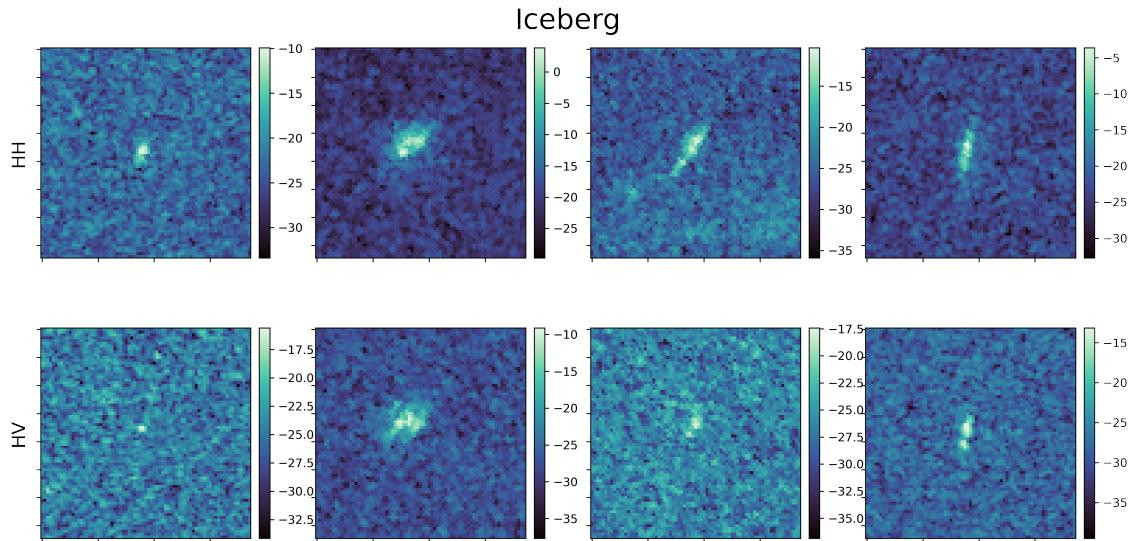


Figure 5.3: Four different icebergs in HH and HV polarisation.

Examples of ships and icebergs can be seen in figure 5.3 and 5.2.

5.1.1 Rescale Normalisation

The input images for the CNN should be normalised either in the range $[-1, 1]$ or $[0, 1]$. This ensures that the network is less sensitive to variations in the data. The data is therefore normalised between $[0, 1]$ by:

$$dB_{norm} = \frac{dB - dB_{min}}{dB_{max} - dB_{min}} \quad (5.1)$$

dB_{min} and dB_{max} is found globally for both the training and the test data. The global minimum and maximum is found to $[-49.08, 37.09]$.

5.1.2 Third Band

A Norwegian Defence Research Establishment (FFI) report, [Lensjø 2018] suggest that combining the two polarisations channels HH and HV can get a better contrast between the object and the ocean. The report suggest that the backscatter values can be combined in the following way

$$\frac{HH \times HV}{C} \quad (5.2)$$

or in dB scale $HH_{dB} + HV_{dB} - C$. Where C is the average ocean backscatter of the two polarisation's. To make sure that the new third band is within the same range of values as the two channels HH and HV the average of the two bands are used:

$$\frac{HH_{dB} + HV_{dB}}{2} \quad (5.3)$$

5.1.3 Size of object

Information about the size of ships (length \times width) and icebergs in SAR images can help classify the object. From figure 5.2 and 5.3 it can be seen that it is not always easy to distinguish the ocean from the object. This is especially the case for HV polarisation. A method used to get a estimated pixel area of the object is to assign pixels that are further than 2 standard deviations away in the positive direction to the object. This was investigated but showed to include pixels in the background. To get a more precise estimate of the size the two channels are converted back to non dB values (σ^0). The channels are then combined by equation 5.2. The rescale constant C is set adaptively and is dependent on the ocean background. The distribution is no longer assumed Gaussian and a hard threshold is set.

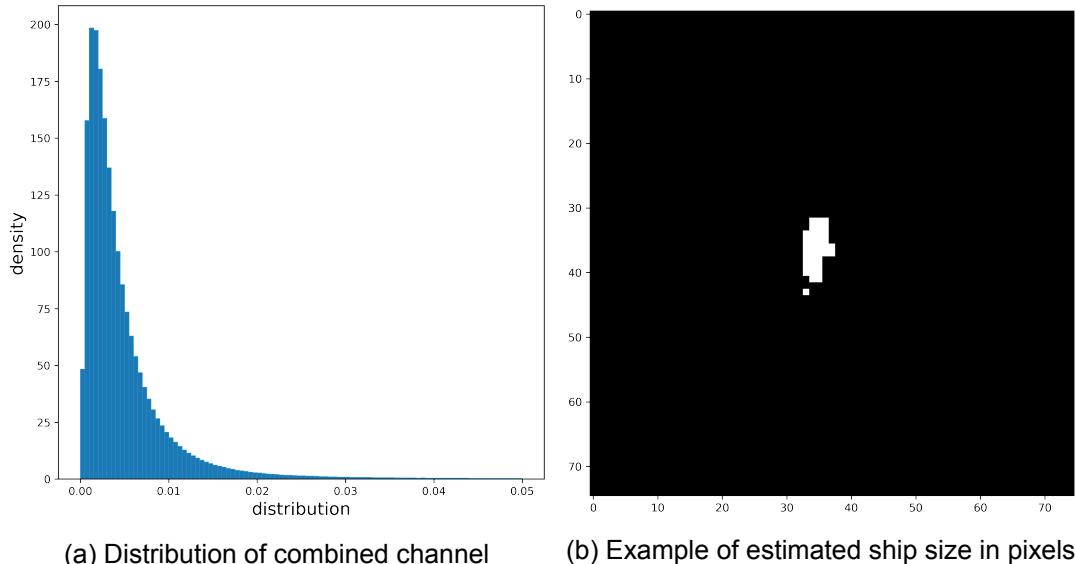


Figure 5.4: Distribution of Kaggle Statil/C-CORE training dataset in the combined band calculated from equation 5.2. A threshold has been set and the dominant pixels of a ship can be seen in b).

In figure 5.4a the distribution for the 1600 ships and icebergs can be seen. The hard threshold T for pixels is set to:

$$pixel = \begin{cases} T \leq 0.04 & \text{Background} \\ T > 0.04 & \text{object} \end{cases}$$

For Small objects the magnitude of the backscatter are not as dominant as for larger objects. The threshold is set to ensure pixels for smaller objects can be detected as well. Figure 5.4b is the same ship as the first ship in figure 5.2. For IW GRD SAR the image is a resolution cell 22×20 m and subsampled to 10×10 m.

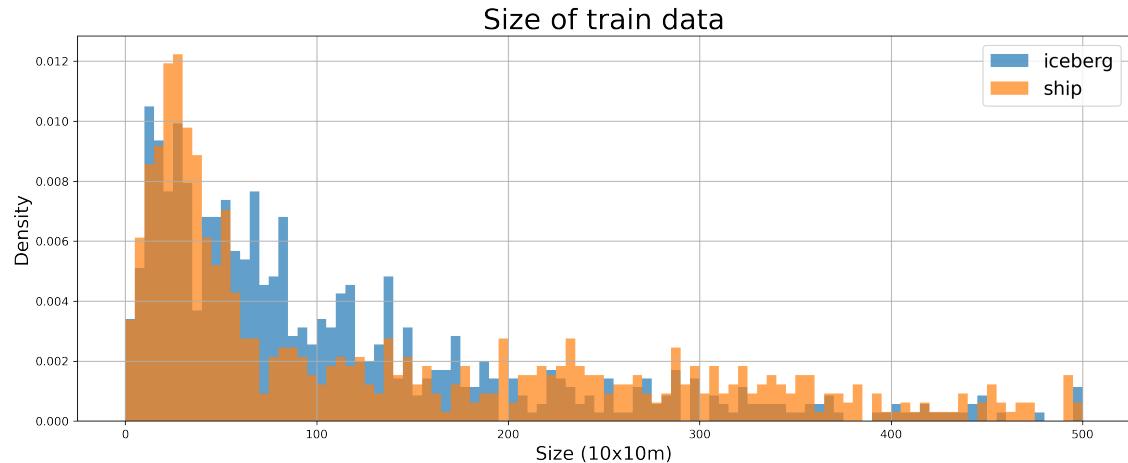


Figure 5.5: Estimated size of ships and icebergs from the Kaggle training set. The estimated size is dependent on the threshold.

In figure 5.5 it shows that there is a difference in the size distribution of icebergs and ships. This information is added to the fully connected part of the neural network. The size node in the neural network is rescaled by $\frac{\text{pixels object}}{\text{total pixels in image}}$.

5.1.4 Data augmentation

With only 1600 ships and icebergs where 80% is used to train the network makes the CNN prone to overfitting. It is assumed that the target is centered in the image. To enlarge the dataset a number of different data augmentation techniques are used. Augmentations should ideally be avoided in the SAR images because the orientation of SAR has great influence on the images. The SAR satellite can be either ascending or descending in its orbit. Because the area of interest being over water the effect of foreshadowing, layover and shadowing is not as dominant. Flipping and rotation of the images is therefore proposed.

A simple way to enlarge the dataset is to flip the images. This was done either with a horizontal or vertical flip or in both directions at once. When the batch for a given epoch is loaded into the network each type of flip is set to a 50% probability to occur. This alone artificially gives 3 times more data to train on.

Rotation of the images is proposed with random rotation angles from 0° to 45° . Together with flipping the images all orientations of the ships and icebergs are possible. When the image is rotated there is no pixels in the corners of the image. All values that are outside the image due to rotation is set to 0. This gives an unwanted skewness of the distribution of pixels. The usage of rotation should be minimised if possible.

Lastly the translation of the objects are proposed. This is done by enlarging a image from 75×75 pixels to 86×86 pixels and then crop the image randomly back to 75×75 pix-

els. Usually in order to resize the image interpolation are used. In this case the size of each pixel have a physical size of 10×10 m and therefore padding is used. For this a padding with reflect mode is proposed. Reflect mode mirrors the borders of the image without reflecting the outermost pixel. A sequence [1, 2, 3, 4] results in [3, 2, 1, 2, 3, 4, 3, 2] if the padding is set to 2.

5.1.5 Proposed Network Architecture

For the convolutional neural network a variation of a U-Net is used. A U-Net is normally used for full segmentation of an image where all pixels are assigned a class. U-Net was originally developed for biomedical image segmentation but can be generalised to other areas. U-Net is capable of extracting information about the image in different scales. The modified U-Net can be seen in figure 5.6.

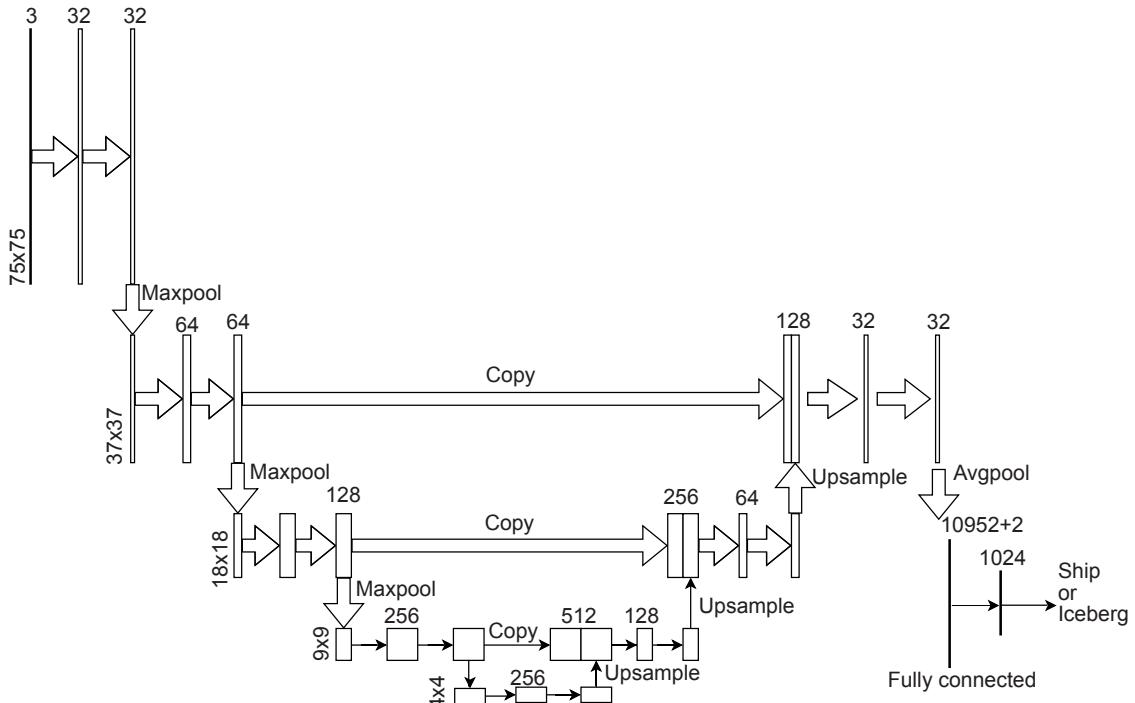


Figure 5.6: Proposed modified Unet. The network is composed of a feature extraction part with convolutional layers, maxpool upsampling and skip connections. The feature extraction part is avgpooled to a classification part where the estimated target size and incidence angle is concatenated to the flattened result from avgpooling.

Three bands is used for input to the neural network, HH, HV aswell as a combined band of the two polarisation's. All convolutional parts of the network consists of:

- 3×3 convolution with 1 padding to keep the dimension
- Batch normalisation
- ReLU

In each scale of the U-Net two convolutions are used continued by either an downsampling or upsampling. This is illustrated in in figure 5.6 where the architecture resemble an U and hence the name. The modified version of U-Net has an upsample less than the original U-Net has. This is to avoid the increasing number of parameters when the convolutional layers are flattened into the fully connected part of the neural network.

The modified U-Net has 4 downsampling (maxpool) and 3 upsampling. The possible depth of the network depends on the size of the input images. The convolutional layers from the downsampled part of the network are concatenated together with the upsampled part. This ensures that information from different scales of the network is contained. After the last upsampling there are two convolutional layers. The last convolutional layers are average pooled and flattened into the fully connected classification part of the network. The first fully connected layer consists of 10952 neurons connected to a layer with 1024 neurons. In the fully connected layer with 10952 neurons there are 2 more nodes concatenated. The two nodes have information about the estimated pixel area of the ship/iceberg from the SAR image and the incidence angle to the target. A second layer with 1024 nodes was connected to the output layer with a sigmoid function since binary cross entropy loss is being used.

After each downsampling (maxpool) and upsampling part of the network a squeeze and excitation block are applied to recalibrate and extract the most important features utilised later in the network.

In appendix A the modified Unet, the layers and number of parameters are shown. The modified Unet has a total of 13,965,025 trainable parameters. A total of 123 layers was utilised in the network. The majority of the parameters emerge from the fully connected part of the network where the 10952 neurons was connected to the next layer with 1024 neurons. The fully connected part of the network alone counts for 10,618,881 parameters.

5.2 Preparing Sentinel-1 Scenes

As described in section 4, there was no prior information about which preprocessing steps that were applied on the Kaggle Statoil/C-CORE dataset. Therefore some assumptions had to be made in regards to what preprocessing steps were appropriate when downloading Sentinel-1 scenes, for the purpose of extracting 75×75 pixels images of ships and icebergs. Many different preprocessing steps can be taken for the Sentinel 1 scenes. The custom labelled data is used to extend the Kaggle dataset and to test if the dataset from Statoil/C-CORE can be generalised on similar data. To get the best possible result within the time frame of a thesis the following preprocessing steps are used.

5.3 Sentinel-1 GRD preprocessing

ESA Sentinel-1 level-1 scenes are used from the Disko bay / Ilulissat area and Nuuk. The available scenes to download are free. The type of SAR scene used is GRD IW with a footprint of 250×200 km and resolution cell of 22×20 m that are subsampled to 10×10 m.

Proposed preprocessing steps with Sentinel-1 GRD IW products:



Figure 5.7: Processing chain for preprocessing Sentinel-1 IW GRD data

ESA has developed a program called SNAP that can perform these preprocessing steps.

5.3.1 Apply orbit file

The position at the time of recording is acquired by the Global Navigation Satellite System (GNSS). The orbit file improve the satellite position and velocity information in the SAR product which is generally not accurate enough from GNSS. The calculation and generation of the precise orbit file takes time. The orbit files are usually available within

a few days-to-weeks after the product is generated. The Precise orbit file has an accuracy less than 5cm for the satellite, [Weiß 2019]. The orbit file is often the first step in the prepossessing chain since it influences and improves later steps, especially for geo-referencing. The orbit file updates the existing metadata in the product file.

5.3.2 Thermal noise removal

Thermal noise is caused by background energy from the SAR receiver. The thermal noise is additive to the backscatter from the SAR receiver. The noise is more dominant in the cross-polarization channel. The thermal noise has greater influence on areas with less backscatter such as calm water bodies. The noise is reduced by normalising the backscatter across the entire Sentinel-1 scene.

5.3.3 Border Noise Removal

The processing steps from the RAW data into L1 product produces artifacts in image borders. These artifacts are mainly caused by sampling start time changes to compensate for the earth's curvature. Azimuth and range compressing contributes to artifacts in the borders of the SAR image, [Filipponi 2019]. Border Noise Removal removes low intensity noise and invalid data near the borders of the Sentinel-1 image.

5.3.4 Calibration

The backscatter of the SAR scene is stored as digital pixel values (DN). In the calibration step the digital pixel values are converted to radar backscatter values. This calibration is necessary to truly represent the radar backscatter of reflecting surfaces and to compare the backscatter at different scenes with each other. All values and parameters used to compute the calibration are available in the product information. The equation to calculate the radiometric calibration is given as:

$$Value(i) = \frac{|DN_i|^2}{A_i^2} \quad (5.4)$$

Where $value(i)$ is either $\beta_i^0 \sigma_i^0 \gamma_i^0$ or original DN . A_i is either $\beta_0(i) \sigma_0(i) \gamma_0(i)$ or $DN(i)$. Depending on the application of the SAR scene and the SAR incidence angle one of the four calibrations are chosen. For an ocean with iceberg and ship detection is σ^0 used to be able to compare the data to Kaggle Statoil/C-CORE dataset.

5.3.5 Speckle Filtering (Optional)

Speckle Filtering removes granular noise from elementary scatterers within a resolution cell. Speckle filtering might remove small spatial structures and therefore should not be used in ship detection if the objective is to find small fishing vessels.

Theoretically the Convolutional Neural Network should be able to learn the speckle filter if it boosts the accuracy of the classification.

5.3.6 Range Doppler Terrain Correction

Range Doppler Terrain Correction corrects for distance distortions in the image. This happens because the satellite is side looking upon the terrain. Some common distortions are foreshortening, layover and shadowing. Pixels within the SAR scene will look "unnatural" because the receiver of the satellite will log the data in terms of distances and not angles. Furthermore the scene is compensated with respect to WGS84 ellipsoid by using a Digital Elevation Model (DEM). The Range Doppler Terrain Correction compensate for these distortions such that the image have a geometric representation which is close to the real world.

The Range Doppler orthorectification is used to geocode the SAR scene from radar geometry to a desired projection. The Range Doppler orthorectification uses orbit state vector information, radar timing annotations and the slant to ground range conversion parameters together with the reference DEM, see [SNAP help documentation]. The user can set a pixel spacing in either degrees or meters. This preprocessing step is useful to merge datagrids from different sensors or a scene at a different time.

5.3.7 Conversion to dB (Optional)

The processing step converts the image to a dB scale. The conversion to dB scale gives pixel values that are easier to interpret for humans when the image is visualised. For Constant False Alarm Rate (CFAR) algorithm it was preferred to work with σ^0 and therefore this preprocessing step was preformed later.

$$\sigma_{dB}^0 = 10 \log_{10} \sigma^0 \quad (5.5)$$

5.3.8 Backscatter Normalisation (optional)

The backscatter normalisation described in section 3.5 requires the ellipsoidal incidence angle to be saved from the Range Doppler Terrain Correction processing step. There is no default implementation of backscatter normalisation in ESA SNAP API. Because of the simplicity of the normalisation this step can easily be done in Python or an other programming language. It can be an advantage to compute the normalisation in decibel by:

$$\sigma_{dB,ref}^0 = \sigma_{dB,\theta_i}^0 + 10 \log_{10} \left(\frac{\cos^n(\theta_{ref})}{\cos^n(\theta_i)} \right) \quad (5.6)$$

5.3.9 Write

The output file is saved as a Geotiff image for further processing in Python instead of the default dim file SNAP are programmed to use.

5.4 Land Masking

The CFAR algorithm is unsupervised and detect targets that have higher values than a given threshold. In coastal regions such as investigated here, land areas will give false positives. It is therefore desired to mask out land, small island and rock formations. A bathymetry model called International Bathymetric Chart of the Arctic Ocean (IBCAO)¹ is used for this purpose.

A Digital Elevation Model could also be used for land masking. A bathymetry model gives additional information about the depth of the sea. Depending on what type of ship which was being searched for the masking could be extended out in the sea. It is not realistic that a tanker sails on shallow water and therefore the masking can be set adaptively dependent on the ship.

¹https://www.gebco.net/data_and_products/gridded_bathymetry_data/arctic_ocean/

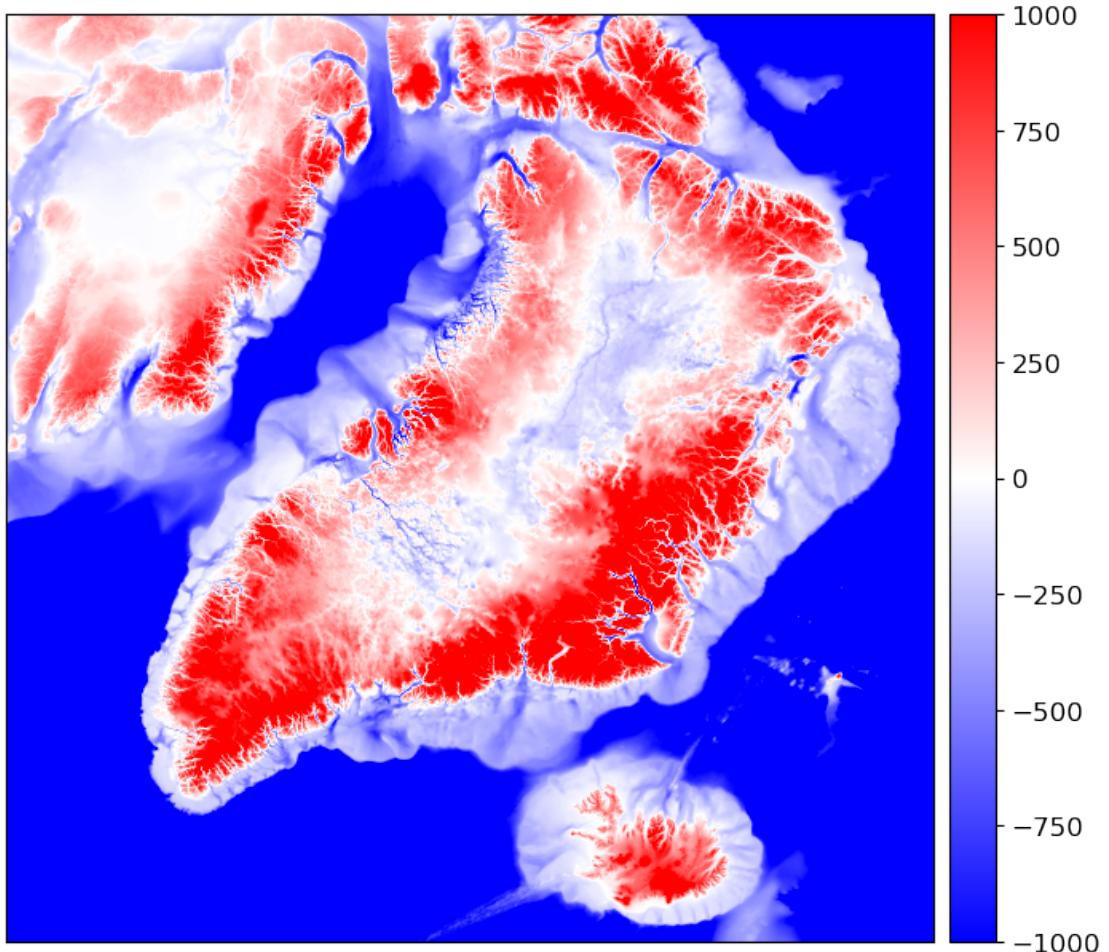


Figure 5.8: Bathymetric data chart for Greenland. The colorbar is given in meters. The chart is displayed in a Polar Stereographic projection.

5.4.1 Processing Steps

The chart was geocoded into the same map projection as the SAR images and the same area as the scene was then extracted. The IBCAO model was converted from Polar Stereographic projection into a local UTM map projection. The IBCAO model has a resolution of 200×200 m. The model is subsampled to 10×10 m to match the resolution of the Sentinel-1 SAR images.

The relative poor resolution of the IBCAO model gives some uncertainty around the transition from land to sea. The SAR images could be distorted due to foreshortening, layover and shadowing in the transition between land and sea. To ensure that all areas of landmass were masked out, a binary mask was created where all areas under the sea surface were set to 0 and positive elevation was set to 1. A image analysis technique called binary dilatation was then used to extent the border of land into the sea. Binary dilation pads 1 to the pixels around a transition area. A matrix M with one dilation step would result in:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \implies \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.7)$$

One dilation step is the same as to extend the land 10 m. 20 dilatation steps are used and hence 200 m zone around land is extended into the sea.

5.5 Constant False Alarm Rate

Constant False Alarm Rate (CFAR) is an adaptive threshold algorithm. CFAR searches for pixels in an image that have high intensity compared to the surrounding. CFAR cannot discriminate between the type of targets but only if its likely that a target of interest is in the image.

The algorithm consists of 3 boxes, a target box, a guard area and a background area. The target box contains the area or pixel that are being investigated for targets. The guard area ensures no smearing or contamination from the target to the background area. The target box is compared to the statistics of the background area. A threshold is set and if a target is unusually bright compared to the background it is considered a target. All three windows slides over one pixel in each iteration such that all pixels in the image is investigated. The setup can be seen in figure 5.9. The size of the background area is set to 75×75 pixels which is the same size as the Kaggle images. It is very unlikely that either ships or icebergs are larger than 750 m in a SAR image. The guard area is set to 200×200 m where the target box is set to 1 pixel.

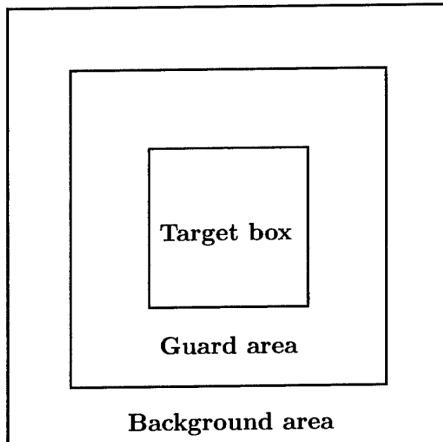


Figure 5.9: CFAR algorithm. The target box is under investigation and compared to the background area. The guard area ensures the target does not influence the background area.

The probability of false alarm (PFA) for a threshold is given by:

$$PFA = 1 - \int_{-\infty}^T f(x)dx = \int_T^\infty f(x)dx \quad (5.8)$$

Where $f(x)$ is the associated parametric probability density function, [Crisp 2004]. The threshold can either be set by trial and error or by using information about the underlying multi-looked gamma distribution. To obtain a better contrast between the background and the target, the two bands HH and HV was merged with equation 5.2. The distribution has thus changed and the threshold is set by trial and error. Furthermore the threshold was set high enough to make sure that ships were always detected while icebergs with weak backscatter was not.

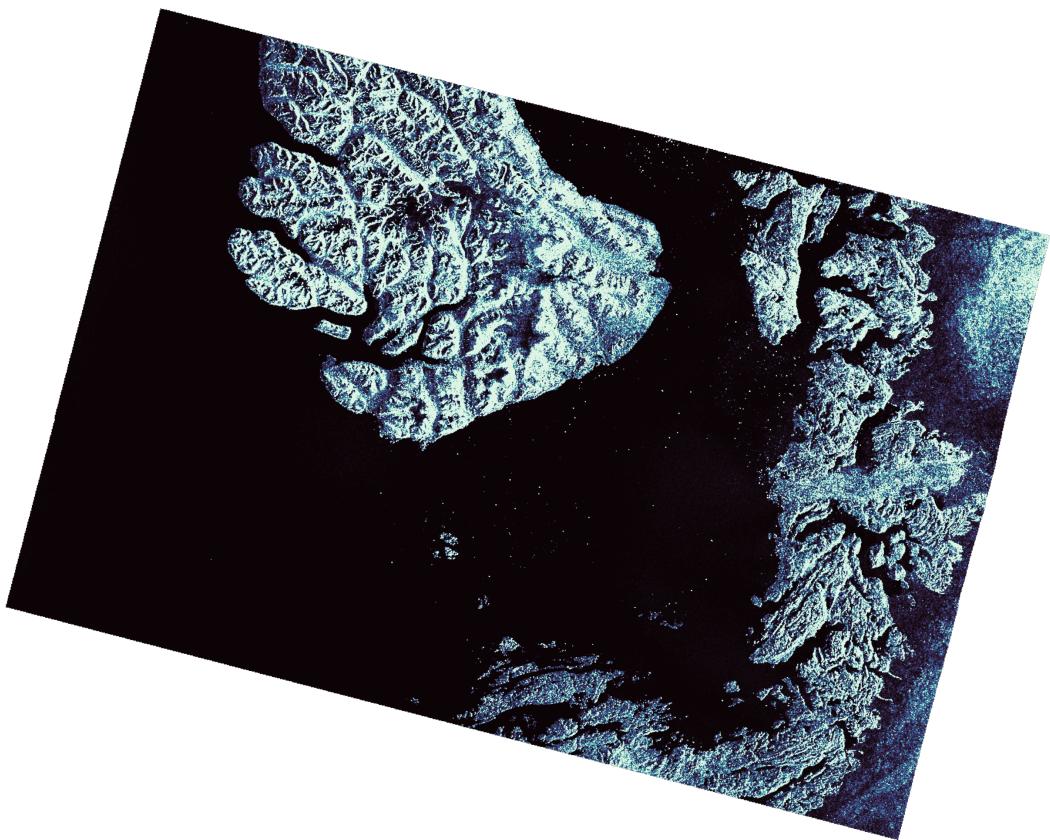


Figure 5.10: Sentinel-1 SAR image from 2020/08/11 – 10:08 of Disko bay in Greenland. The image is made from equation 5.2 with the two bands, HH and HV. The image is of size 250×200 km. The white dots in the ocean are likely icebergs.

5.6 Automatic Identification System

Automatic identification system (AIS) is a transmitter / receiver system for ships. All EU fishing boats over 15m are required to have a AIS transmitter. In coastal areas ground stations are used to receive and log the signal. In open water or less populated places are the signals received and logged by satellites. It is up to the individual ship to transmit the correct information. The ships can transmit wrong information or have the AIS transmitter turned off. This can either be due to human error or illegal activity. For marine surveillance SAR images are used to find and detect vessels. The ships transmits different information. The important information for ship detection in satellite images are:

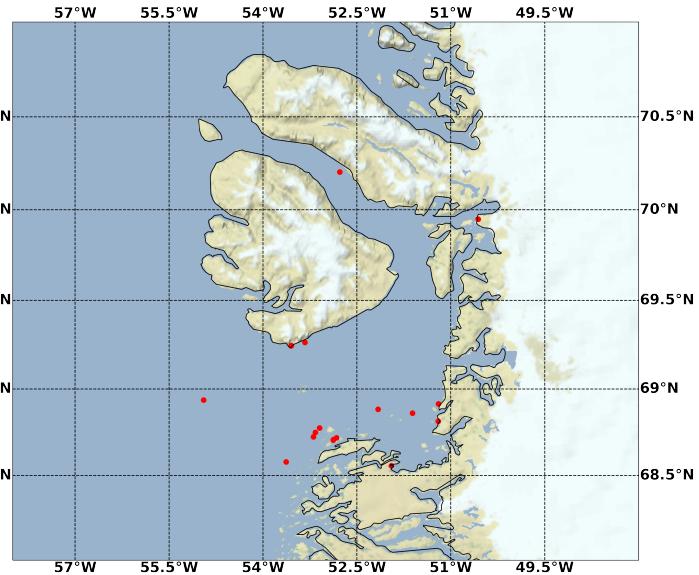
- timestamp
- Location in longitude and latitude

- MMSI number, a unique id for ships
- Ship type (fishing, cargo, military etc.)
- length and width of the ship
- Speed over ground

In the area around Nuuk and Disko bay in Greenland the AIS signal is logged by satellites. A satellite needs to have communication with the AIS transmitter. There can be some gaps in the data logging if no satellites are available. The displacement in SAR images of moving targets were not compensated for, due to the possible time difference in the timestamp from the AIS and the SAR image. Furthermore there is an uncertainty of the position of the vessel tracked by satellite around 200 m [Sørensen 2021].

lat	lon	length	width	shiptype
68.814513	-51.201038	108.0	22.0	Cargo ship
68.776267	-53.096833	46.0	13.0	Cargo ship
68.708860	-52.874457	63.0	12.0	Cargo ship
68.556870	-51.951135	20.0	6.0	Fishing vessel
68.724963	-53.196590	30.0	8.0	Fishing vessel
68.883245	-52.162795	21.0	6.0	Fishing vessel
68.706400	-52.877800	23.0	6.0	Fishing vessel
68.718913	-52.825603	19.0	6.0	Fishing vessel
68.750900	-53.162997	25.0	6.0	Fishing vessel
69.245248	-53.552035	15.0	6.0	Pilot vessel
68.862037	-51.612538	35.0	8.0	Fishing vessel
68.708230	-52.875685	20.0	7.0	NaN
68.936705	-54.949453	42.0	10.0	Fishing vessel
68.579128	-53.632718	34.0	12.0	Other type of ship
68.914952	-51.199092	14.0	4.0	Passenger ship
69.264850	-53.330215	NaN	NaN	Passenger ship
69.948652	-50.563110	12.0	4.0	Fishing vessel
69.949600	-50.565950	15.0	6.0	Fishing vessel
70.205000	-52.775000	15.0	4.0	Fishing vessel
69.247230	-53.548053	NaN	NaN	Fishing vessel

(a) Ship attributes from AIS data.



(b) Ships location in Disko bay marked with red dots.

Figure 5.11: Ships are tracked by AIS and their location is shown in subfigure b). The timestamp where these ships were logged 2020/08/11 - 10:08

In figure 5.11b the small red dots were ships from the AIS data. The tracks were matched with a Sentinel-1 image from 2020/08/11 – 10:08. For cities or settlements with many ships it can be necessary to mask out the harbour because the captain of a ship has to turn off the AIS signal and this is not always done. In figure 5.11 Ilulissat has been masked out. There were 20 ships remaining but not all could be found in the SAR image. Some of the ships dock in a harbour at other settlements and some of the ships were too small to be detected.

5.7 Generating Ship Iceberg Images

Candidates for ships and iceberg were found using the CFAR algorithm. In one Sentinel-1 SAR image up to 500 targets were located. With the AIS data the candidates were split up into one of the two classes, ship or iceberg. The same variables as for the Kaggle data were saved. As an additional variable the backscatter normalisation degree was saved

for both ships and icebergs, further the ship type, length, width and speed over ground were saved for ships.

6 Results

These results have been made in Python. In this thesis, large satellite images (30000×25000 pixels) have been used. The processing of these images were very computational expensive. Many custom solutions had to be invented because a powerful computer was not provided. The workflow for preprocessing Sentinel 1 images and merging the bathymetry model with the Sentinel 1 images have been done in terminal/bash scripts. The CNNs have been trained on DTUs HPC cluster. The networks have been trained on a Tesla V100 GPU with the deep learning library Pytorch. The HPC cluster was not always stable and sometimes it was fully occupied. Projects such as this could be considered "pay to win" because of the BIG DATA aspect in it. It is strongly advised that a powerful computer is available for future students.

6.1 Backscatter dependence on incidence angle

The backscatter dependence on incidence angle is now described.

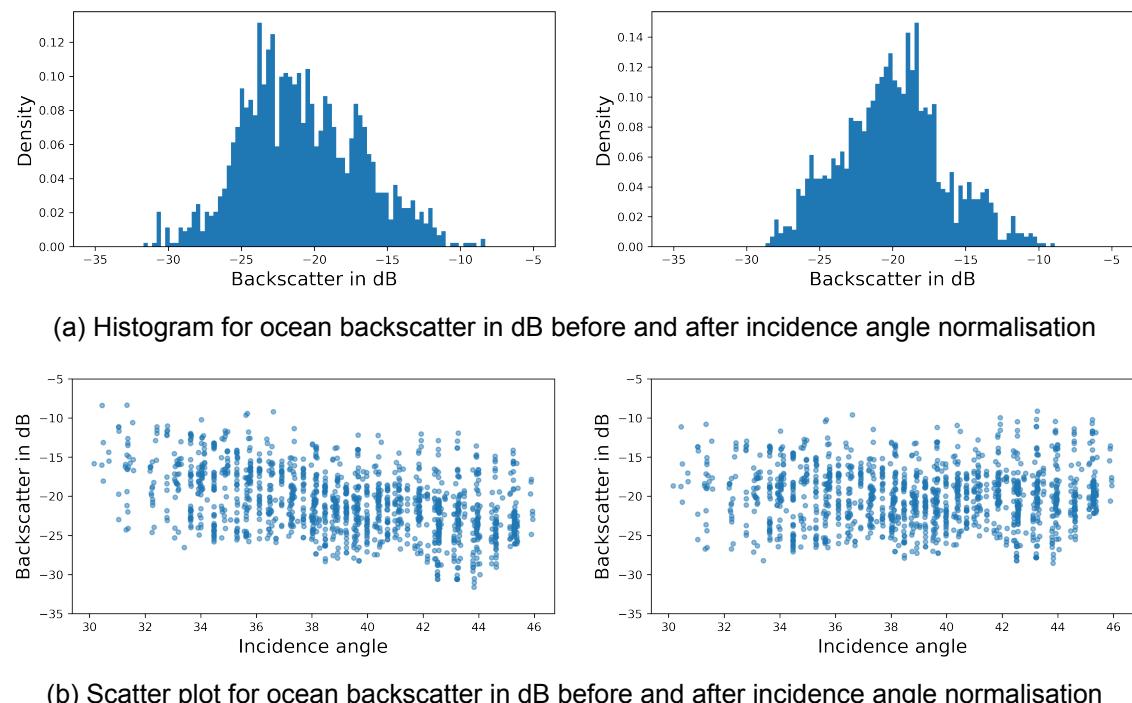
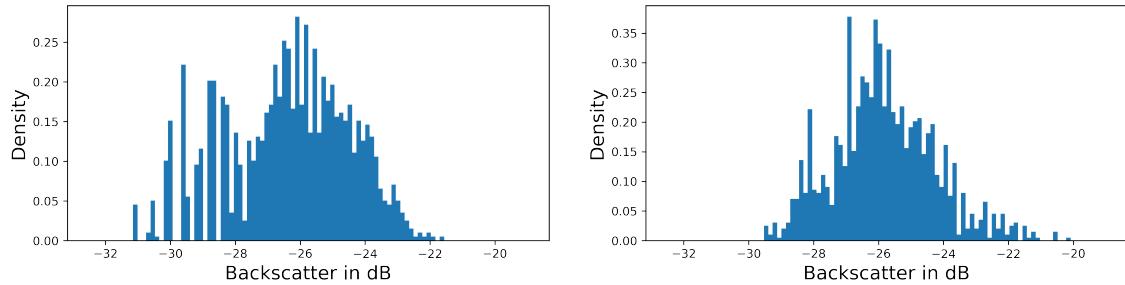
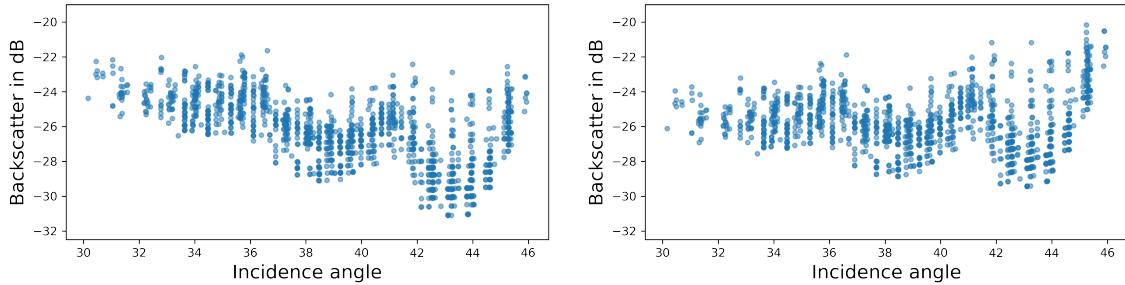


Figure 6.1: Mean ocean backscatter in dB for HH polarisation for the 1600 training images in Kaggle Statoil/C-CORE dataset. The left image is before the normalisation correction of the backscatter incidence angle. The normalisation coefficients $n = 7.52$



(a) Histogram for ocean backscatter in dB before and after incidence angle normalisation



(b) Scatter plot for ocean backscatter in dB before and after incidence angle normalisation

Figure 6.2: Mean ocean backscatter in dB for HV polarisation for the 1600 training images in Kaggle Statoil/C-CORE dataset. The left image is before the normalisation correction of the backscatter incidence angle. The normalisation coefficients $n = 4.62$

In figure 6.1 and 6.2 the Kaggle Statoil/C-CORE training data, the two backscatter incidence angle normalisation coefficients have been calculated to be 7.52 for HH and 4.62 for HV.

The modified Unet proposed in section 5.1.5 has been trained with the original data. The input data has been corrected for the incidence angle. The two nodes for incidence angle and target size was not included in the fully connected part of the network. The network has been trained 5 times in each mode.

The two trained convolutional neural network with and without incidence angle normalisation are evaluated on the test data. The Kaggle score for the test data fluctuates and depends on the model that were evaluated. The Kaggle score should be very similar to the validation loss in figure 6.3c and 6.4c. Here it was assumed that the same coefficients for incidence normalisation could be used on the test data.

In the network without the incidence angle normalisation the Kaggle score was estimated to: 0.219.

In the network with the incidence angle normalisation the Kaggle score was estimated to: 0.237.

Similar plots as figure 6.1 and 6.2 are plotted for the test data.

Model	Without incidence angle normalisation	With inc angle norm
Kaggle test score	0.219	0.237

Table 6.1

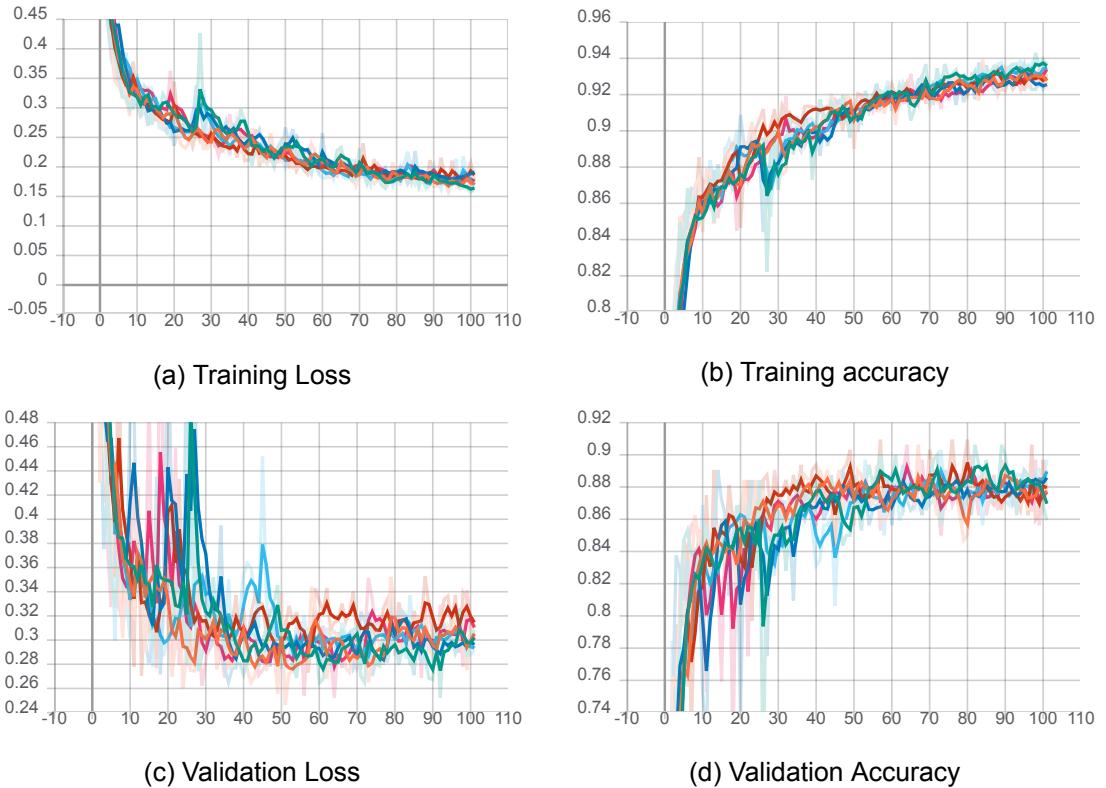


Figure 6.3: Modified Unet trained without incidence angle normalisation.

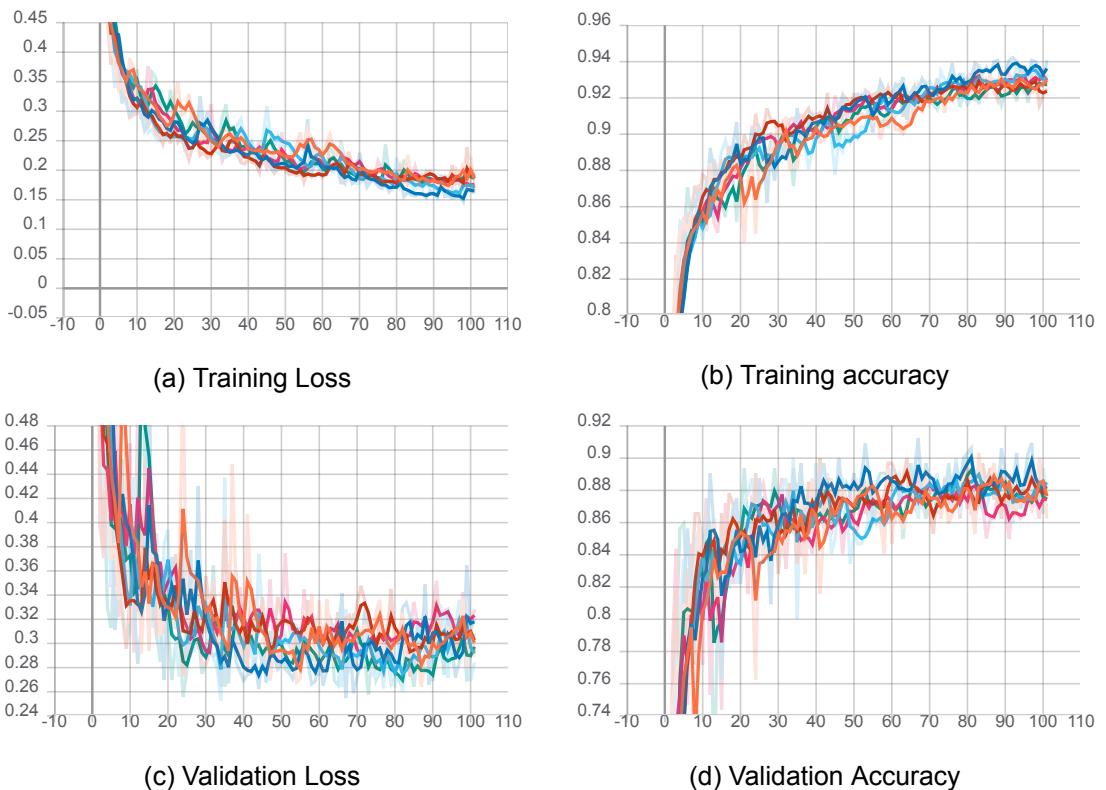
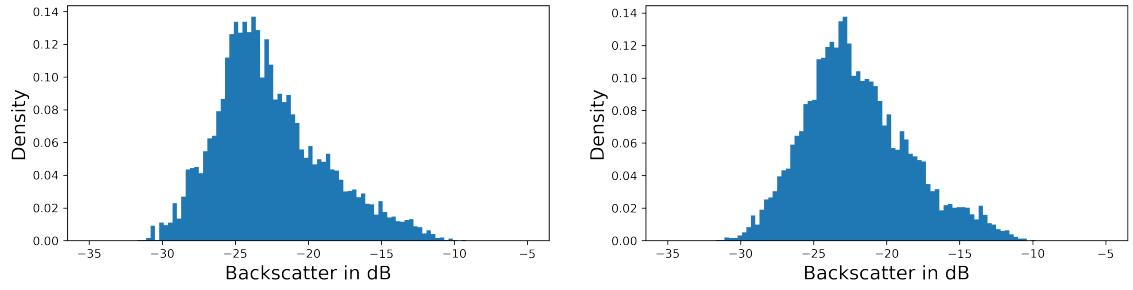
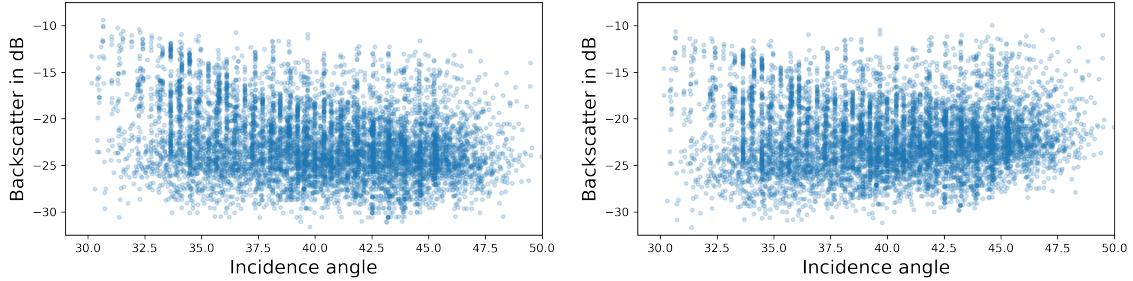


Figure 6.4: Modified Unet trained where incidence angle normalisation is included.

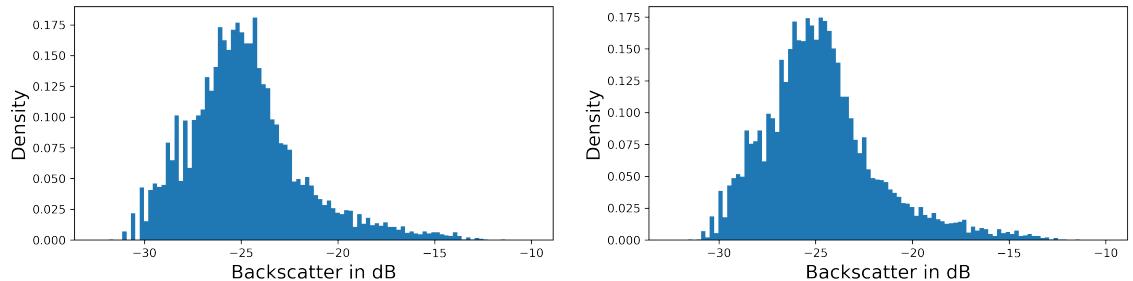


(a) Histogram for ocean backscatter in dB before and after incidence angle normalisation

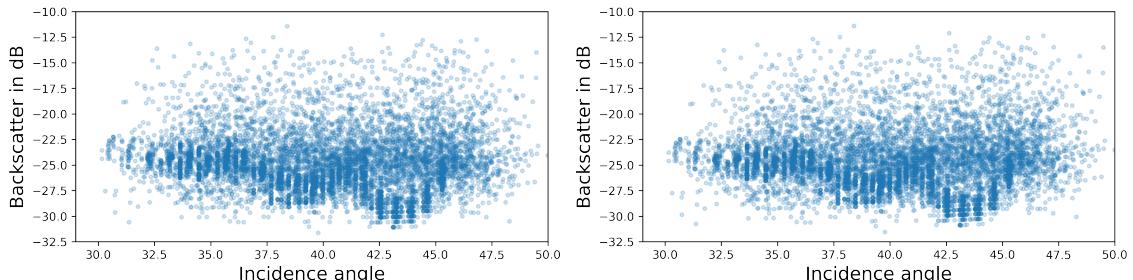


(b) Scatter plot for ocean backscatter in dB before and after incidence angle normalisation

Figure 6.5: Mean ocean backscatter in dB for HH polarisation for the 8400 test images in Kaggle Statoil/C-CORE dataset. The left image is before the normalisation correction of the backscatter incidence angle. The normalisation coefficient for HH is $n = 3.47$



(a) Histogram for ocean backscatter in dB before and after incidence angle normalisation

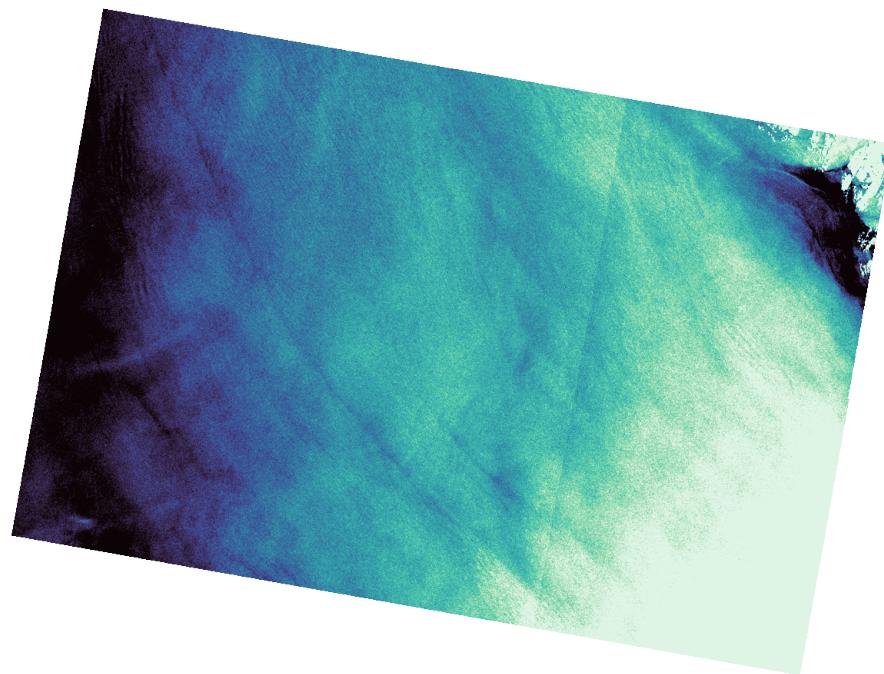


(b) Scatter plot for ocean backscatter in dB before and after incidence angle normalisation

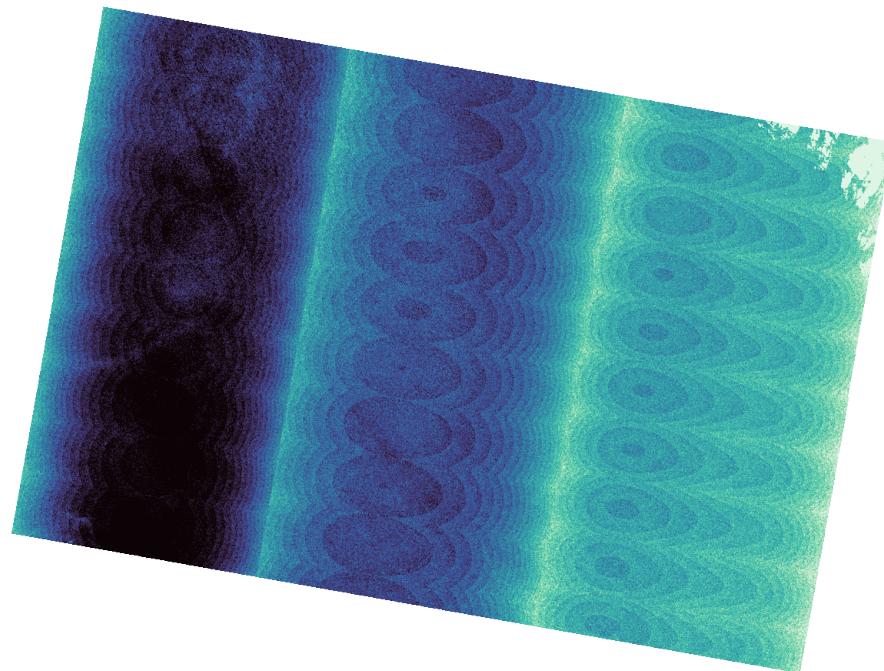
Figure 6.6: Mean ocean backscatter in dB for HV polarisation for the 8400 test images in Kaggle Statoil/C-CORE dataset. The left image is before the normalisation correction of the backscatter incidence angle. The normalisation coefficient for HV is $n = 0.57$

For the HV polarisation a non linear trend in the ocean backscatter versus the incidence angle can be observed. This is very clear in figure 6.2b and can also be seen in figure 6.6b.

To see if this is common for Sentinel 1 SAR images an image from the ocean close to Nuuk was visualised.



(a) HH polarisation



(b) HV polarisation

Figure 6.7: Sentinel 1 SAR image of the ocean close to Nuuk of HH and HV polarisation in dB. In HV polarisation there is clear visual artifacts from thermal noise in the subswaths and by the bursts. The image is from 2020/08/11 - 10:09.

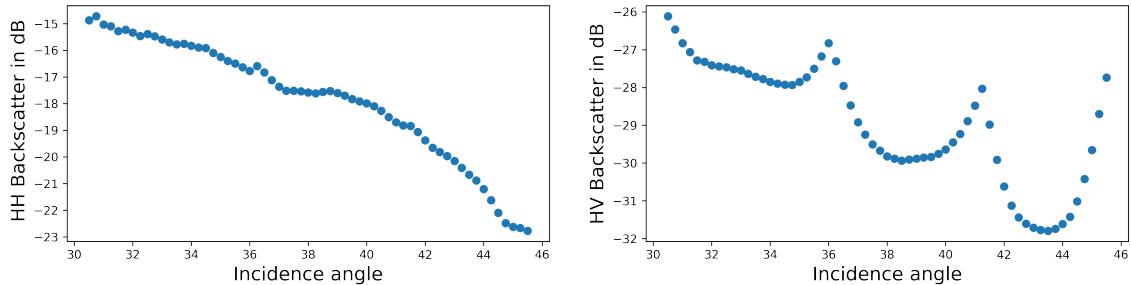


Figure 6.8: Incidence angle vs backscatter in dB for HH and HV polarisation. HH polarisation is close to linear. In HV there is artifacts from each subswath. The backscatter incidence normalisation coefficients are 8.27 for HH and 4.76 for HV.

The backscatter incidence normalisation coefficients changes from scene to scene. Therefore it is not possible to correct for the incidence angle in the Kaggle Statoil/C-CORE dataset. There are unwanted effects from thermal noise over the ocean in HV polarisation. See table 6.2 for a summation of the backscatter incidence normalisation coefficients:

data	HH coefficient	HV coefficient
Kaggle Train	7.52	4.62
Kaggle Test	3.47	0.57
Nuuk image	8.27	4.76
Disko bay August 2020	3.86 to 15.38	3.34 to 6.15

Table 6.2

In table 6.2 coefficients are included from Disko bay area in Greenland. The coefficients from Disko bay are shown as a range as they have been derived from multiple Sentinel 1 scenes.

The incidence angle normalisation coefficient changes from scene to scene. Therefore the incidence angle normalisation for all the following trained network are not included.

6.2 Transfer Models

There are many different convolutional network architectures. Three different architectures were adopted, VGG, Resnet and Densenet. Each type of the top level networks have different sub networks. The sub networks for each architecture tested in this study were:

- VGG11, VGG13, VGG16 and VGG19 all with batch normalisation
- Resnet18, Resnet34, Resnet50, Resnet101 and Resnet152
- Densenet121, Densenet161, Densenet169 and Densenet201

All these models can be found at¹. The models varies in a number of layers and length of the network. The number in the network name indicates the amount of weight layers. Weight layers are both convolutional and fully connected layers. The accuracy- and loss curves for training and validation data were plotted for the each top level network. The validation loss curves were plotted for each sub model. All these models starts with a learning rate of 1×10^{-4} and have reduced learning rate on plateau.

6.2.1 VGG

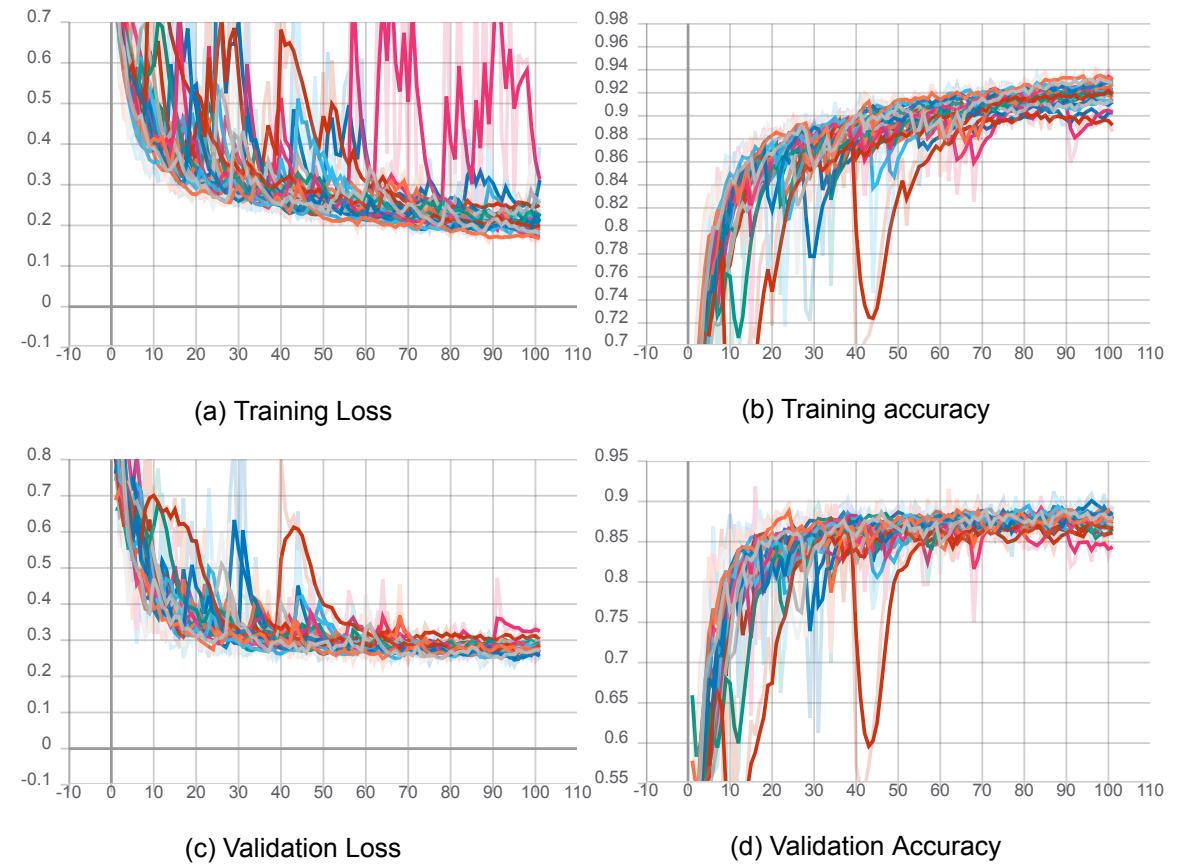


Figure 6.9: VGG

¹<https://pytorch.org/docs/stable/torchvision/models.html>

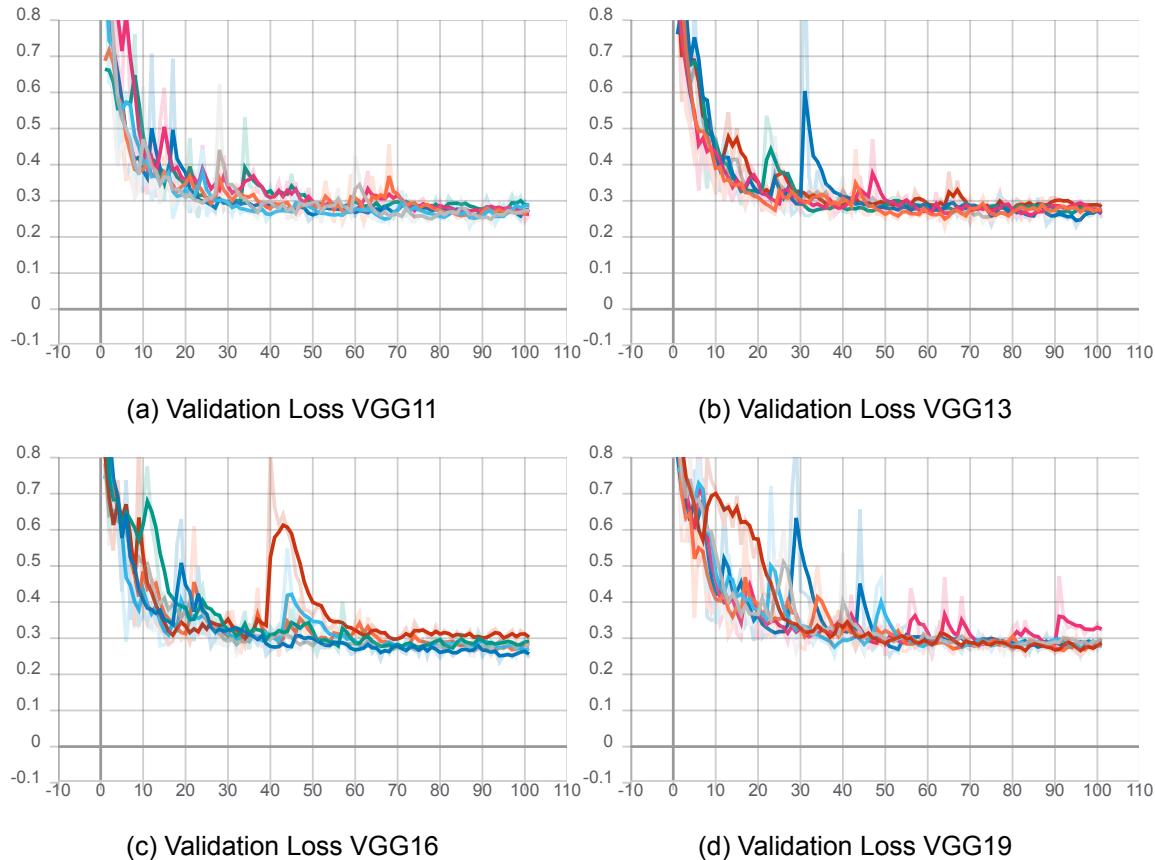


Figure 6.10: VGG

The four VGG models, VGG11, VGG13, VGG16 and VGG19, have been trained on the Kaggle Statoil/C-CORE training data. The same 20% of the training data is used as validation for each model. A validation loss curve for each model can be seen in figure 6.10. Curves for training-loss, accuracy and validation loss and accuracy can be seen in figure 6.9. In figure 6.9 all four VGG models are displayed in the same plot. For these figures shown here, no significant difference between the four models could be observed.

The four VGG models was evaluated after 60 epochs on the Kaggle test dataset and score is given for each model in table 6.6.

Model	VGG11	VGG13	VGG16	VGG19
Kaggle test score	0.232	0.24	0.245	0.246

Table 6.3

6.2.2 Resnet

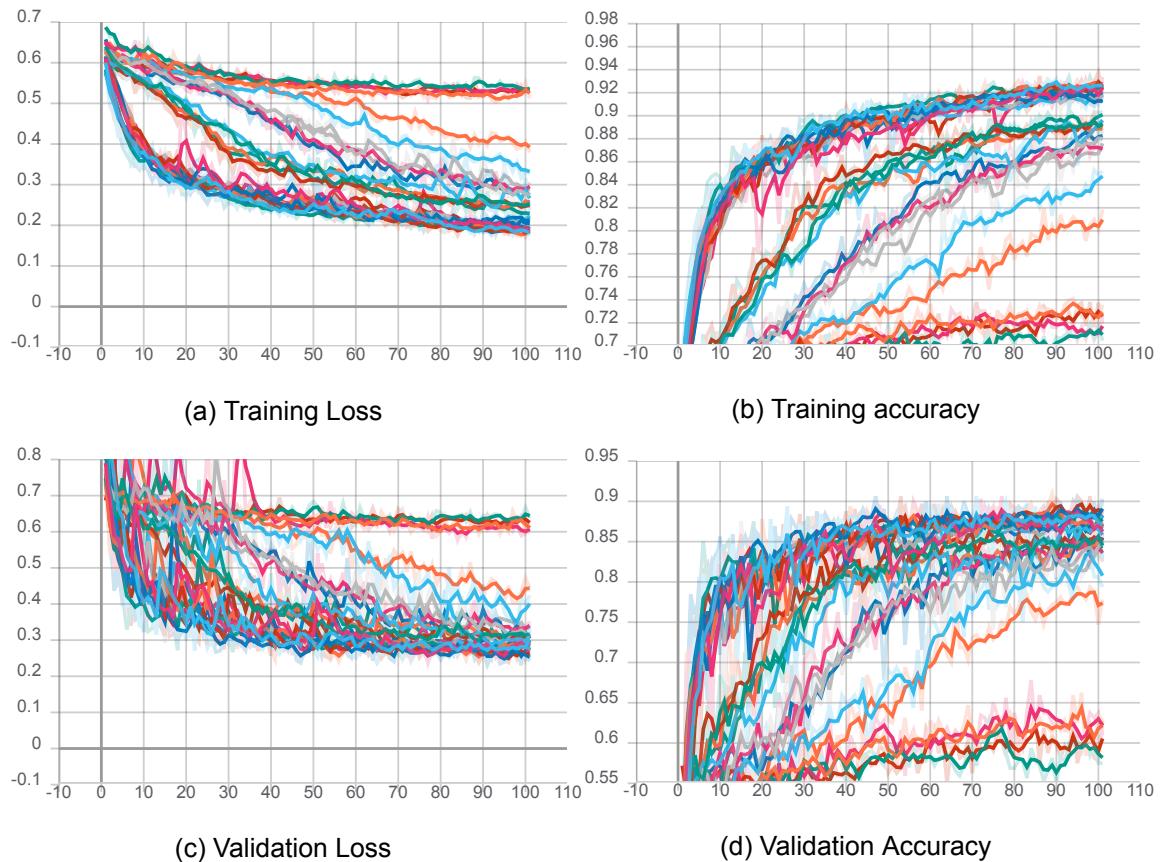


Figure 6.11: Resnet

The five Resnet models, Resnet18, Resnet34, Resnet50, Resnet101 and Resnet152, have been trained on the Kaggle Statoil/C-CORE training data. The same 80% 20% training- validation split have been used for each model. A validation loss curve for each model can be seen in figure 6.12. Curves for training- loss, accuracy and validation-loss and accuracy can be seen in figure 6.11. In figure 6.11 all five Resnet models are displayed in the same plot. There is a significant difference between each model. Only Resnet18 and Resnet34 are comparable in these curves. In figure 6.11 and 6.12 it can be seen that the Resnet models with more weight layers (convolutional and fully connected layers) take longer time to train. In figure 6.12 there is a shift in when the validation loss stagnates and no longer improves.

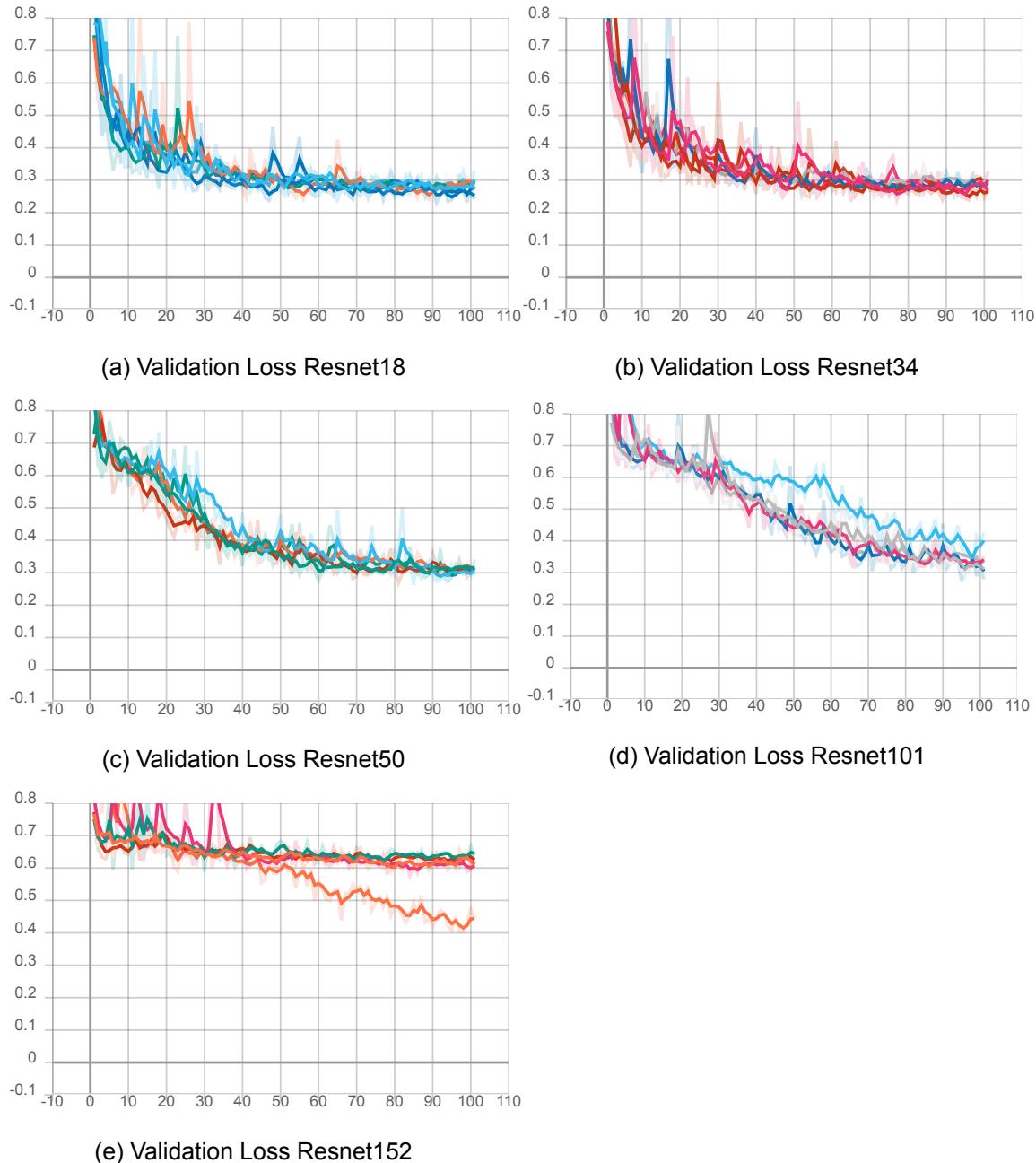


Figure 6.12: Resnet Validation Loss

Resnet 18 and 34 are evaluated after 60 epoch. Resnet 50, 101 and 152 are evaluated after 90 epoch. The scores are evaluated on the Kaggle test dataset and a score is given for each model in table 6.4.

Model	Resnet18	Resnet34	Resnet50	Resnet101	Resnet152
Kaggle test score	0.216	0.284	0.281	0.282	0.574

Table 6.4

6.2.3 Densenet

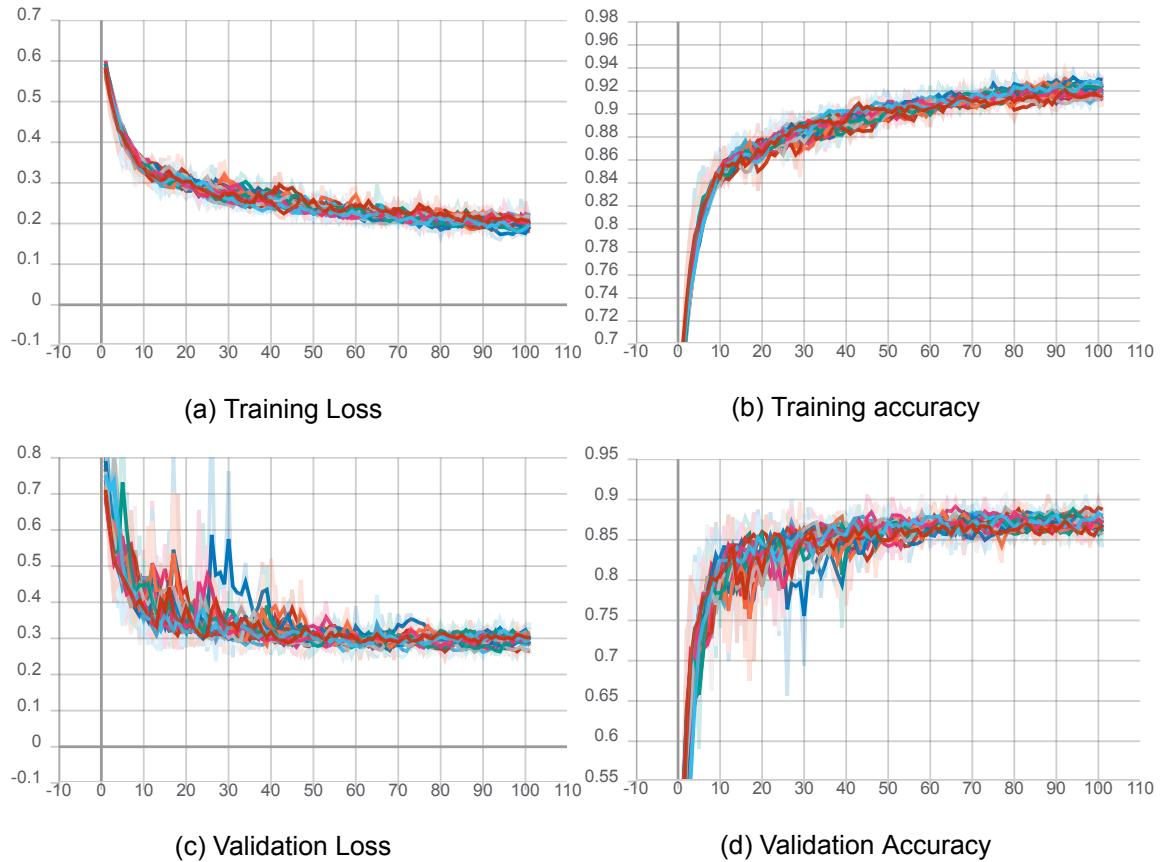


Figure 6.13: densenet

The four Densenet models, Densenet121, Densenet161, Densenet169 and Densenet201, have been trained on the Kaggle Statoil/C-CORE training data. The same 20% of the training data is used as validation for each model as the previous transfer models. A validation loss curve for each model can be seen in figure 6.14. Curves for training loss, accuracy and validation loss and accuracy can be seen in figure 6.13. In figure 6.13 all four Densenet models were displayed in the same plot. These figures illustrates that there were no significant difference between the four models. Even though the models have many weight layers the skip connections avoid the problem with slow training as shown with the long Resnet models.

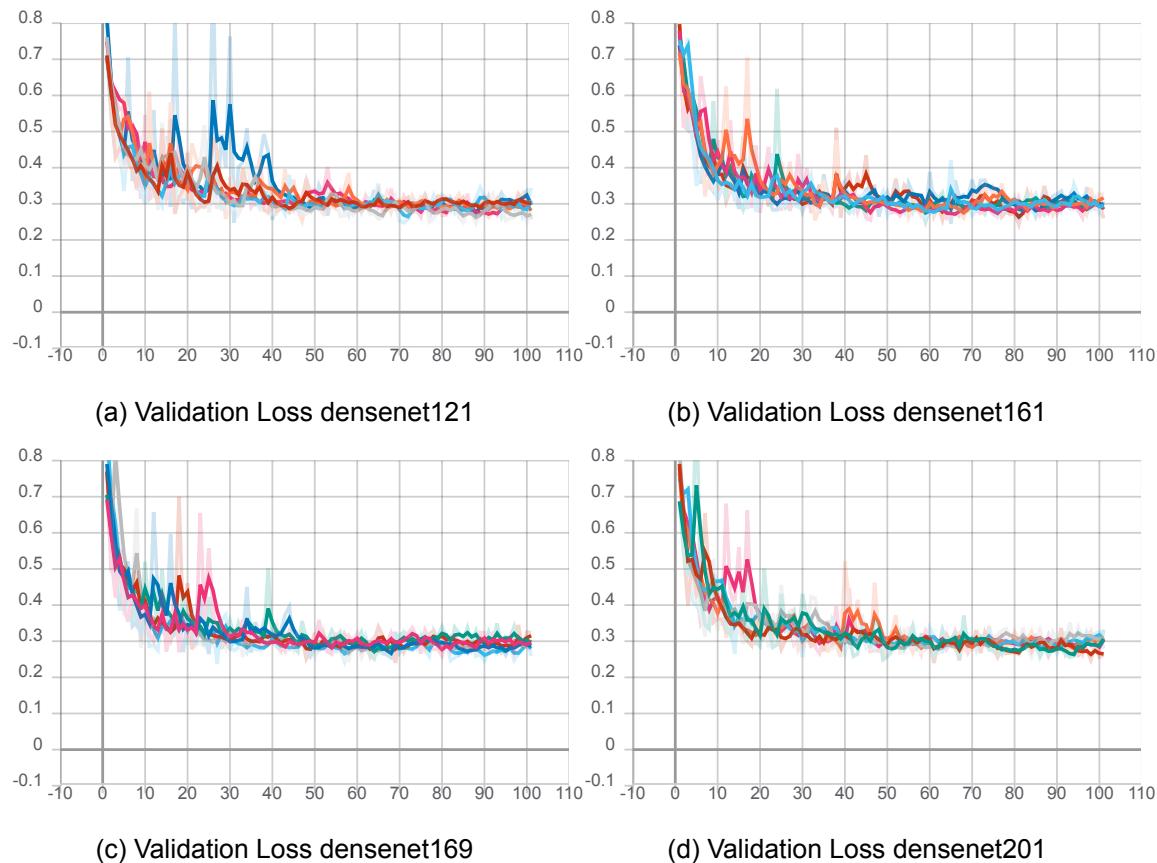


Figure 6.14: densenet Validation Loss

All four models are evaluated after 60 epoch. The scores are evaluated on the Kaggle test dataset and a score is given for each model in table 6.5.

Model	Densenet121	Densenet161	Densenet169	Densenet201
Kaggle test score	0.218	0.220	0.212	0.214

Table 6.5

6.3 Custom data

Sentinel 1 scenes from the Disko bay and Nuuk area in Greenland have been processed. A total of 2891 icebergs have been located in the Disko bay from August 2020. 62 ships have been located in the SAR images from the Disko bay and Nuuk area from August 2020.

3 icebergs and ships are visualised below.

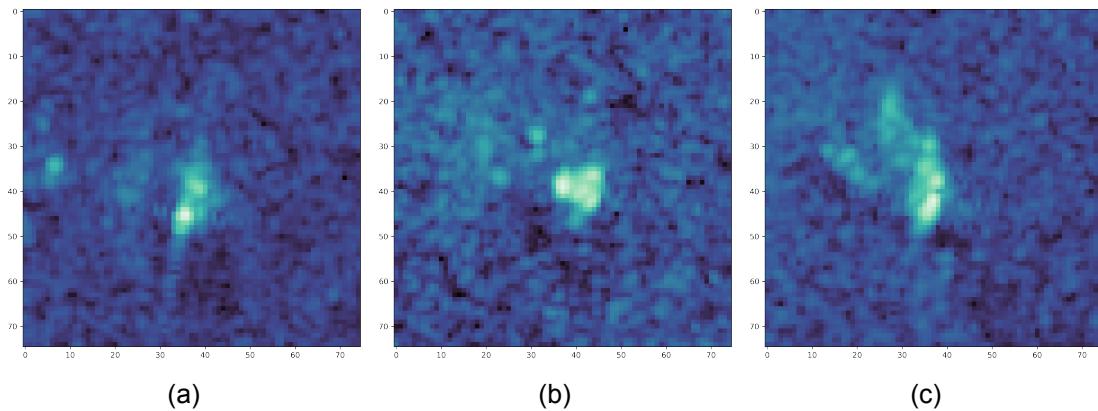


Figure 6.15: Three different iceberg shown for HH polarisation. The three icebergs are from Disko bay August 2020.

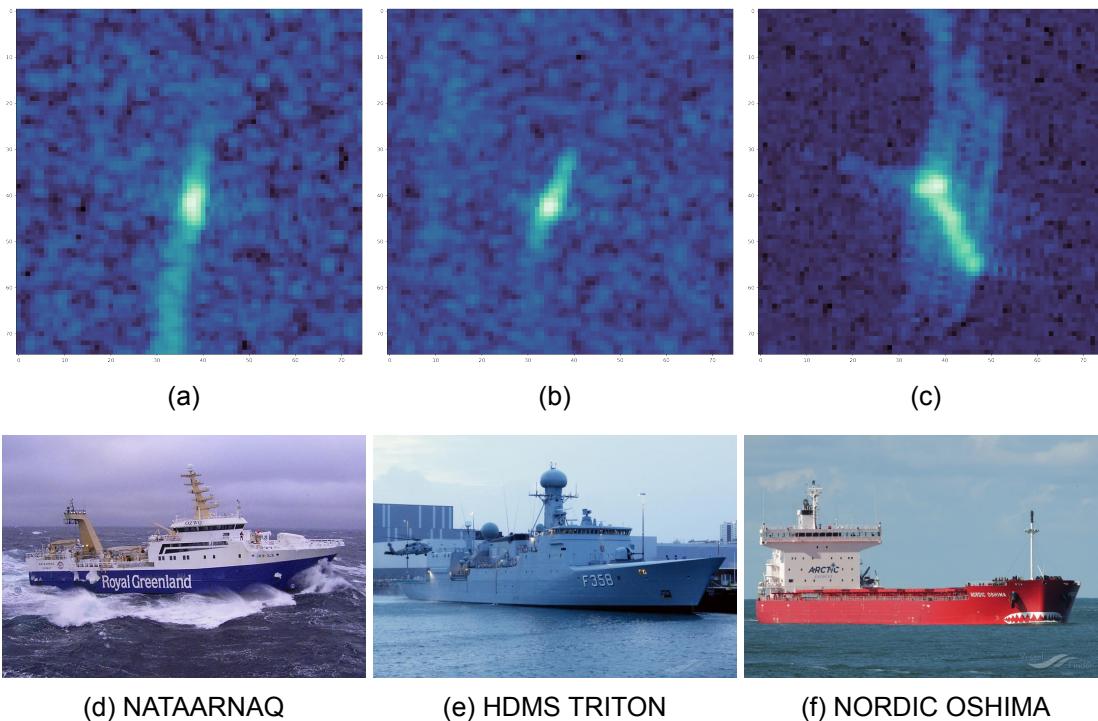


Figure 6.16: Three ships are located near Nuuk that can be seen in the SAR image. The MMSI number from the AIS data is matched so sub figure a) is the same as d), b=e and c=f. In c) sidelobes can be seen from multiple places in the ship due to of very bright scatterers.

6.4 Pseudo labelling and Size

To make the network perform better more data were needed. 2891 icebergs have been found in the Disko bay from August 2020. In august 2020 only 62 ships have been located in the Disko bay and Nuuk area. The dataset is now very unbalanced and this is not recommendable. Neural networks favours the class that is occurring the most. To re balance the dataset pseudo labelling were proposed.

The Kaggle test dataset has 8400 unlabelled ships and icebergs. The labels of these ships and icebergs were predicted with my trained convolutional neural network. An estimate of whether the target is a ship or iceberg are returned. All predictions which have an accuracy above 95% are pseudo labelled. 2498 ships and 1055 icebergs are predicted with above 95% accuracy. The size of these ships and icebergs are estimated.

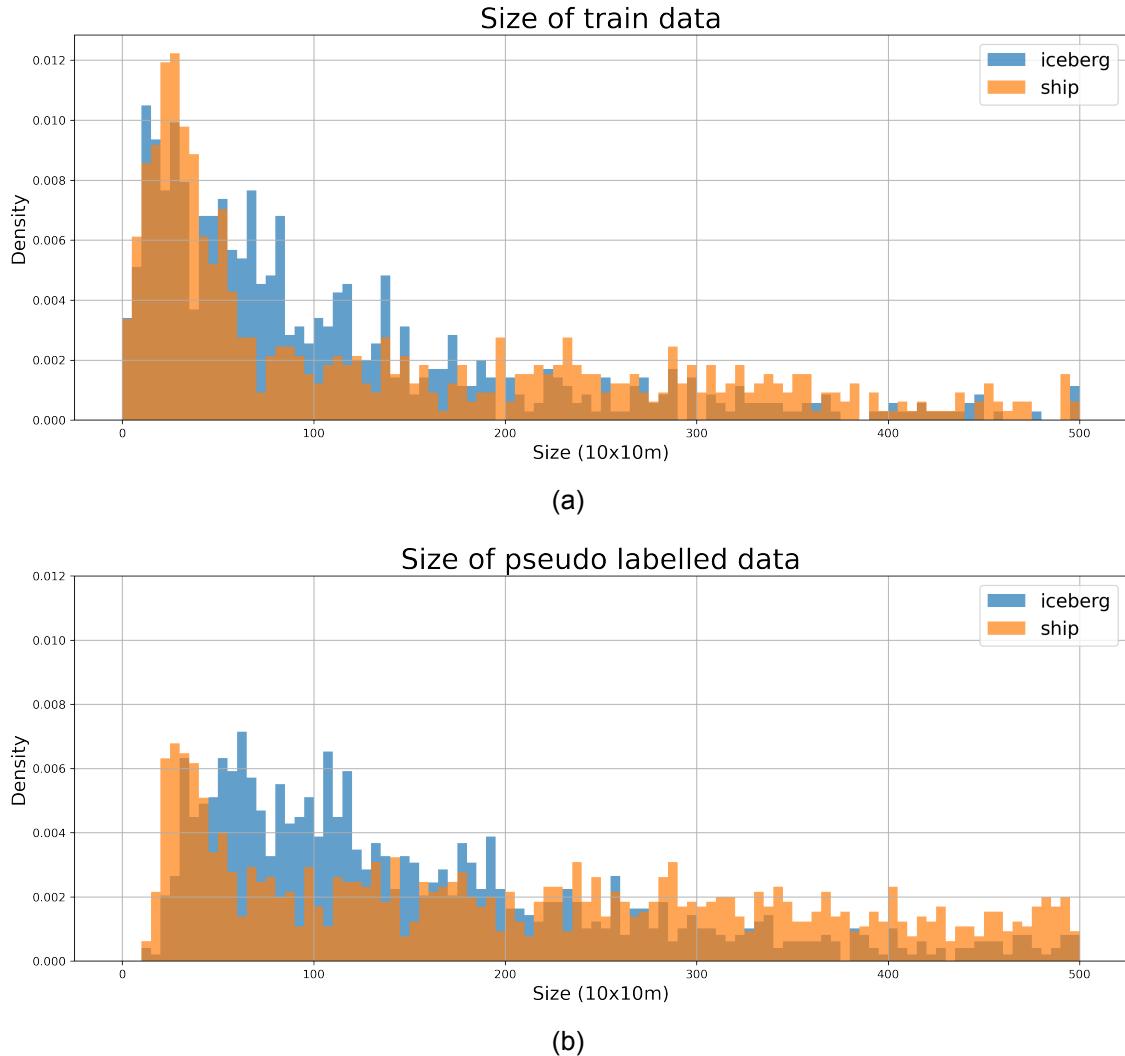


Figure 6.17: Estimated size of the training data and the pseudo labelled test data. The size is estimated from the combined HH+HV band with a threshold of 0.04.

All estimated ships (2498) from pseudo labelled Kaggle Statoil/C-CORE test data and an equal number of icebergs from tracked in the Disko bay area used for further training.

6.5 Extended model and custom data prediction

The new training set includes the training and pseudo labelled test data from Kaggle Statoil/C-CORE and icebergs from the Disko bay. A total of 6700 images are available in the extended dataset.

A good starting point for the learning rate is 1×10^{-4} . For higher learning rates than 1×10^{-4} the network was unable to learn from the start of the training. The training has included Reduce Learning Rate On Plateau. It will therefore not make sense to optimise this parameter further by searching for lower learning rates. It was tested whether L2 Weight decay regularisation should be included or not and the size of the parameter λ , see section 2.3.1.

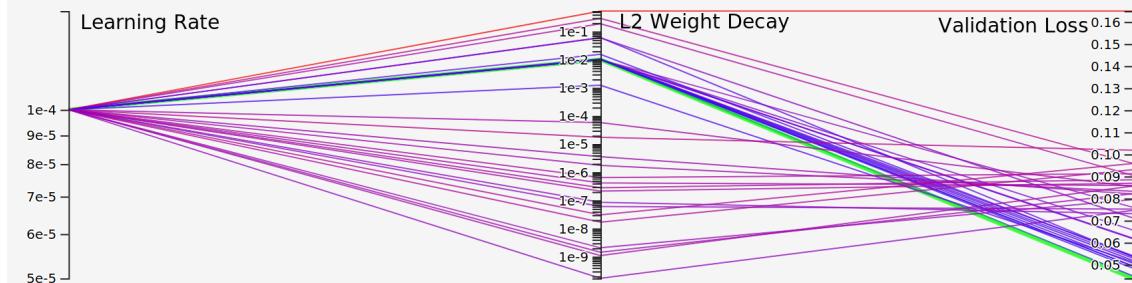


Figure 6.18: Hyperparameter optimisation for L2 weight decay regularisation

From Figure 6.18 the optimal L2 weight decay regularisation parameter was estimated to be $\lambda = 1 \times 10^{-2}$.

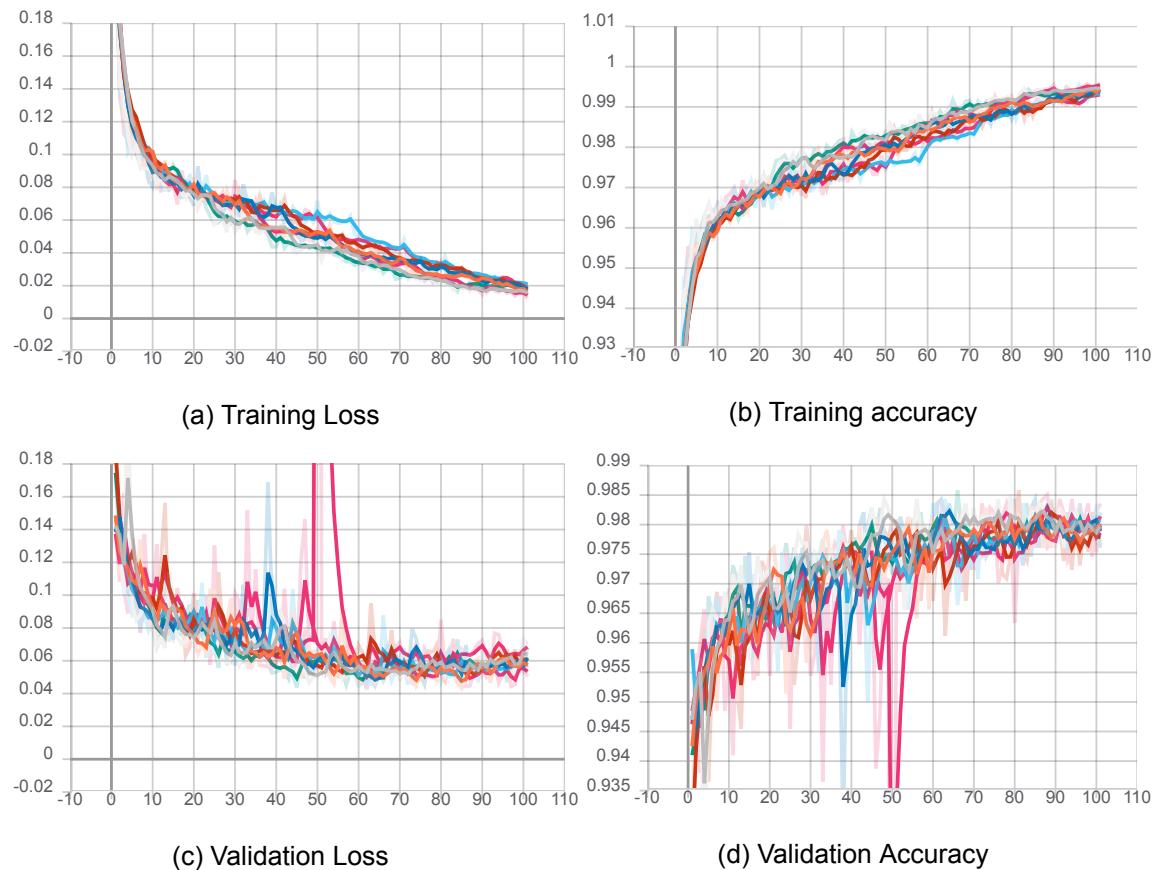


Figure 6.19: Training and validation curves for modified Unet with extended dataset.

Two of these models are evaluated on the Kaggle test score after 60 epoch.

Model	model 1	model 2
Kaggle test score	0.2119	0.2120

Table 6.6

The two models are evaluated on the custom ships and remaining icebergs. 291 icebergs and 62 ships are evaluated. With binary classification 1 indicate that the target was predicted as an iceberg and 0 indicate the target is predicted as a ship.

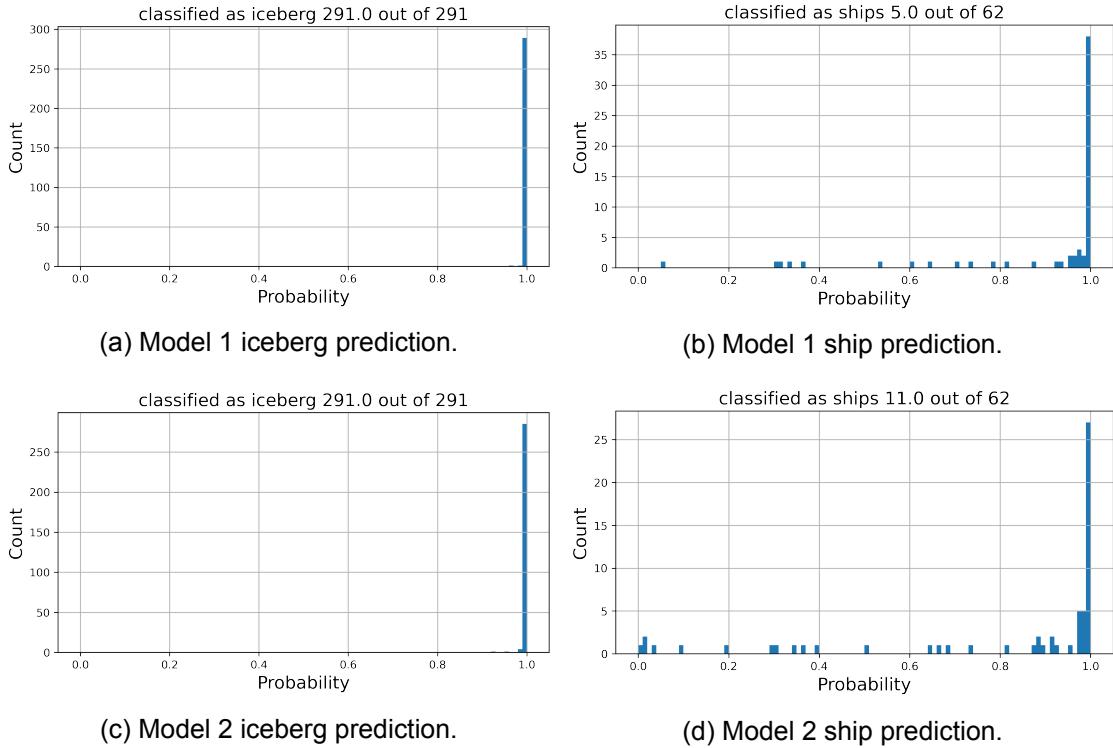


Figure 6.20: Custom icebergs and ships prediction with modified Unet trained on the extended dataset.

As can be seen in figure 6.20, everything are classified as icebergs. A model has been saved for every 5 epoch and 5 models have been trained for the extended dataset. From epoch 30 to epoch 100 all these models are evaluated to see if another model better predicts the custom data.

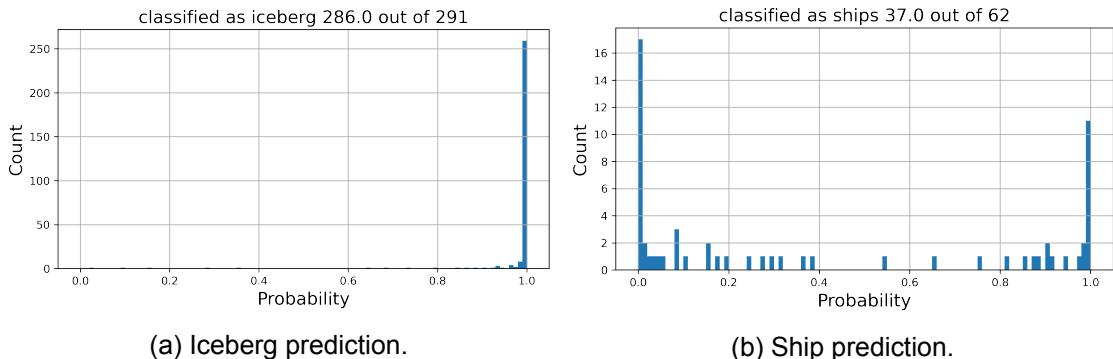


Figure 6.21: Custom icebergs and ships prediction with modified Unet trained on the extended dataset. The best network to predict the extended data.

The model in figure 6.21 was evaluated after 35 epoch. The model still overfits the data for icebergs but now classifies the ships with 60% accuracy. The Kaggle score was calculated for this model and was found to be 0.407.

The Custom ships and icebergs were evaluated on the modified Unet model without the

extended dataset and the best transfer model. The modified Unet and Densenet169 was evaluated after 60 epoch.

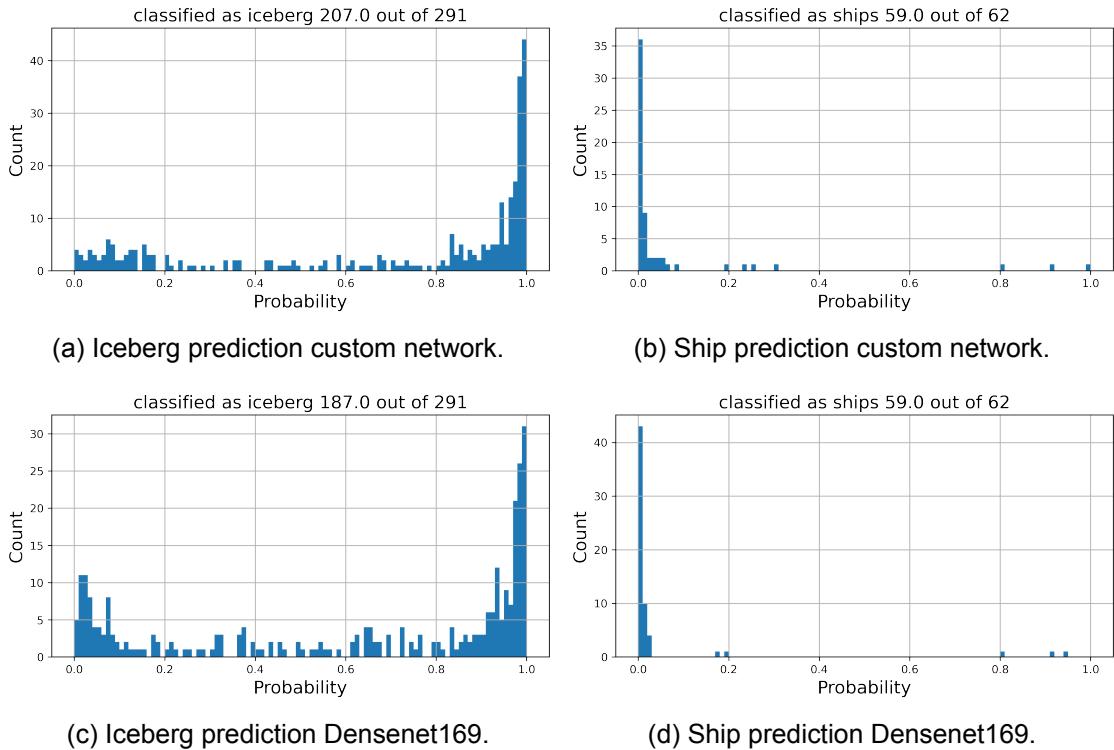


Figure 6.22: Custom icebergs and ships prediction with modified Unet and Densenet169 trained on the original Kaggle training data.

The trained network with the non extended dataset has trouble classifying icebergs. The best result with the custom network can classify icebergs correctly 207 out of 291 times, i.e with a 71% accuracy.

6.5.1 Ensemble network

An ensemble of the 8 best models were proposed. The 8 models were:

- Custom model with modified Unet, figure 6.3
- Resnet18
- Densenet 121, 161, 169 and 201
- Two custom models with modified Unet trained on extended dataset, figure 6.19
- The Kaggle score for these models varies between 0.212 to 0.220

A mean and median ensemble were calculated from the individual models.

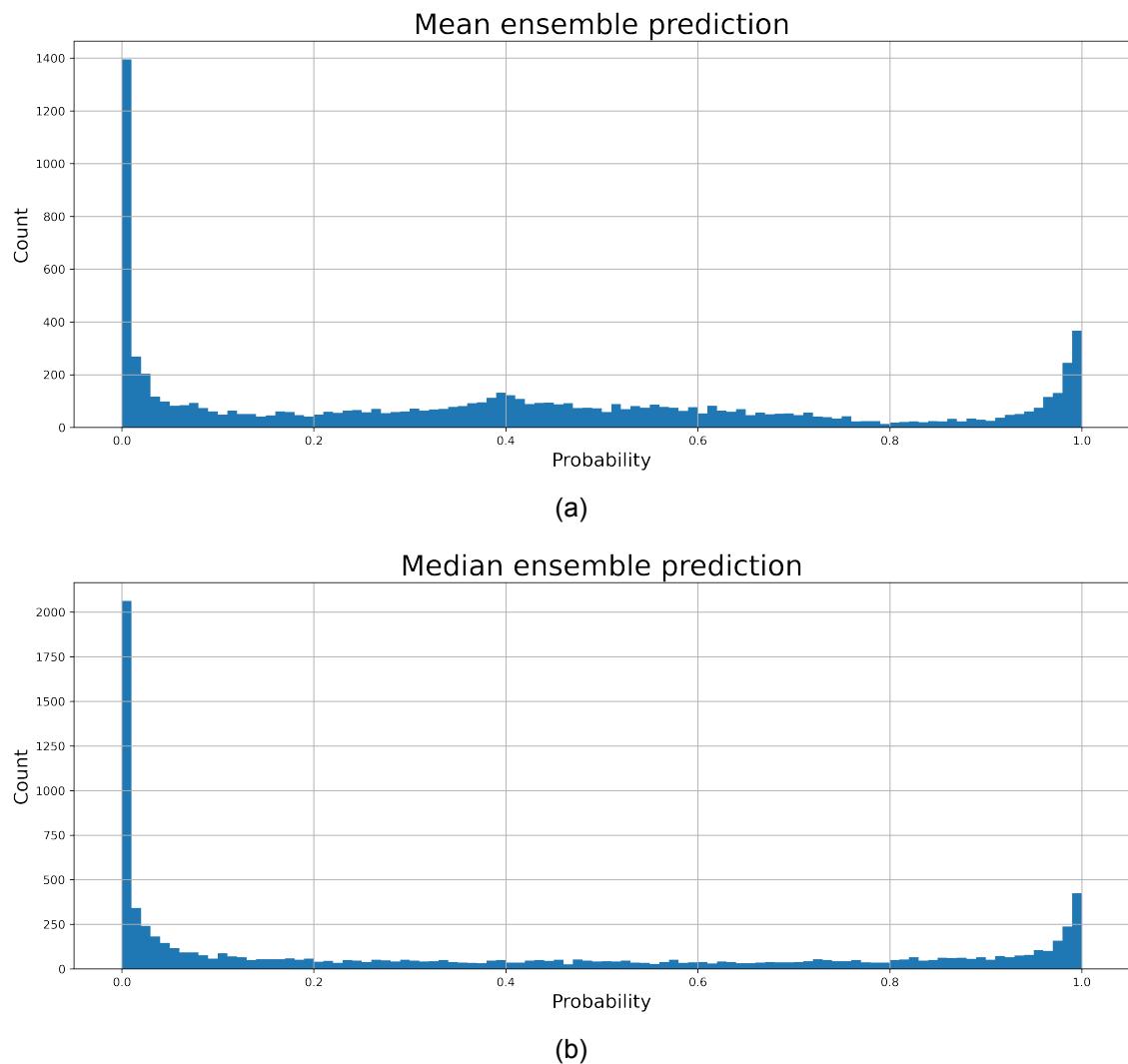


Figure 6.23: Consist of a prediction of 8400 ships and icebergs from Kaggle test set. 0 is ship and 1 is iceberg. 8 models were combined to a mean (a) and a median (b) ensemble.

	Mean ensemble	Median ensemble
Kaggle test score	0.188	0.194

Table 6.7

7 Discussion

A number of methods have been investigated to find the optimal configuration to be able to distinguish between ships and icebergs.

7.1 Preprocessing steps

The Kaggle Statoil/C-CORE dataset has no information about the processing steps used and how the ships/icebergs were located. It is assumed the majority of the proposed preprocessing steps in section 5.3 have been applied. Looking at the images it can be seen with certainty that Range Doppler Terrain Correction has been applied. Very intense backscatter leaves artifacts in the image as can be seen in figure 7.1.

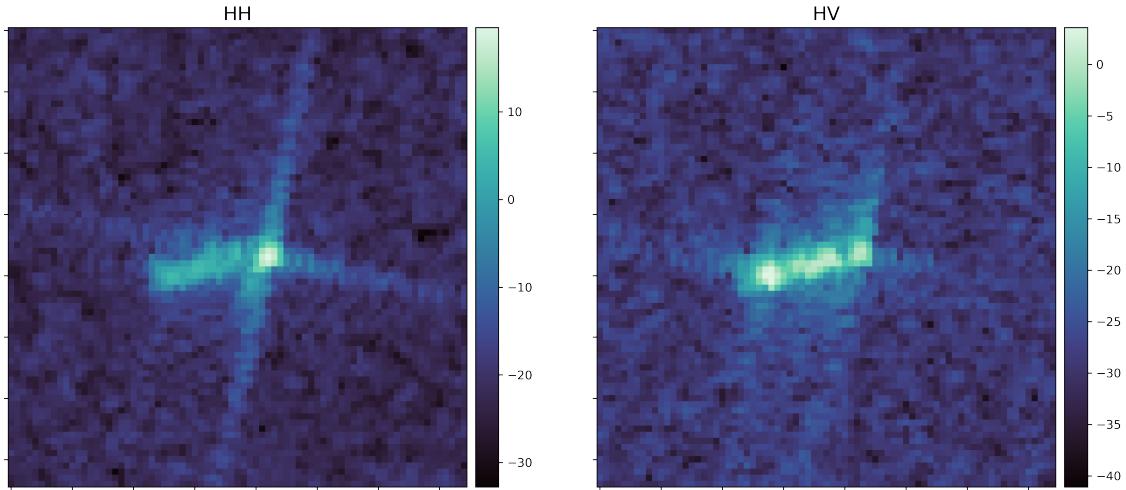


Figure 7.1: Ship in Kaggle Statoil/C-CORE training dataset. In the HH band there are stripes called sidelobes with increased backscatter. The ship acts as a "corner reflector" which is likely to cause sidelobes. The sidelobes are artifacts from the strong backscatter.

Strong reflection from ships e.g. due to corner reflections often lead to sidelobes. The sidelobes from a corner reflector should be parallel with the satellites path. In figure 7.1 the image stripes are rotated. The satellite images have thus been geocoded to another map projection. Further the Range Doppler Terrain Correction is required to get the incidence angle to the target. If a ship acts as a corner reflector and the image is geocoded it can be seen whether the orbit has been ascending or descending. The Ship in figure 7.1 has been acquired from a descending orbit. For reference the orbit can be seen in figure 3.5.

In section 6.1 the influence of the incidence angle on the backscatter has been investigated. It is possible that a small correction has been applied with the default proposed backscatter incidence angle normalisation factor (1-2). The default proposed factor is not enough when normalising over the ocean. In table 6.2 the coefficients varies between HH and HV polarisation. The sentinel 1 scenes used in this study have shown the normalisation coefficients can fluctuate by a large margin. The fluctuation is more dominant in HH polarisation than in HV polarisation.

A backscatter incidence angle normalisation factor of 1-2 corrects the backscatter at most 0.5 to 1 dB for the highest and lowest incidence angles (29° and 46°). The maximum coefficient found at 15.38 corrects the backscatter up to 7.5 dB.

HV polarisation has shown a non-linear dependence for the backscatter on the incidence angle. The non-linear dependence can be seen in figure 6.2b and 6.6b. These images are from multiple scenes. It is even more clear when only one scene is used, see figure 6.7b and 6.8. These non-linear contributions are due to thermal noise from the Sentinel 1 receiver and how the scene is generated. The image in figure 6.7b, 3 subswaths can be seen. These subswaths are from the TOPSAR scanmode for IW images, see figure 3.4. In the proposed preprocessing steps "Thermal Noise Removal" are included. Theoretically this processing step should remove the non-linear contributions from the 3 subswath TOPSAR scanmode.

In HV polarisation there is an additional noise contribution which cannot be seen in the Kaggle Statoil/C-CORE data. In figure 6.7b lines can be seen that are perpendicular to the subswaths. Sentinel 1 collects data in "bursts", [ESA 2021]. The majority of the noise from the bursts should have been removed in a processing called "S-1 TOPS deburst" when the GRD image is generated from a SLC image.

The proposed methods to remove thermal noise and bursts are insufficient for the ocean. The ocean has weak backscatter and therefore the relative noise are greater. To correct for these effects the preprocessing steps should be revisited. The preprocessing should start from IW SLC instead of IW GRD to correct for the bursts. [Lee, Xu, and Clausi 2020] propose a method to improve the thermal denoising. With the limited time of this thesis the ESA default methods have not been improved further.

Ideally the data should be made independent of both incidence angle and sensor noise contributions. It was attempted to make the Kaggle Statoil/C-CORE data independent on incidence angle. The validation loss for the two trained network with and without backscatter incidence normalisation in figure 6.4c and 6.3c showed no significant difference. When the two models were evaluated on the test data, the model where backscatter incidence normalisation was included, perform slightly worse than without. The correction for the incidence angle changes from scene to scene and therefore it is not possible to generalise. The spread of the scatter plots (figure 6.1b and 6.2b) for the training data are more narrow than for the test data (figure 6.5b and 6.6b) where it is harder to see a pattern. The incidence angle is used as an additional parameter in the fully connected part of the convolutional neural network. The network might learn the same patterns from noise with the included incidence angle.

7.2 Transfer models

There are many different predefined convolutional neural networks that can be trained. These models are usually pretrained on ImageNet with 1000 different categories. For further training the pretrained network is usually a good starting point. The pretrained network is able to generalise between many different classes. A quick method is to "lock" all the convolutional layers and only train the fully connected part of the network. This can obtain good and fast results for regular RGB images. Another possibility is to train the entire network but with the starting point of the pretrained network. To use these two methods the images needs to be RGB, with an assumed mean and standard deviation

for each colour band.

The SAR images for ships and icebergs in dB scale could be approximately Gaussian distributed. The distribution for the SAR images were very elongated in the positive direction. The SAR images have only two bands and does not have the same mean and standard deviation as RGB images would have. The pretrained models were therefore not utilised but trained with a pseudo random initialisation.

The three common and well known models, VGG, Resnet and Densenet were trained on the Kaggle Statoil/C-CORE dataset. Each model has 4 to 5 submodels with varying architectures. 13 different models have been trained in section 6.2. For all trained transfer model architectures it was not possible for the validation loss to go lower than 0.25 where all models stagnated.

All VGG network training (figure 6.9 and 6.10) follows the same pattern where the validation loss stagnates after 30 epochs. VGG11 indicates 11 weight layers (convolutional and fully connected layers). VGG 13 has 13 weight layers and so forth. There are no benefits for ship/iceberg discrimination to use more weight layers. The four VGG models have been evaluated after 60 epochs. The Kaggle test score in table 6.6 for VGG shows no significant difference between the models.

The five Resnet models performs very differently. The length of these models varies more than for VGG. The naming scheme is the same as VGG, Resnet50 has 50 weight layers. For networks with many weight layers it takes longer time and hence more iterations were needed to train. Resnet 50, 101 and 152 stagnates either very late (80-100 epochs) or not at all for Resnet152. The Kaggle test score in table 6.4 performs best for the smallest model, Resnet18. The four other Resnet models performed worse than all VGG and Densenet models. Resnet 18 and 34 are evaluated after 60 epoch and Resnet 50, 101 and 152 are evaluated after 90 epochs. The complexity of the data with a binary classification does not suit very long models. In deep learning binary classification is a relative easy task compared to training a model with 1000 different categories. Theoretically the very long Resnet should work better against such problems.

Densenet performed the best out of the three transfer models. All Densenet models follow the same learning and validation curves, figure 6.13. There is no penalty for longer network since the skip connections prevent to not learn the same convolutional layers multiple times. All Densenet models are evaluated after 60 epochs.

For the VGG, Resnet and Densenet models it should be noted that the Kaggle test score is not an absolute measure of how well the model perform. The test score varies dependent on the model and at what epoch number they were evaluated at. For the models where the validation loss has stagnated all these models are similar. By evaluating 4-5 sub models for each type of network a general trend can be seen. In these tests, Densenet has the best overall result with the best score for Densenet169. The score for Densenet is very close for the four models and therefore it is not possible to say if one of the sub models are better. Resnet performs the worst of all the models. The exception was Resnet18 which had similar test score as the four Densenet models. This is likely because, the network

did not include many weight layers and therefore suited the data better.

7.3 Custom data

The purpose of creating a custom dataset is to see if the Kaggle Statoil/C-CORE trained convolutional neural network can be applied on the new data and make a more robust model. Ideally the Kaggle Statoil/C-CORE dataset should not be used. The Kaggle Statoil/C-CORE dataset has several limitations.

- No information about the preprocessing steps
- No information about ship type and length/width
- The data is not corrected for incidence angle

In the Arctic region where HH+HV polarisation are used it was difficult to locate ships due to limited marine traffic. From the Sentinel 1 scenes 2891 icebergs and 62 ships was located from August 2020. If time allowed to generate large enough dataset with ships the classification problem could be split up into ship type categories, i.e multiclass classification. For surveillance it could be advantageous to have an estimate whether a ship was likely a fishing vessel or a military ship. Sentinel 1 images in combination with AIS data gives new possibilities for close to real time tracking. In the arctic region around Greenland activity of interest could be illegal fishing, trespassing or search and rescue. Fishing vessels that perform illegal activity often turn off their AIS signal before they enter a zone where they are not allowed to fish. An example of illegal fishing were from a Chinese fishing fleet entering North Korea maritime waters¹.

7.4 Pseudo labelled data and size

To extend the training dataset more icebergs and ships was required. The icebergs are located in Sentinel 1 scenes used in this study. If no ships were added to the training data it would make the dataset very unbalanced. Deep learning networks are naive and finds the easiest method to get a good score. In a network where 90% of the data belongs to the first class and 10% to the second class it is likely that the network would label everything as the first class.

To get more data pseudo labelling of the test data was proposed. All ships and icebergs where the trained network has a higher probability than 95% of being either of the classes are given a label. In this way 2498 ships and 1055 icebergs were predicted with above 95% accuracy. The network used for pseudo labelling was the modified Unet.

In further training it should be noted that the training and loss curves are **not** comparable with previous results. By pseudo labelling data ships are extracted that are very similar to the ships in the original training set. A new trained network where pseudo labelled ships are included will therefore always have a higher chance of predicting correct.

The estimated size of the ships and icebergs can be seen in figure 6.17. The majority of ships and icebergs in the training set are relative small targets. The size of the estimated pseudo labelled test data is larger than for the training data.

For the convolutional neural network it is not possible to easily distinguish between ship and iceberg if there is less than 20 pixels of the target. This is somewhat expected since there cannot be many structural changes in the image with only a few dominant pixels of the target. Fortunately there is smearing effects on a SAR image from the target so the actual size of the ship or iceberg does not have the same size in a SAR image. The

¹<https://www.nbcnews.com/specials/china-illegal-fishing-fleet/>

Sentinel 1 GRD IW scenes have a spatial resolution of 22×20 m subsampled to 10×10 m. If the actual size of the ships were representative for the SAR only very large ships could be classified. In figure 7.2 the actual ship size can be seen versus the estimated size in the SAR image. The figure is from the 62 ships detected for this study in the Disko bay and Nuuk area.

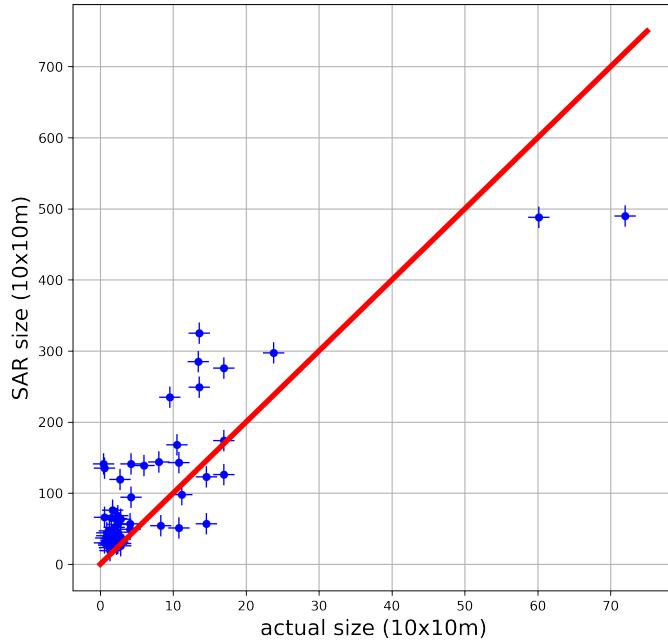


Figure 7.2: The length \times width from the AIS data of the 62 custom ships located in SAR images are compared with the estimated SAR pixel area from equation 5.2 and set to a threshold of 0.04.

The majority of these ships are small fishing vessels. In figure 7.2 it can be seen that the ships size in a SAR image is approximate 10 times greater than the actual size. The sample size are too limited to give a more accurate estimate of the actual size versus SAR ship size. From these results a ship should be approximately longer than 20 m to be classified with a good accuracy in a SAR image.

The threshold of 0.04 is user defined. The cargo ship in figure 6.16c and 6.16f is used as an example to understand the actual size versus estimated SAR size. The cargo ship NORDIC OSHIMA has a length of 225.0 m and a width of 32.0 m. The actual size is 72 (10×10 m) pixels and was the largest ship in figure 7.2. With a threshold of 0.04 the estimated SAR size was 490 (10×10 m). From the AIS data, the speed has been estimated to 10.7 knob or 5.5 m/s at the time the Sentinel 1 image was captured.

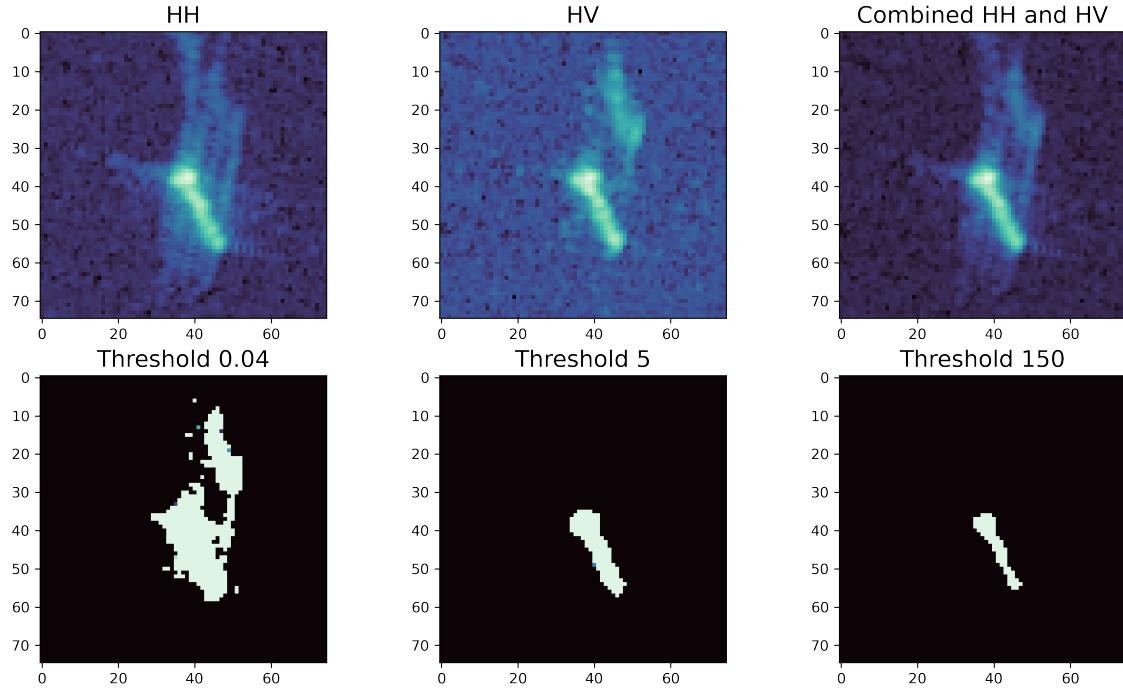


Figure 7.3

In figure 7.3, 3 different thresholds were tested and visualised. The combined HH and HV band in dB scale is shown in the upper right corner. The threshold is estimated on the non dB image. The accommodation house can be seen as a larger structure in the center of the images. The cargo ship is therefore sailing in a south southeast direction. HH band in the upper right corner of figure 7.3, multiple sidelobes can be seen. In HV another phenomena can be seen with increased backscatter. A bright area are shown in the northern azimuth direction of the cargo ship. This phenomena is not sidelobes but likely because of the azimuth displacement and the wakes, [Greidanus 2008]. In [Greidanus 2008] is a similar example shown in Fig. 3. where a offset can be seen in HV polarisation of a moving target. The displacement in the azimuth direction was calculated by 3.4. Without knowing the exact bearing from the image and the sidelobes in HH band it were estimated to be 45° with respect the SAR. The azimuth displacement was estimated to be $\cos(45^\circ) \cdot 5.5 \text{ m/s} \cdot 129.2 \text{ s} \approx 500 \text{ m}$ or 50 pixels. The displacement in the image is roughly 35 pixels. [Greidanus 2008] describes the increased backscatter from the displacement artifacts or ghosting in HV polarisation were located between a large ship and the apex of its wake.

The threshold is a user defined parameter. The threshold applied to estimate the pixel area in figure 7.2 was 0.04. The three tested threshold and the effect on the pixel area were:

Threshold	0.04	5	150
Estimated SAR area	490	131	72

The increased pixel area for a threshold of 0.04 includes both part of the sidelobes in HH and the ghosting in HV. In the subplot in figure 7.3 where the threshold is set to 5 the pixel area matches well with the combined HH and HV image. The estimated area in the SAR is close to twice that of the actual area. The backscatter returned from the ship

to the receiver includes both when the ship is hit first and bounced on the water as well as the other way around. This effectively increases the width of the ship seen from the SAR. Furthermore the poor spatial resolution. If the actual width of the ship should be represented in the SAR image the image should be 3 pixels wide. A threshold of 150 was implemented to match the actual size with the estimated SAR pixel area. Such a high threshold should not be set because the highlighted pixels barely resembles the shape in the SAR image. For all three threshold there are a significant change in the estimated width of the ship but the length remains in the same approximate range.

In section 5.1.3 a threshold of 0.04 was proposed to include targets with smaller pixel area but this was probably set a bit too low. The threshold could be implemented adaptively for smaller and larger ships such that the estimated pixel area better fits the different ships.

Many types of phenomena can occur that can cause increased backscatter near a ship. Figure 6.16a shows a "trail" after the ship. This is probable because of azimuth defocusing after consulting with John Marryman, Head of Microwaves and Remote Sensing DTU. As mentioned in the SAR theory, the SAR assumes targets are stationary. Moving targets and its different artifacts is another subject that has not been fully explored in this master thesis. The main point from the pixel area estimation was to help the neural network to better predict ships and icebergs.

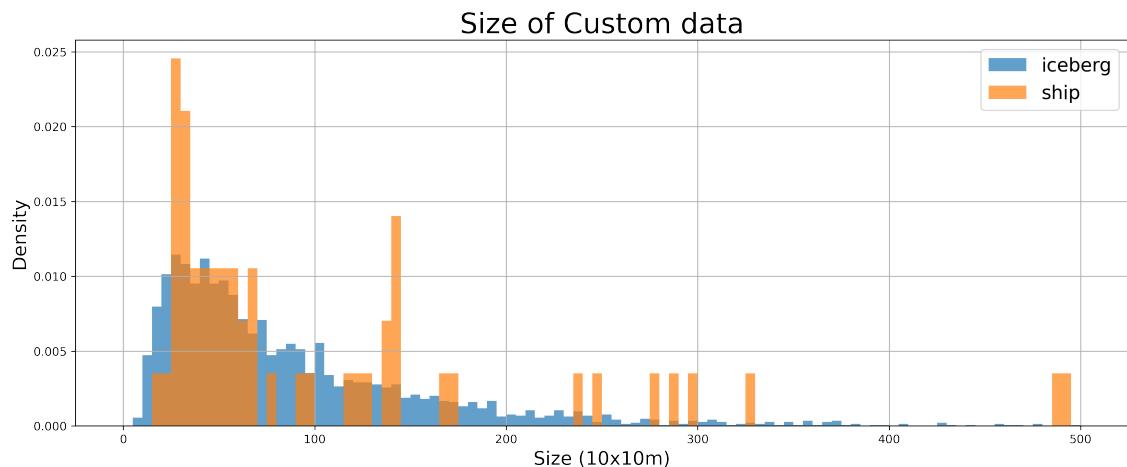


Figure 7.4: Custom ships and icebergs size or pixel area estimated from the combined HH and HV band with a threshold of 0.04.

In figure 7.4 the size density of the 2891 icebergs and 62 ships in the custom data was visualised where a threshold of 0.04 was used. The very limited sample of ships cannot be interpreted much. The majority of the small ships are fishing vessels and therefore it can be assumed that the majority of ships in the Kaggle Statoil/C-CORE data are fishing vessels.

The difference of the iceberg size density are interesting compared to the icebergs from Kaggle Statoil/C-CORE training data in figure 6.17a. There is a shift in where the estimated size start. Each bin in the figure is 5 pixel. The combination of HH and HV polarisation give a more robust estimate of what should be defined as targets. The smallest icebergs are thus filtered out and no icebergs smaller than an estimated size of 10 pixels are included. If a large enough custom dataset was created in this study the prediction

accuracy should theoretically increase.

7.5 Extended model

The Kaggle Statoil/C-CORE training data has been extended with pseudo labelled data from the Kaggle test set and custom data acquired for this study. The training dataset is extended from 1600 ships and icebergs to 6700 ships and icebergs. The 6700 images have been shuffled and a 80% 20% split has been made for training and validation.

It was tested in figure 6.18 if L2 weight decay regularisation should be applied, see section 2.3.1. An optimal L2 weight decay parameter was found to be 1×10^{-2} . This hyperparameter was found from training the network 50 times and see which model, with a given weight decay, gave the best validation loss. A good start for the learning rate for all models was found to be 1×10^{-4} . When reduced learning rate on plateau is included it does not make sense to do a hyperparameter search for the learning rate. For the learning rate it is important that the network starts its learning from the first epoch. With higher values for the learning rate than 1×10^{-4} the network was unable to learn in the beginning of the training.

The custom modified Unet has been trained 5 times with the found optimal L2 weight decay hyperparameter. The accuracy and loss curves for these models can be seen in figure 6.19. These curves cannot be compared to previous results. The pseudo labelled ships from the Kaggle Statoil/C-CORE test data is very similar to the original training data. Therefore these images will be classified with a very good accuracy. The custom prepared images of icebergs used in this study were located with an improved technique that eliminates detection of very small targets. With more dominant pixels of the target are they easier to classify correct. The validation accuracy has thus increased from 88%-90% to 98%. Two models has been evaluated after 60 epoch. The Kaggle test score with the new trained model including the extended dataset improved the score slightly but not statistically significant. This model has included the incidence angle and target size as a node in the fully connected part of the neural network.

The two models has been evaluated on the custom data prepared for this study. This includes 291 icebergs and 62 ships. To be able to use the trained network for applications the network have to be able to generalise to new data. Authorities are interested in real-time or close to near-real time monitoring of the ocean.

In figure 6.20 the prediction of the custom data can be seen. The plot is for binary classification so 1 represent iceberg and 0 is a ship. All icebergs 291 are classified correct. Unfortunately the majority of ships also gets classified as icebergs. Model 2 can only predict 18% of the ships correct and model 1 performs even worse. These trained network on the extended dataset are thus very skewed toward icebergs for the custom data.

5 models were trained with the extended dataset with the modified Unet. For every 5 epoch a model was saved. All these saved model was examined to see if an other model could predict ships with a more precise classification. A model was found after 35 epoch with a ship accuracy of 60%. The prediction for this model can be seen in figure 6.21. The Kaggle test score was calculated for this model. The model performed far worse on the Kaggle test set with a score of 0.407. The Kaggle test score is therefore not a very good estimation of how well the model is able to generalise on other data. It is far more important how well the model performs on "real" images for application purposes.

To see if there was a general problem or a problem with the extended dataset, the custom

ships and icebergs predicted with the modified Unet and Densenet169 were trained with the original training data. The prediction can be seen in figure 6.22. These two network can predict the ships with very good accuracy (59/62). Both models struggles more with classifying iceberg correct. The best of the two models for icebergs could classify 71% correct.

The two model trained on the extended dataset has seen similar images before. All the icebergs are taken from Disko bay in Greenland. Therefore the other icebergs from the same area should be easier to classify. All these icebergs are "new" from Ilulissat glacier. The icebergs in the Kaggle Statoil/C-CORE dataset have potentially taken a long journey in the ocean. On the journey of the icebergs the iceberg can melt and freeze again. This can change the backscatter signature of the iceberg. Further if the same processing steps have not been taken this could influence the classification.

The models where only the original training dataset has been used the models were not as biased.

7.5.1 Ensemble network

The 8 best models have been combined in an ensemble network. An ensemble network combines differently trained network. How these models are combined varies. The models could be weighted if a model is significant better an other. For the 8 models the estimated Kaggle were score very similar in the range between 0.212 to 0.220. Weights have therefore not been applied to the ensemble network.

Two ensemble network are proposed. The first model is a mean ensemble and the second is a median ensemble. For a given ship or iceberg a probability is calculated for each model. The 8 probabilities are then combined by either applying the mean or median of all these individual probabilities. The combination of the 8 models gives a more robust estimate than a single model. The two ensemble models gives a better Kaggle test score than any of the individual models.

7.6 Comparison to the Kaggle Competition

The winner and the best Kaggle test score was 0.08227. This is far superior to the best score from ensemble network in this study (0.188). The winner has heavily exploited the incidence angle and argued this was the key to the good score. The data can be split into 2 groups as illustrates in figure 7.5.

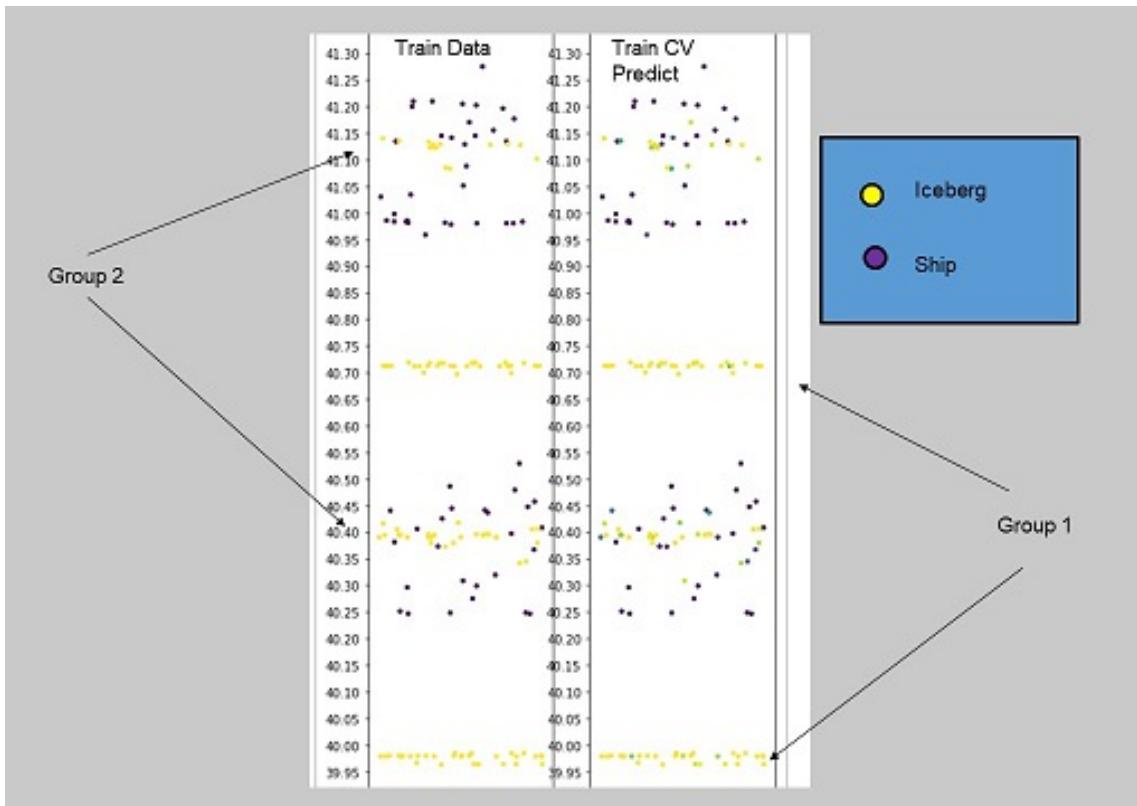


Figure 7.5: Ships and icebergs location at different incidence angles. Figure from the Kaggle winners, <https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/discussion/48241>

To win the competition this was smart. For real world applications this information cannot be used. Ships and icebergs can be located anywhere in a SAR image. There should be no correlation between the incidence angle and the target. This is clearly an error in the Kaggle Statoil/C-CORE dataset that should not occur. The focus of this master thesis was to use the Kaggle dataset as a stepping stone to classify custom data prepared for this study. This information has therefore not been exploited. The winner used an ensemble of more than 100 different Convolutional neural networks and tested many combinations to find the optimal model.

8 Future Work

As described, the Kaggle Statoil/C-CORE dataset have several drawbacks. To improve the classification further a custom dataset had to be used for training. Part of a custom dataset was generated for this study to propose a default method to work with Sentinel 1 scenes. In this study Sentinel 1 GRD IW scenes have been used as a starting point for the preprocessing steps. It was shown that noise remained in the image from the bursts and noise from TOPSAR scanmode was present in the IW images. More control over the preprocessing steps are required to improve the results further.

Sentinel 1 Single Look Complex (SLC) IW images are proposed as the starting point of the preprocessing steps. By starting from SLC IW images and improving already proposed preprocessing steps, 3 things can be improved which could potentially help with the classification between ships and icebergs.

- Remove noise from bursts and thermal noise in the subswaths
- Investigate multilooking
- Generate wind field estimates

It is desired to find a better noise removal for bursts and thermal noise than the default proposed processing step by ESA. With less noise from bursts and thermal noise in the subswaths is possible to correct better for the incidence angle in HV polarisation. It can be important for the sub images of ships and icebergs that they are in the same range of backscatter. It is critical to make these corrections if object detection on full images are preferred as for the You Only Look Once (YOLO) algorithm, [Bochkovskiy, Wang, and Liao 2020].

The Sentinel 1 SLC IW images were not multi-looked yet. The original spatial resolution in SLC SAR images are $2.7 \times 22\text{ m}$ to $3.5 \times 22\text{ m}$. 5 "looks" are by default applied when the GRD images are created so the resolution are $20 \times 22\text{ m}$. The backscatter types from ocean and ships are primarily categorised as specular or double bounce. It might not be necessary to multi-look the images as much to reduce speckle over the ocean. This is especially true for calm weather conditions. Information about the target could be improved with the better spatial resolution if the image is multi-looked less as a trade-off with increased speckle noise.

It was briefly mentioned in the SAR theory that wind influence the ocean backscatter. To compare multiple SAR scenes from different dates the influence of wind on the backscatter should be minimised. A wind map could be inserted as an additional band in the convolutional neural network. [Ahsbahs 2018] propose a method to estimate the local wind field from a SLC SAR image.

With good preprocessing steps and full control over how the images are processed the workflow can be further automated. By generating SAR images of ships from AIS data new possibilities arise. The custom dataset should be expanded further with ship information about shiptype, length and width. With enough ships the classification problem could be split up to a multiclass classification problem, with shiptype and iceberg in the Arctic. The size of the ship can be cross referenced with the estimated pixel area from

the SAR image. This can possibly detect if a ship sends out false information.

Many previous Convolutional Neural Network (CNN) have been designed to predict RGB images. 4 different types of CNNs architecture have been trained and their results have been investigated in this master thesis. No specific network was estimated to be superior on this type of data. Further investigation on network type, data augmentation and hyper-parameter could be beneficial.

More data sources other than Sentinel 1A and 1B IW images could be beneficial. Sentinel 1 IW images are only used over land and in coastal areas. If monitoring of ships further away from the coast are desired, other data sources should be utilised or ESA could be requested to capture more IW images. ESA has planned to launch Sentinel 1C and 1D, both are similar SAR satellites as 1A and 1B. Sentinel 1C is scheduled to be launched in 2022. Sentinel 1C is planned to have an included AIS receiver to log ship information. It would therefore be ideal if only IW images were captured. Data from more SAR satellites increases the temporal resolution and monitoring capabilities.

Sentinel 1 data could be merged with Sentinel 2 that are Multi-spectral with 13 bands in the visible, near infrared, and short wave infrared range of the spectrum. Sentinel 2 can only be used in the summer months since the surface have to be illuminated by the sun. Optical bands cannot see through skies so there have to be clear skies. Optical images can give better predictions for ship/iceberg classification but due to these limitations it is not feasible as a primary source of data. Sentinel 2 images could be used as a supplement for SAR satellites.

9 Conclusion

Classification of ships and icebergs in Synthetic Aperture Radar (SAR) images have been investigated. A Kaggle dataset from Statoil/C-CORE has been used as a starting point. A subcategory of machine learning called deep learning has been used to predict if an image was an iceberg or ship. 4 different Convolutional Neural Network (CNN) has been applied on the data. Of all trained models a custom modified Unet, Resnet18 and different Densenet showed the best results when the data was evaluated. None of 8 best models were significantly better than another.

An objective of this master thesis was to see if a trained CNN could be used to predict other images of ships and icebergs. The Kaggle Statoil/C-CORE dataset images were from Sentinel 1 SAR satellite. Many preprocessing steps have to be taken for the SAR scenes before an image of a iceberg or ship can be generated. The Kaggle Statoil/C-CORE dataset had no record of how the images were generated and what preprocessing steps should be taken. To generate a custom dataset for this study a range of preprocessing has been applied. From the processed SAR scenes over the Disko bay and Nuuk area a land masking has been applied from a bathymetry model. The influence of incidence angle as a function of backscatter intensity have been investigated. It is shown and confirmed that there is a strong dependence on backscatter as a function of incidence angle. The ocean acts as mirror if there is no wind. Very little backscatter is thus returned to the receiver in the SAR satellite. The small backscatter values gives some unwanted additive noise contribution in HV polarisation scenes. The additive noise is due to how the SAR images are generated. With limited time for this thesis a solution to the additive noise problem has not been made. The problem have been investigated and should be solved to improve the classification of ships and icebergs.

2891 icebergs and 62 ships were located in SAR scenes for this study. The icebergs were located in the Disko bay area. The ships were located in the Disko bay and Nuuk area by AIS positions. There is limited marine traffic in the arctics and therefore it is difficult to locate many ships. A Constant False Alarm Rate (CFAR) were used to find potential targets. The target were found from a combined HH and HV polarisation image. The combined image gave a more robust estimate of what should be classified as a target. Smaller insignificant icebergs were thus excluded and potential false positives in the later ship iceberg classification was avoided.

Custom icebergs found in SAR scenes and pseudo labelled ships from Kaggle Statoil/C-CORE test set were merged into the the original training data to extend the training dataset. The extended dataset gave better results for the Kaggle test set but was unable to predict custom ships and icebergs well. Both of the classes were predicted as icebergs for the extended dataset. The Custom data was also predicted with a modified Unet and Densenet169 trained on the original data. These two networks were better to distinguish between ships and icebergs. For the remaining custom dataset the best network could predict correctly 95% for ships and 71% for icebergs.

To improve the Kaggle test score an ensemble network was applied. The 8 best models and their prediction of the test data were merged. A mean and median of these scores were calculated and the combination of these network resulted in a better Kaggle test score than any of the individual scores.

Bibliography

- Nielsen, Michael (2019). *Neural Networks and Deep Learning*. URL: www.neuralnetworksanddeeplearning.com.
- He, K. et al. (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.
- Srivastava, Nitish et al. (Jan. 2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1, pp. 1929–1958. ISSN: 1532-4435.
- Ioffe, Sergey and Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 448–456.
- Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980.
- Simonyan, Karen and Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556.
- He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385.
- Huang, Gao et al. (2018). *Densely Connected Convolutional Networks*. arXiv: 1608.06993.
- Hu, Jie et al. (2019). *Squeeze-and-Excitation Networks*. arXiv: 1709.01507.
- Crisp, David (May 2004). *The state-of-the-art in ship detection in Synthetic Aperture Radar imagery*.
- Mladenova, I. E. et al. (2013). "Incidence Angle Normalization of Radar Backscatter Data". In: *IEEE Transactions on Geoscience and Remote Sensing* 51.3, pp. 1791–1804.
- Statoil/C-CORE Iceberg Classifier Challenge* (2017). URL: <https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/overview> (visited on 09/01/2020).
- Lensjø, Øyvind (May 2018). *SAR ship detection in dual polarization combination channels*.
- Weiß, Thomas (Feb. 2019). *MULTIPLY SAR pre-processing Documentation*.
- Filipponi, Federico (June 2019). "Sentinel-1 GRD Preprocessing Workflow". In: *Proceedings* 18, p. 6201. DOI: 10.3390/ECRS-3-06201.
- Sørensen, Kristian (2021). *AIS tracking of ships using Neural Networks and correlation with satellite images*.
- ESA (2021). *Interferometric Wide Swath*. URL: <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/interferometric-wide-swath> (visited on 01/15/2021).
- Lee, Peter Q., Linlin Xu, and David A. Clausi (2020). "Sentinel-1 additive noise removal from cross-polarization extra-wide TOPSAR with dynamic least-squares". In: *Remote Sensing of Environment* 248, p. 111982. ISSN: 0034-4257. DOI: <https://doi.org/10.1016/j.rse.2020.111982>. URL: <http://www.sciencedirect.com/science/article/pii/S0034425720303527>.
- Greidanus, H. (2008). "Satellite Imaging for Maritime Surveillance of the European Seas". In: *Remote Sensing of the European Seas*. Ed. by Vittorio Barale and Martin Gade. Dordrecht: Springer Netherlands, pp. 343–358. ISBN: 978-1-4020-6772-3. DOI: 10.1007/978-1-4020-6772-3_26. URL: https://doi.org/10.1007/978-1-4020-6772-3_26.
- Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv: 2004.10934.
- Ahsbahs, Tobias Torben (2018). "Remote Sensing of Offshore Winds for Wind Energy Applications". In: DOI: <https://doi.org/10.11581/dtu:00000070>.

A Modified Unet

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 75, 75]	896
BatchNorm2d-2	[-1, 32, 75, 75]	64
ReLU-3	[-1, 32, 75, 75]	0
Conv2d-4	[-1, 32, 75, 75]	9,248
BatchNorm2d-5	[-1, 32, 75, 75]	64
ReLU-6	[-1, 32, 75, 75]	0
DoubleConv-7	[-1, 32, 75, 75]	0
AdaptiveAvgPool2d-8	[-1, 32, 1, 1]	0
Linear-9	[-1, 2]	64
ReLU-10	[-1, 2]	0
Linear-11	[-1, 32]	64
Sigmoid-12	[-1, 32]	0
SELayer-13	[-1, 32, 75, 75]	0
MaxPool2d-14	[-1, 32, 37, 37]	0
Conv2d-15	[-1, 64, 37, 37]	18,496
BatchNorm2d-16	[-1, 64, 37, 37]	128
ReLU-17	[-1, 64, 37, 37]	0
Conv2d-18	[-1, 64, 37, 37]	36,928
BatchNorm2d-19	[-1, 64, 37, 37]	128
ReLU-20	[-1, 64, 37, 37]	0
DoubleConv-21	[-1, 64, 37, 37]	0
Down-22	[-1, 64, 37, 37]	0
AdaptiveAvgPool2d-23	[-1, 64, 1, 1]	0
Linear-24	[-1, 4]	256
ReLU-25	[-1, 4]	0
Linear-26	[-1, 64]	256
Sigmoid-27	[-1, 64]	0
SELayer-28	[-1, 64, 37, 37]	0
MaxPool2d-29	[-1, 64, 18, 18]	0
Conv2d-30	[-1, 128, 18, 18]	73,856
BatchNorm2d-31	[-1, 128, 18, 18]	256
ReLU-32	[-1, 128, 18, 18]	0
Conv2d-33	[-1, 128, 18, 18]	147,584
BatchNorm2d-34	[-1, 128, 18, 18]	256
ReLU-35	[-1, 128, 18, 18]	0
DoubleConv-36	[-1, 128, 18, 18]	0
Down-37	[-1, 128, 18, 18]	0
AdaptiveAvgPool2d-38	[-1, 128, 1, 1]	0
Linear-39	[-1, 8]	1,024
ReLU-40	[-1, 8]	0
Linear-41	[-1, 128]	1,024
Sigmoid-42	[-1, 128]	0
SELayer-43	[-1, 128, 18, 18]	0

MaxPool2d-44	[-1, 128, 9, 9]	0
Conv2d-45	[-1, 256, 9, 9]	295,168
BatchNorm2d-46	[-1, 256, 9, 9]	512
ReLU-47	[-1, 256, 9, 9]	0
Conv2d-48	[-1, 256, 9, 9]	590,080
BatchNorm2d-49	[-1, 256, 9, 9]	512
ReLU-50	[-1, 256, 9, 9]	0
DoubleConv-51	[-1, 256, 9, 9]	0
Down-52	[-1, 256, 9, 9]	0
AdaptiveAvgPool2d-53	[-1, 256, 1, 1]	0
Linear-54	[-1, 16]	4,096
ReLU-55	[-1, 16]	0
Linear-56	[-1, 256]	4,096
Sigmoid-57	[-1, 256]	0
SELayer-58	[-1, 256, 9, 9]	0
MaxPool2d-59	[-1, 256, 4, 4]	0
Conv2d-60	[-1, 256, 4, 4]	590,080
BatchNorm2d-61	[-1, 256, 4, 4]	512
ReLU-62	[-1, 256, 4, 4]	0
Conv2d-63	[-1, 256, 4, 4]	590,080
BatchNorm2d-64	[-1, 256, 4, 4]	512
ReLU-65	[-1, 256, 4, 4]	0
DoubleConv-66	[-1, 256, 4, 4]	0
Down-67	[-1, 256, 4, 4]	0
AdaptiveAvgPool2d-68	[-1, 256, 1, 1]	0
Linear-69	[-1, 16]	4,096
ReLU-70	[-1, 16]	0
Linear-71	[-1, 256]	4,096
Sigmoid-72	[-1, 256]	0
SELayer-73	[-1, 256, 4, 4]	0
Upsample-74	[-1, 256, 8, 8]	0
Conv2d-75	[-1, 128, 9, 9]	589,952
BatchNorm2d-76	[-1, 128, 9, 9]	256
ReLU-77	[-1, 128, 9, 9]	0
Conv2d-78	[-1, 128, 9, 9]	147,584
BatchNorm2d-79	[-1, 128, 9, 9]	256
ReLU-80	[-1, 128, 9, 9]	0
DoubleConv-81	[-1, 128, 9, 9]	0
Up-82	[-1, 128, 9, 9]	0
AdaptiveAvgPool2d-83	[-1, 128, 1, 1]	0
Linear-84	[-1, 8]	1,024
ReLU-85	[-1, 8]	0
Linear-86	[-1, 128]	1,024
Sigmoid-87	[-1, 128]	0
SELayer-88	[-1, 128, 9, 9]	0
Upsample-89	[-1, 128, 18, 18]	0
Conv2d-90	[-1, 64, 18, 18]	147,520
BatchNorm2d-91	[-1, 64, 18, 18]	128
ReLU-92	[-1, 64, 18, 18]	0
Conv2d-93	[-1, 64, 18, 18]	36,928

BatchNorm2d-94	[-1, 64, 18, 18]	128
ReLU-95	[-1, 64, 18, 18]	0
DoubleConv-96	[-1, 64, 18, 18]	0
Up-97	[-1, 64, 18, 18]	0
AdaptiveAvgPool2d-98	[-1, 64, 1, 1]	0
Linear-99	[-1, 4]	256
ReLU-100	[-1, 4]	0
Linear-101	[-1, 64]	256
Sigmoid-102	[-1, 64]	0
SELayer-103	[-1, 64, 18, 18]	0
Upsample-104	[-1, 64, 36, 36]	0
Conv2d-105	[-1, 32, 37, 37]	36,896
BatchNorm2d-106	[-1, 32, 37, 37]	64
ReLU-107	[-1, 32, 37, 37]	0
Conv2d-108	[-1, 32, 37, 37]	9,248
BatchNorm2d-109	[-1, 32, 37, 37]	64
ReLU-110	[-1, 32, 37, 37]	0
DoubleConv-111	[-1, 32, 37, 37]	0
Up-112	[-1, 32, 37, 37]	0
AdaptiveAvgPool2d-113	[-1, 32, 1, 1]	0
Linear-114	[-1, 2]	64
ReLU-115	[-1, 2]	0
Linear-116	[-1, 32]	64
Sigmoid-117	[-1, 32]	0
SELayer-118	[-1, 32, 37, 37]	0
AvgPool2d-119	[-1, 32, 18, 18]	0
Linear-120	[-1, 1024]	10,617,856
Dropout-121	[-1, 1024]	0
Linear-122	[-1, 1]	1,025
Sigmoid-123	[-1, 1]	0

=====

Total params: 13,965,025

Trainable params: 13,965,025

Non-trainable params: 0

Input size (MB): 0.06

Forward/backward pass size (MB): 28.49

Params size (MB): 53.27

Estimated Total Size (MB): 81.83

Technical
University of
Denmark

Akademivej, Building 356
2800 Kgs. Lyngby
Tlf. 4525 1700