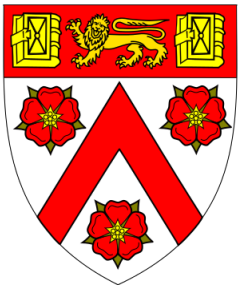
Ship Detection using Convolutional Neural Networks and Augmented SAR Data



Syed Akhass Adnan Wasti

Supervisor: Dr. Joan Lasenby

Department of Engineering
University of Cambridge

A Fourth-year Project Report
Submitted in Partial Fulfilment of the Requirements
For the Degree of
Master of Engineering
In
Information and Computer Engineering

Trinity College 30th May 2018

Acknowledgements

This project was completed under the supervision of Dr Joan Lasenby, whom I would like to thank for her invaluable advice, guidance and encouragement. I am also grateful to Flavio Bergamashce for all the helpful discussions and suggestions. Their input was crucial to this project.

The data for this project was obtained from the European Space Agency's Copernicus Open Access Hub which has proved to be an excellent resource. I am also grateful to the Copernicus Research and User Support Service for allowing me to use their virtual machines for data handling.

My Directors of Studies in Trinity College have always been excellent mentors and generous with their time and advice. They are Professor Nick Kingsbury (2014-2015, 2016-2017), Dr Per Ola Kristensson (2015-2016), and Dr Joan Lasenby (2017-2018). I am eternally in their debt.

I have been supported by scholarships provided by the Cambridge Trust, Oxford and Cambridge Society Karachi Educational Trust and Trinity College during my four years at Cambridge and I am deeply grateful to them for their generosity.

Abstract

The aim of this project was to develop a model that detects marine vessels from Synthetic Aperture Radar (SAR) data with an accuracy comparable to the state-of-the-art segmentation models. The implications of such a technology can be far-reaching for marine surveillance and would help tackle illicit activities such as piracy and smuggling, which are otherwise difficult to detect and intercept as marine surveillance largely relies on short range radar mounted on airplanes and marine vessels, and requires active monitoring of the waterways.

While previous work on SAR data has focussed on land-based targets and creating annotated SAR datasets for them, no such dataset exists in the public domain for the water-based targets, more specifically marine vessels. Hence, we first investigated ways of creating a labelled SAR dataset. The Automatic Identification System (AIS) data was investigated as a potential ground truth to be used for annotating the SAR data. However, we found that AIS and SAR data do not coincide exactly due to unreliability of the AIS timestamps.

Adopting a manual approach to data labelling, we created an initial labelled dataset which was then augmented by applying combinations of random affine transformation. A benchmark algorithm in the form of Constant False Alarm Rate (CFAR) was implemented to base the performance of more sophisticated classification models that we developed subsequently.

A U-Net inspired symmetric Convolutional Neural Network (CNN) was developed and the orignial image segmentation problem was re-formulated into pixel-wise classification problem. Binary and multi-class classifiers were trained on the labelled SAR data and their performance evaluated over several metrics.

We found that the one-vs-all approach to classifying SAR image pixels as corresponding to either land, sea or ship was best suited to the ship detection task, while a multi-class classifier would be more suited for optimum image segmentation if accuracy of ship detections is not a priority. All classifiers were based on the symmetric CNN model that we developed.

Table of contents

Chapter 1

Introduction

Maritime surveillance has been receiving a growing interest during recent years [1, 2]. It can be performed with information from vessel monitoring systems based on cooperative transmitting technologies or from detection sensors. Cooperative sources are equipped with electronic navigation sensors in order to estimate the position of a ship (usually a global navigation satellite system (GNSS) or an inertial sensor), and with radio equipment for communication purposes. Conversely, one usually relies on remote sensing equipment such as radars or image sensors for non-cooperative scenarios. Such equipment can be deployed in coastal stations, surveillance ships, aircraft and satellites.

Despite providing useful information, cooperative systems alone are not fully appropriate for maritime surveillance due to the diversity of surveillance scenarios and the presence of non-cooperative targets. As an example, in scenarios with illicit activities such as piracy, illegal fishing or smuggling, cooperative data can be counterfeit, masked or even not transmitted at all (small boats). On the other hand, non-cooperative systems are less sensitive to deception in non-military situations and thus have proven to be useful for some specific maritime surveillance scenarios [3, 4]. However, there are still plenty of remaining challenges related to maritime surveillance using only non-cooperative systems since information recovery is limited and the detections need to be interpreted somehow. These challenges include ship detection, identification and tracking, or speed and heading estimation.

In this project, we will focus on ship detection using one such non-cooperative system that is Synthetic Aperture Radar (SAR). SAR has the advantage over optical remote sensing methods in that it is based on an active illumination system, meaning that it doesn't rely on any external illumination source, such as the Sun. Radar has strong penetrating ability in the atmosphere and is unaffected by cloud-cover. Therefore, SAR can practically operate in all-weather day-and-night conditions.

Traditional machine learning methods for the SAR automatic target recognition include Support Vector Machines (SVM) [5], local texture features [6, 7], dictionary learning [8, 9],

and sparse representation [10]. These methods have produced some promising results, but they heavily rely on the hand-crafted feature extraction [11]. Because of the imaging nature, clutter and speckle noise exist in the SAR images, which increases the difficulty of feature extraction despite the fact that experts are involved.

In recent years, with the development of deep learning techniques, Convolutional Neural Networks (CNNs) have received great attention in object recognition [12–14]. They can automatically extract the target features without experts' intervention. Compared with the traditional machine learning methods, the CNN is more effective and robust and has been successfully applied to SAR image recognition.

Deep learning techniques on SAR image recognition have only been successful in applications where extensive labelled SAR datasets were available for training the Neural Networks. Take for example the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset [15] that has been used widely by researchers to train Neural Networks for land-based target recognition. However, such a labelled SAR dataset does not exist (at least in the public domain) for sea-based targets and this proved to be a major stumbling block for our project, prompting us to investigate viable methods of creating a labelled dataset before progressing on to training a CNN for image recognition. As we soon found, the former task was a major undertaking in itself.

This project aimed to explore the performance of CNNs on SAR data for ship detection. Our approach consisted of three stages: 1) implementing the Constant False Alarm Rate (CFAR) algorithm as benchmark to assess the performance of more sophisticated frameworks; 2) creating an initial manually-labelled SAR dataset and applying affine transformations to augment this dataset; and 3) training a U-Net [16] inspired CNN on this augmented dataset. It is hoped that the trained CNN can then be used to further extend the unlabelled SAR dataset using Active Learning techniques [17].

The rest of this report is organised as follows: Section 2 discusses previous work that has been done in SAR image recognition, Section 3 contains the theory about the SAR imagery, AIS data, and CFAR algorithm. Section 4 describes the data pre-processing pipeline, the CNN architecture and network training. Section 5 presents and discusses the results, and the conclusions are finally drawn in Section 6.

Chapter 2

Background

The use of Neural Networks on SAR images has received growing interest during recent years [18, 19]. A fully convolutional deep neutral network constructed for SAR automatic target recognition (ATR) was tested on the Moving and Stationary Target Acquisition and Recognition (MSTAR) [15] benchmark dataset where it achieved 99% accuracy on classification of ten-class targets [20]. Such successful application of Neural Networks in SAR-ATR on land-based targets encouraged us to explore the potential of Neural Networks in classification of SAR images of sea-based targets as well. We focussed on ship detection due to the diversity in their shapes and sizes to test the robustness of our framework.

Alternate methods to Neural Networks, such as hand-crafted image segmentation methods that rely on edge information, do not work well with SAR data due to the presence of speckle noise. Speckle noise in SAR images arises from coherent summation of signals scattered from the ground scatterers distributed randomly within each pixel. This noise renders the edge information in the SAR image unreliable.

Using intensity information on its own for SAR image classification is also unreliable because intensity values for different surfaces are not necessarily unique and depend on a number of factors: types, sizes, shapes and orientations of the scatterers in the target area; moisture content of the target surface; frequency and polarisation of the radar pulses; as well as the incident angles of the radar beam.

The European Space Agency's (ESA) Copernicus mission [21] acquires terabytes of SAR data and makes it freely available to researchers. However, this data is unlabelled. Manually labelling SAR data is very labour intensive and suffers from operator variability [22]. The lack of a labelled dataset severely undermines our ability to assess the performance of any automated segmentation technique on SAR images of vessels, making it impossible to train Neural Networks.

A framework for labelling SAR images based on pixel-wise annotation was developed in [18]. This allowed CNNs to be trained for image segmentation. However, this approach requires 3D-Computer Aided Design (CAD) models of every target object and scene for accurate annotation of objects in SAR images. This is simply not feasible given the large variety of vessels.

An automatic segmentation technique using histogram thresholding together with morphological filtering was developed in [23]. However, this technique fails to connect the target and contiguous shadow regions due to a boundary where the intensity is similar to background intensity.

The algorithm proposed in [17] uses Active Learning to find the most reliable unlabelled samples to extend the training set, and a semi-supervised method to maximally utilise the class probability contained in the remaining unlabelled samples. This is a promising approach, free from the pitfalls of aforementioned techniques, that we can use to extend an initial labelled SAR dataset.

Chapter 3

Theory Relevant to the Project

3.1 Sythetic Aperture Radar

Environmental monitoring, earth-resource mapping, and military systems require broad-area imaging at high resolutions. Often, this imagery must be acquired at night or during inclement weather. SAR provides such a capability due to its active illumination system. SAR systems take advantage of the long-range propagation characteristics of radar signals and the complex information processing capability of on-board digital electronics to provide high resolution imagery. A detailed description of the theory of SAR is complex and beyond the scope of this report. Instead, this section is intended to give the reader a working understanding of how SAR functions

3.1.1 Radar Geometry and Acquisition

Consider a spaceborne Synthetic Aperture Radar (SAR) imaging perpendicular to the satellite velocity as shown in the Figure 3.1. Typically, SAR produces a two-dimensional (2-D) image. One dimension in the image is called range (or cross track) and is a measure of the "line-of-sight" distance from the radar to the target. Range measurement and resolution are achieved in SAR in the same manner as most other radars: range is determined by measuring the time from transmission of a pulse to receiving the echo from a target and, in the simplest SAR, range resolution is determined by the transmitted pulse width, i.e. narrow pulses yield fine range resolution.

The other dimension is called azimuth (or along track) and is perpendicular to range. SAR's ability to produce relatively fine azimuth resolution sets it apart from other radars. The azimuth resolution is related to radar wavelength and length of antenna as shown in Equation 3.1. To obtain fine azimuth resolution, a physically large antenna is needed to focus the transmitted and received energy into a sharp beam. The sharpness of the beam defines the azimuth resolution.

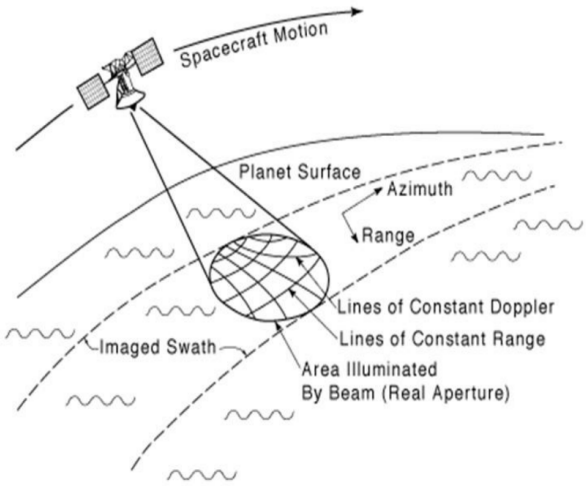


Fig. 3.1 An antenna, mounted on the spaceborne platform, transmits a radar signal in a side-looking direction towards the Earth's surface. The reflected signal, known as the echo, is backscattered from the surface and received a fraction of a second later at the same antenna. The distance travelled by the antenna during this period forms the 'Synthetic Aperture' which can be significantly larger than the actual length of antenna. Source: [24]

Similarly, optical systems, such as telescopes, require large apertures (mirrors or lenses which are analogous to the radar antenna) to obtain fine imaging resolution. Since SAR is much lower in frequency than optical systems, even moderate SAR resolutions require an antenna too large to be practically carried by a spaceborne platform: antenna lengths several hundred meters long are often required. However, spaceborne radar can collect data while flying this distance, and then process the data as if it came from a physically long antenna. The distance the aircraft flies in synthesizing the antenna is known as the synthetic aperture. A narrow synthetic beam-width results from the relatively long synthetic aperture, which yields finer resolution than is possible from a smaller physical antenna.

$$d_{azimuth} \approx \lambda R/L \tag{3.1}$$

where:

$d_{azimuth}$ = resolvable distance in the azimuth direction
λ = wavelength of radar
R = range
L = length of antenna

While this section attempts to provide a working understanding, SAR is not as simple as described above. Transmitting short pulses to provide range resolution is generally not practical. Typically, longer pulses with wide-bandwidth modulation are transmitted, which complicate the range processing but decreases the peak power requirements on the transmitter. For even moderate azimuth resolutions, a target's range to each location on the synthetic aperture changes along the

synthetic aperture. The energy reflected from the target must be "mathematically focused" to compensate for the range dependence across the aperture prior to image formation. Additionally, for fine-resolution systems, the range and azimuth processing are coupled (dependent on each other) which also greatly increases the computational processing.

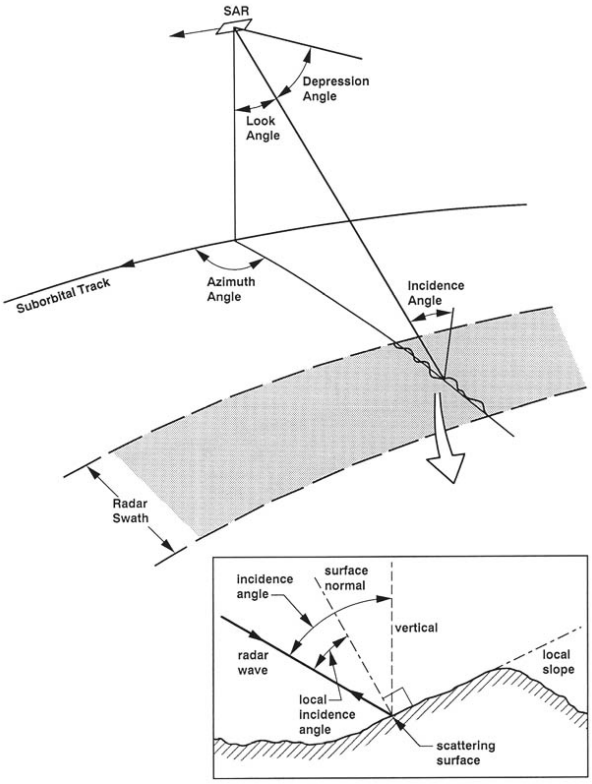


Fig. 3.2 SAR Geometry [25]

3.1.2 SAR Dataset from the European Space Agency's Sentinel-1 Mission

The SAR dataset used in this project was sourced from the European Space Agency's Sentinel-1 Mission which provides accurate and real-time SAR data. The mission consists of two polar-orbiting satellites: Sentinel 1A and Sentinel 1B. The radar on-board these satellites operate in dual polarisation the C-band (5.4Ghz) spectrum, enabling SAR imagery to be acquired regardless of the weather. The radar operate continuously, round the clock, with the satellites' revisit times being as low as 4 days in some regions.

SAR Imaging Mode

Of the four imaging operating modes that Sentinel-1 operates in, shown in Figure 3.3, we are particularly interested in the Interferometric Wide (IW) swath imaging mode as it offers the highest spatial resolution (5m by 20m), which enables imaging of small boats.

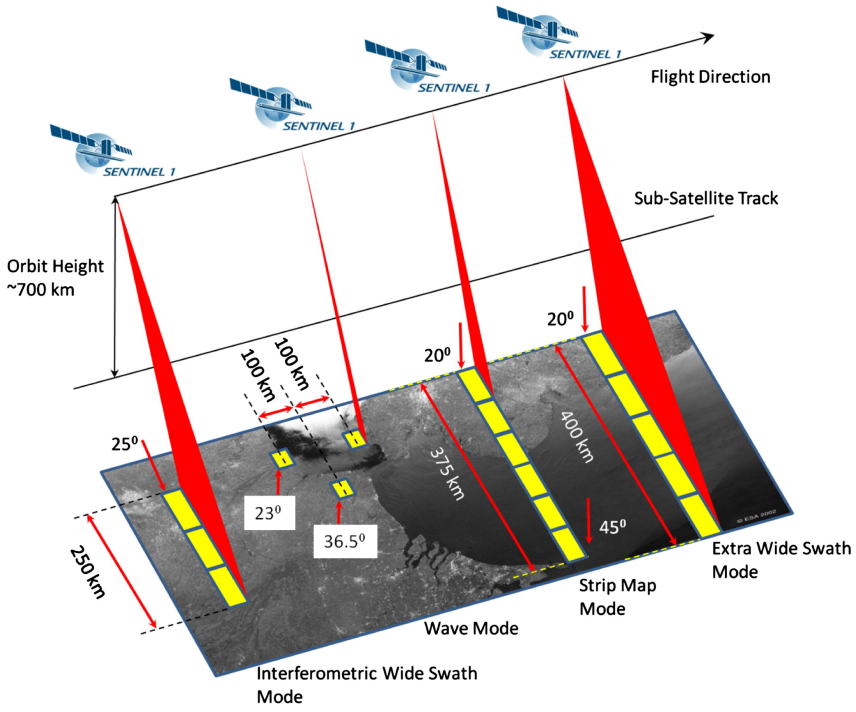


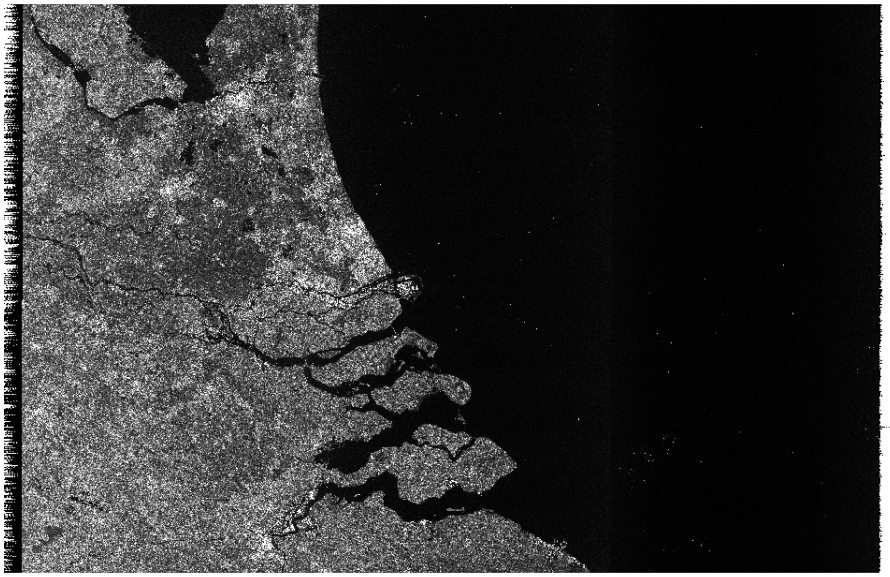
Fig. 3.3 Four SAR imaging modes are shown here with their respective angles of incidence and swath width. Source: [26]

The IW swath mode captures a 250 km swath, constructed from three sub-swaths using Terrain Observation with Progressive Scans SAR (TOPSAR) [27]. In addition to steering the beam in range, the beam is also electronically steered from backward to forward in the azimuth direction for each burst, avoiding scalloping artefacts when the sub-swaths are stitched together and resulting in homogeneous image quality throughout the swath.

Region of Interest

The IW swath mode is mainly used to image land masses, while the lower resolution Extra Wide (EW) swath mode is used to image the open sea. We decided to limit our efforts to coastal areas in the hopes of finding IW swath mode imaged areas with ships.The English Channel proved to be a suitable region with several busy ports on either side attracting heavy marine traffic and providing a diverse range of target objects to train our image segmentation model. Furthermore, since the English Channel is quite narrow, the radar remains in IW imaging mode as it traverses between mainland Europe and the UK, making it an ideal region to source high resolution SAR data for our project.

(a) Map view of English Channel



(b) A 26479x16678 pixels SAR image. Notice the radar-reflecting marine vessels appear as white specks, while the sea which reflect radar away from the satellite appears black.

Fig. 3.4 The English Channel was chosen as the region of interest. The region off the coast of Rotterdam, indicated by the red box in (a), is imaged by SAR and shown in (b). The image in SAR reference frame appears flipped in the range direction.

3.2 Automatic Identification System

The SAR data provided by the Sentinel-1 mission is un-labelled and can not be used directly to train and image segmentation model. We therefore needed some ground truth to label the SAR data and assess the accuracy of our model. The Automatic Identification System (AIS) was one such independent data source that seemed a good choice for ground truth.

Most cargo ships and all passenger ships regardless of their size are nowadays equipped with AIS transmitters and are required by the International Marine Organisation to broadcast ship information to nearby vessels at regular intervals [28]. The nominal reporting interval depends on the ship's dynamic conditions, which can vary from 3 minutes (for anchored ships not moving faster than 3 knots) to 2 seconds (e.g. for ships whose speed is greater than 23 knots). The AIS data is transmitted in the Very High Frequency (VHF) spectrum and has a range of 20-40 km, large enough to cover the English Channel. The information transmitted includes: vessel identity, position, size, speed, heading and timestamp. This information can be correlated with the SAR images to label individual marine vessels.

3.3 Constant False Alarm Rate Algorithm

The Constant False Alarm Rate Algorithm (CFAR) algorithm is able to detect a target object against a noisy background by comparing pixel intensities to a set threshold. This approach is far from optimal as it has an inherent trade off: setting too low a threshold results in higher target detections but at the expense of increased false detections. On the other hand, setting a high threshold results in fewer false detections but also lowers the number of target detections.

Since background noise is rarely constant in space and time, we implemented an adaptive algorithm that calculates the threshold, t, from pixel values such that the resulting Probability of False Alarm (PFA) is always constant, hence the name. The PFA is the probability of false target detection we expect to have.

The CFAR algorithm implemented here consists of two stages: 1) Adaptive Thresholding; and 2) Object Discrimination.

Adaptive thresholding

A window, illustrated in Figure 3.5, is moved across the input image pixel-by-pixel. At each position, the pixels that fall under the training cells are used to calculate the mean, μ_b, and standard deviation, σ_b, of the background noise, which is assumed to be normally distributed, Figure 3.6. Although the background noise is better modelled by a K-distribution, we approximate it with a Gaussian distribution here as this is sufficient for the purposes of our benchmark algorithm.

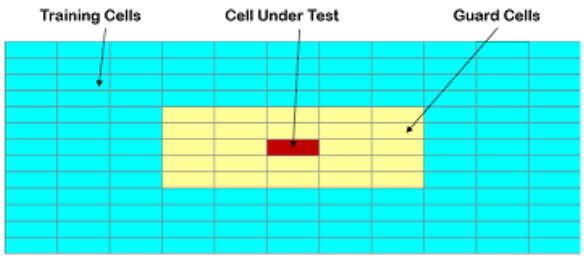


Fig. 3.5 Window used in Adaptive Thresholding. The pixels under Guard Cells are ignored to avoid overlap in the other two region [29]

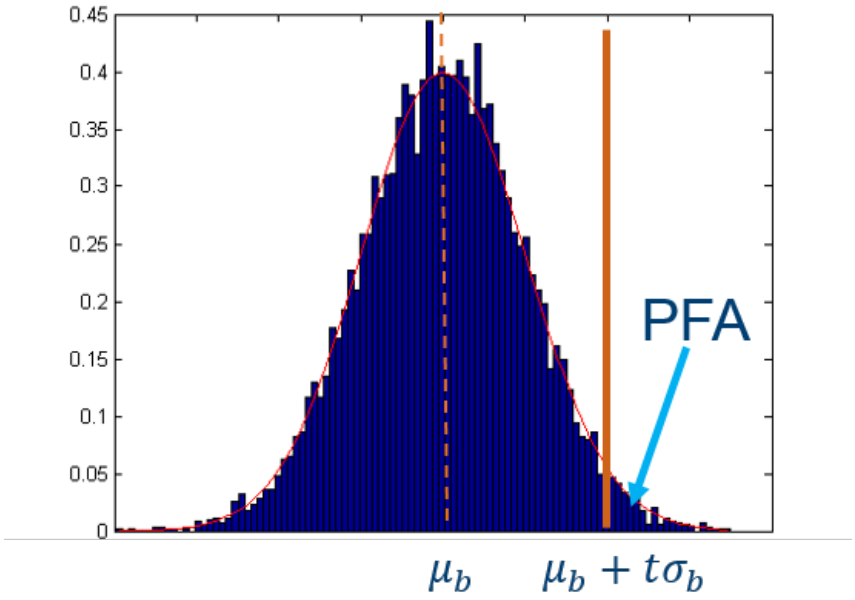


Fig. 3.6 Normally distributed background noise. Threshold, t, is calculated using Equation 3.2 with PFA expected to be $10^{-12.5}$ [30]

$$\text{PFA} = \frac{1}{2}\left(1 - \text{erf}\left(\frac{t}{\sqrt{2}}\right)\right) \tag{3.2}$$

The mean, μ_t, of the pixels under cell under test (CUT) is also calculated and the following criterion is applied to determine if the pixels under CUT are part of the target object.

$$\mu_t > \mu_b + t\sigma_b \leftrightarrow \text{target} \tag{3.3}$$

Object Discrimination

Once the window has traversed across the whole image, contiguous pixels that have been identified as belonging to a target object are grouped into clusters. If a cluster size falls within a pre-specified range, it is accepted as a target object, otherwise it is rejected. The minimum and

maximum acceptable sizes used were 30m and 600m, respectively, as we expect to find most vessels within this range.

Chapter 4

Data Processing and Network Model

4.1 SAR Data Pre-Processing

SAR data is freely available to download from the Copernicus Open Access Hub [31] in the form of a variety of 'products' with different levels of preprocessing already done. We use the Level-1 Ground Range Detected (GRD) product imaged in IW dual-polarisation (VH+VV) mode. Level-1 GRD products consist of focused SAR data that has been detected, multi-looked and projected to ground range using an Earth ellipsoid model. The dual-polarisation (VH+VV) mode contains both horizontal and vertical polarisation intensity measurements for radar emitted in vertical polarisation.

A typical SAR data product file is around 2GB, and contains amplitude and intensity bands, in addition to metadata such as pre-processing graphs, latitude and longitude grids, and vector data of associated ground-control points. Handling such files requires a lot of processing power and we were fortunate enough to have access to a powerful virtual machine at the Copernicus Research and User Support (RUS) Service [32] for the duration of this project. The virtual machine came pre-installed with the Sentinel Application Platform (SNAP) [33], a SAR data processing toolkit which we used to apply the relevant orbit file, perform re-calibration of the SAR data and extract the intensity bands from the products to use in our custom-built python data processing pipeline. Re-calibration with the orbit file, which contains the exact coordinates of the satellite at the time of data acquisition, ensures that the projection from SAR range to ground range is accurate.

4.1.1 Data Labelling

Complications with Using AIS

Initially we had hoped to implement a framework, similar to that in [2], for marine vessel location prediction based on historical AIS data. However, historical AIS data is not free and quite costly. We received a quote of upwards of £500 for AIS data recorded in a 24-hr period in our area of investigation shown in Figure 3.4b. We were, however, able to find AIS data for the region around Denmark from the Danish Maritime Authority's website [34]. This AIS data was in the form of a csv file and included thousands of entries for AIS signals recorded in a 24-hr period on May 1st, 2018. We downloaded the SAR data for that region for the same date and filtered out the AIS data entries based on the acquisition timestamp from the SAR data. The data entries with timestamps matching that of the SAR data were rendered as points at their respective geo-coordinates (latitude, longitude) using the QGIS toolkit and overlayed onto the Intensity_VH raster image. QGIS is an open source geographic information system that allows us to manipulate geospatial data.The combined data is illustrated in Figure 4.1.

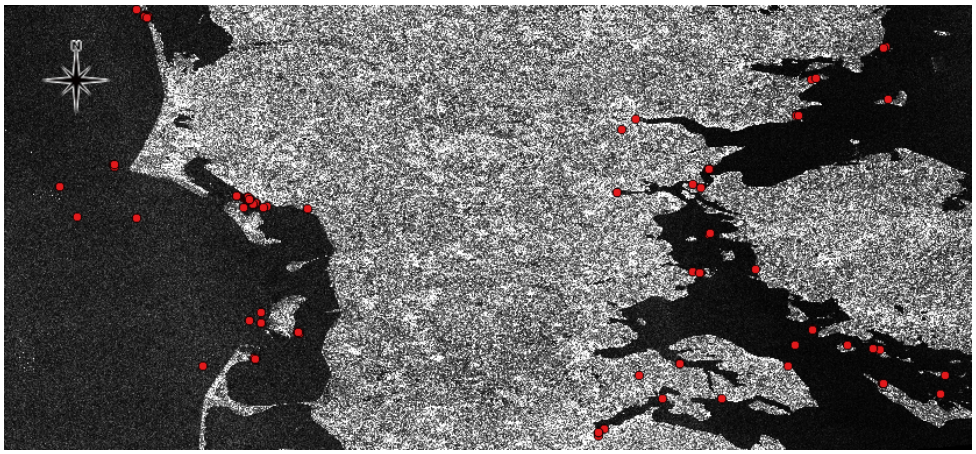


Fig. 4.1 AIS data points shown in red, overlayed onto the SAR Intensity_VH raster image. The AIS data points were fused with SAR data by matching their timestamps and geo-coordinates.

Closer inspection of the fused data revealed that several of the marine vessel locations indicated by AIS data were not representative of all marine vessels, but more problematic were the false positives, illustrated in Figure 4.2, which indicate that the AIS data did not exactly align with the SAR data, despite timestamp matching.

This misalignment between marine vessel locations represented in SAR data and AIS data meant that we could not use AIS data to carry out pixel-wise annotation, which we had hoped to use to create binary label masks for the SAR images.

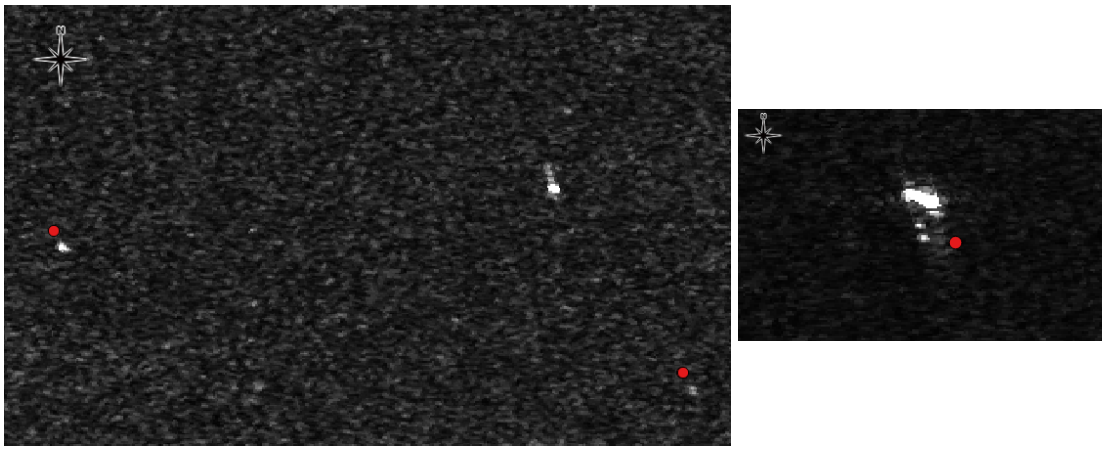


Fig. 4.2 Closer inspection of the fused AIS and SAR data shows that the marine vessel locations indicated by AIS data do not coincide with those represented in SAR data. Left: SAR data indicates presence of three marine vessels, however AIS provides only two data points and even they do not coincide. Right: By zooming in further on another marine vessel we observe that the AIS data point might indicate a location on the marine vessel's trajectory.

Creating a Shapefile

Faced with the complication of misalignment between the AIS data and SAR data, we decided to manually label an initial SAR dataset and augment it for training our CNN model. Initially, a SAR product file containing SAR imagery of just sea and ships was used, raster image shown in Figure 4.3(left). However, we later realised that to improve the performance of our model, we needed to train it on SAR data that contained not only ships in sea but also instances of land so that the model would be able to detect ships docked at port. Hence, we expanded out training and test dataset using a second SAR product file that contained SAR imagery of the Rotterdam port, raster image shown in Figure 4.3(right). The process of labelling the SAR data was the same and outlined below.

Each SAR product file was downloaded and its Intensity_VH and Intensity_VV bands were extracted using SNAP, and exported as a two-channel .tiff image. The marine vessels in this image were annotated as polygons in the QGIS toolkit. We created a shapefile consisting of polygons defined by the geo-coordinates of their vertices, where each polygon covered an area representing a ship.

The shapefile was exported as a CSV file and processed in Jupyter Notebook to create a binary mask array of size $5120 \times 5120 \times 1$. Pixels found within a polygon (representing a ship) were set to 1, while those outside (representing sea) were set to 0, using the fillpoly function from the OpenCV library. Next, the .tiff image containing the intensity bands was converted to an array and concatenated with the binary mask array to give a three-channel merged array. This merged array was reshaped into a $1600 \times 128 \times 128 \times 3$ array, ready for the augmentation stage.

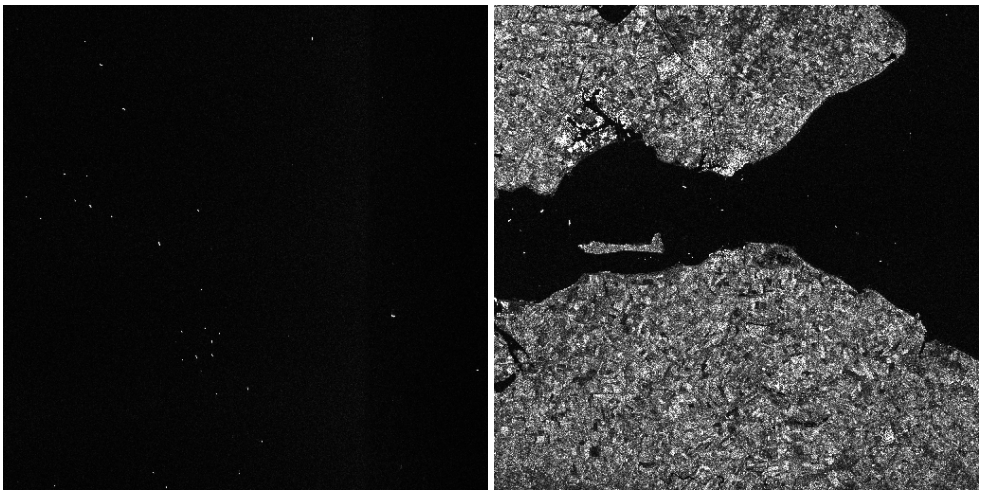


Fig. 4.3 Raster images of the two SAR product files that were labelled and processed into training and test datasets. Left: SAR data containing only sea and ships. Right: SAR data containing land, sea and ships.

While this approach of manually labelling the data is of course prone to operator error, it allowed us to progress towards the main part of this project: evaluating the performance of Neural Networks on ship detection from SAR images.

4.1.2 Data Augmentation

The merged array was augmented by applying affine transformations such as rotation, translation, flip, shear, and scaling. By defining the maximum extent of these transformations using the ImageDataGenerator from the Keras preprocessing library, we extended the original labelled dataset by a factor of 5 by applying random combinations of the aforementioned transformations. This data augmentation allowed us to use the available annotated samples more efficiently.

4.2 Convolutional Neural Network Model

Our CNN model was inspired by the family of U-Net architectures, which has been shown to outperform the sliding-window convolutional network [16]. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization.

While the U-Net was originally developed for segmentation of high resolution microscopy images, its application to SAR image segmentation was motivated by similarities in the two problem tasks: namely detecting regular shaped objects (cell organelles in microscopy images and ships in SAR images) from noisy background, although the noise in SAR data is far more complex than in microscopy images.

Apart from the symmetric structure, our CNN model is different from the original U-Net model in a number of respects. Two modifications in particular improved the performance of the network: keeping constant the number of feature maps (channels) between neurons and introducing batch-normalisation at the input of each neuron. These changes are discussed in the following sections.

Note on terminology: in the context of neural networks a neuron is a unit that receives several inputs, takes a weighted sum over them, and passes it through an activation function. In our CNN, we define a neuron to consist of batch-normalisation, convolution operation and activation function (e.g. BN_CONV_RELU in Figure 4.4 is one neuron).

4.2.1 Binary Classification vs. Multi-class Classification

Initially we built a pixel-wise binary classifier that was trained on images of ships in the open sea. This performed well on the test data that contained only ships and sea, but less so when the images included shoreline (results discussed in Section 5). For the model to be truly versatile at detecting ships, we decided to modify it to perform pixel-wise multi-class classification. So the problem statement was changed from detecting ship vs. no-ship to land vs. sea vs. ship.

For the most part, our binary classifier and multi-class classifier have the same architecture. The differences are between the choice of output activation, the output label shape, and loss function. These differences are highlighted in the following discussion where they arise.

4.2.2 Network Architecture

The network architecture is illustrated in Figure 4.4. It consists of a contracting path (left side) and an expansive path (right side). Typical CNN architectures involves increasing the number of feature maps (channels) with each max pooling operation. In our network we decided to keep a constant number of 4 feature maps throughout the network. This choice was motivated by two observations. Firstly, we can allow the network to lose some information after the downsampling layer because the model has access to low level features in the upsampling path. Secondly, in satellite images there is no concept of depth or high-level 3D objects to understand, so a large number of feature maps in higher layers may not be critical for good performance.

The contracting path consists of the repeated application of two BN_CONV_RELU units followed by a 2×2 max pooling operation with stride 2 for downsampling. The BN_CONV_RELU unit consists of batch normalisation layer (BN), a convolution layer (CONV), and a rectified linear unit (ReLU). The expansive path repeatedly concatenates the feature maps from the previous layer with the corresponding cropped feature maps from the contracting path (upsampling) via the CONCAT layer, followed by application of the BN_CONV_RELU and BN_UPCONV_RELU

units. The BN_UPCONV_RELU unit is similar to the BN_CONV_RELU unit except that it has a deconvolution layer (UPCONV) instead. At the final layer a 1x1 convolution (CONVOUT) is used to map each 4-component feature vector to a 1-component segmentation mask and the activation function appropriate to the classification task is applied. These individual network units and their parameters are discussed in the following sections.

Network Input

During training, the CNN was fed with mini-batches of 10 images, each image being an array of size $128 \times 128 \times 2$. The first channel corresponds to the Intensity_VH values and the second to Intensity_VH.

The frequency of pixels labelled as ship is naturally quite small in our dataset, compared to the pixels labelled as ships or land. This could be problematic when we train one-vs-all classifiers for land, sea and ship (as we do indeed later in the project). One-vs-all classifier for a rare class essentially has fewer data points to train, compared to other classes. The predictions from such a classifier might be weaker than those of a classifier corresponding to frequently occurring class, which makes it difficult to combine the classifiers for a multi-class classification task. We tried to mitigate this effect by training the network only on those images that contained at least one ship, and by filtering out images that contained more than 50% of the pixels classed as land.

Batch Normalisation

Batch Normalisation applies a transformation $\mathrm{BN}_{\gamma,\beta} : x_{1\ldots m} \rightarrow y_{1\ldots m}$, shown in Equation 4.1, over a mini-batch $B = \{x_{1\ldots m}\}$ that maintains the mean activation close to 0 and the activation standard deviation close to 1. It has been shown that by making normalisation a part of the model architecture and performing the normalisation for each training mini-batch, networks can be trained at much higher learning rates without the risk of divergence and less care is required in initialization.

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B + \varepsilon}} + \beta \tag{4.1}$$

where:

$\mu_B =$ mini-batch mean
$\sigma_B =$ mini-batch variance

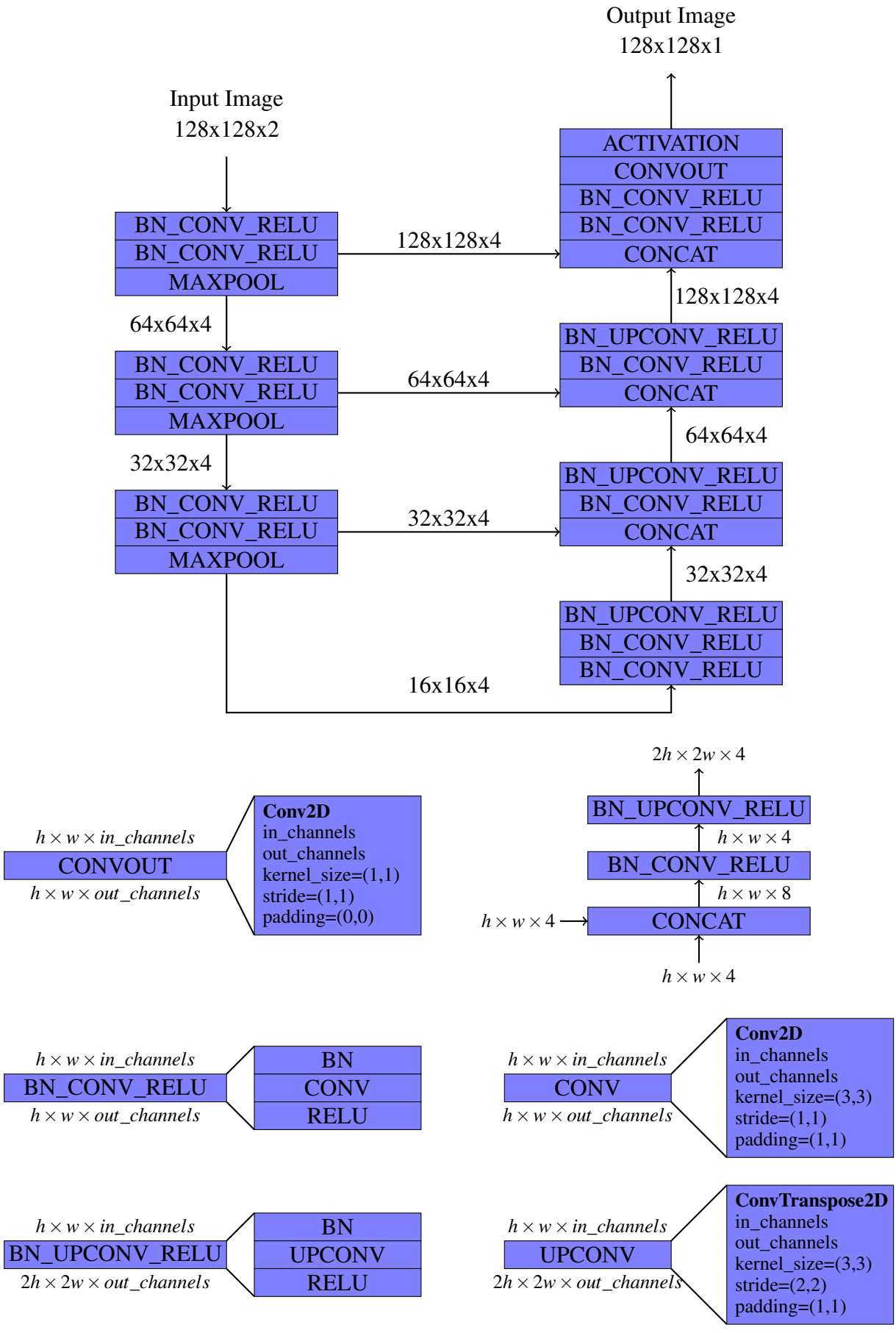


Fig. 4.4 CNN Architecture

2D Convolution

The convolution operation forms the basis of CNNs. It is sparse (only a few input units contribute to a given output unit) and reuses parameters (the same weights are applied to multiple locations in the input). In a hierarchical structure consisting of convolution operations at each layer, this sparse property of convolution becomes very useful as local features in one layer contribute to only a few output units in the next layer, leading to abstraction of information in the higher layers. Re-usability of parameters allows CNN to be trained much more efficiently than equivalent fully connected networks for the same task.

We use the Keras Conv2D function to specify the parameters of the convolution operation. A 3 by 3 convolution kernel size is used, which is small enough to be sensitive to local features such as edges. The number of output features is determined by the number of convolution filters we use at a given layer. Since we want to maintain 4 features throughout the network, the number of convolution filters used at each layer is set to 4. The convolution filter stride is set to 1, which means that the output is computed each time the kernel moves across the input image by one pixel. While convolving a kernel generally decreases the output size with respect to the input size, we are able to preserve the $h \times w$ dimensions by applying unit zero-padding at the image edges, such that the output of convolution is the same size as the input.

Activation Function

Following each convolution and deconvolution operation is a non-linear activation function. The choice of activation function involves several considerations such as propensity to saturate during training, computation cost, and output range. Our choice of Rectified Linear Unit (ReLU) was motivated by the fact that it has a linear, non-saturating form $f(x) = max(0,x)$. Compared to tanh/sigmoid functions that involve expensive operations (exponentials, etc.), ReLUs can be implemented by simply thresholding a matrix of activations at zero. Furthermore, ReLUs have been found to greatly accelerate (e.g. by a factor of 6 in [12]) the convergence of stochastic gradient descent compared to the sigmoid/tanh functions.

Max Pooling

Pooling the activation output after each convolution layer introduces shift invariance, and allows features to vary spatially and still contribute to the same neuron in the next layer. Its function is to progressively reduce the spatial size of the representation, to reduce the number of parameters and amount of computation in the network, and hence to also control overfitting. It also allows higher-level features to incorporate information from a wider domain. We perform pooling using the Max operation: filters of size 2×2 are applied with a stride of 2 which downsample every depth slice in the input by 2 along both width and height, discarding 75% of the activations.

2D Transpose Convolution

The 2D Transpose Convolution layer performs the inverse operation of the 2D Convolution layer. Not to be confused with deconvolution in signal processing, the transposed convolution achieves the inverse by applying a convolution kernel to a zero-padded image, much in the same way as 2D Convolution, but while the 2D Convolution operation reduces the $width \times height$ of the image, the 2D Transpose Convolution expands it, given the appropriate stride and zero-padding are set. In our model, the 2D Transpose Convolution is done with a 3×3 kernel, stride of 2 and unit zero-padding.

Network Output

For the binary classifier, the *out_channels* for the CONVOUT layer is set to 1. The per-pixel probability $p(\mathbf{x})$ is provided by the sigmoid function, given in Equation 4.2. The network output from the binary classifier has the shape $128 \times 128 \times 1$.

$$p(\mathbf{x}) = \frac{1}{1+\exp(-a(\mathbf{x}))} \tag{4.2}$$

where $a(\mathbf{x})$ denotes the activation at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \in \mathbb{Z}^2$.

For the multi-class classifier, the *out_channels* for the CONVOUT layer is set to $K = 3$, such that activations $a_k(\mathbf{x})$ in k^{th} channel correspond to either land, sea, or ship. The per-pixel probability $p_k(\mathbf{x})$ for class k is provided by the softmax function, given in Equation 4.3. The network output from the multi-class classifier has the shape $128 \times 128 \times 3$.

$$p_k(\mathbf{x}) = \frac{\exp(-a_k(\mathbf{x}))}{\sum_{k=1}^{K} \exp(-a_k(\mathbf{x}))} \tag{4.3}$$

where $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \in \mathbb{Z}^2$.

4.2.3 Training

Weight Initialisation

Initialising network weights for training deep neural networks can be a challenge in itself. Improper initialisation of weights can lead to saturation of non-linear activation functions and stall learning. Indeed it has been observed that random initialisation of deep networks with sigmoid activation functions drives the top hidden layer into saturation [35]. The same paper suggests initialising the weights of each layer such that the first and second moment statistics of the inputs and outputs of that layer are preserved. Consider the variance of weights w_i in the

case of 1D inputs x_i and 1D outputs y_i (bias term is ignored here as it is constant and has zero variance):

$$\begin{aligned} var(y) &= var\left(\sum_{i=0}^{N} w_i x_i\right) \\ var(y) &= N var(w_i x_i), \text{ identically distributed} \\ var(y) &= N var(w_i) var(x_i), \text{ inputs are independent of weights} \end{aligned} \tag{4.4}$$

Hence, for the variance of outputs to equal the variance of inputs, the initial weights should be drawn from a distribution with a variance $var(w_i) = 1/N$, where N is the number of input neurons. We use Xavier normal initialisation in which the weights are drawn randomly from a zero-mean normal distribution and variance $var(w_i) = 1/N_{ave}$, where $N_{ave} = (N_{in} + N_{out})/2$. This takes into account both the number of input neurons N_{in} and the number of output neurons N_{out}, and is therefore able to preserve the back-propagated signal as well.

Loss Function

The loss function used in the binary classifier is the binary cross entropy function given in Equation 4.5.

$$E = \sum_{\mathbf{x}\in\Omega} y(\mathbf{x}) \log p(\mathbf{x}) + (1 - y(\mathbf{x})) \log(1 - p(\mathbf{x})) \tag{4.5}$$

where:

$y(\mathbf{x})$ = true label of pixel at position $\mathbf{x}$
$p(\mathbf{x})$ = predicted label of pixel at position $\mathbf{x}$

For the multi-class classifier, we used the categorical cross entropy function given in Equation 4.5.

$$E = \sum_{\mathbf{x}\in\Omega} \log(p_{l(\mathbf{x})}(\mathbf{x})) \tag{4.6}$$

where:

$l(\mathbf{x})$ = true label of pixel at position $\mathbf{x}$, $l : \Omega \to \{1, \ldots, K\}$
$p_{l(\mathbf{x})}(\mathbf{x})$ = predicted label of pixel at position $\mathbf{x}$ corresponding to $l(\mathbf{x})$

Optimizer

The choice of optimization algorithm for deep learning models can mean the difference between good results in minutes, hours, or days. We use the Adam optimiser [36] which is based on adaptive estimates of lower-order moments. As [36] suggests, the method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters.

While the classical Stochastic Gradient Descent maintains a single, constant learning rate (termed alpha) for all parameter updates during training, Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Adam can be fine-tuned through its settings. We use the default settings as given in the paper which have been tested for machine learning problems: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$.

Chapter 5

Results and Discussion

5.1 CFAR

The Constant False Alarm Rate (CFAR) Algorithm was applied on the SAR image shown in Figure 5.1 Left, and the outcome is illustrated in Figures 5.1 Right to 5.2. We observe that CFAR is able to perform well in detecting pixels that represent a ship against a background of pixels representing the sea. However, this algorithm is not able to differentiate between pixels representing ships and pixels representing land, as illustrated by the false ship detections along the edge of land in Figure 5.2 Right.

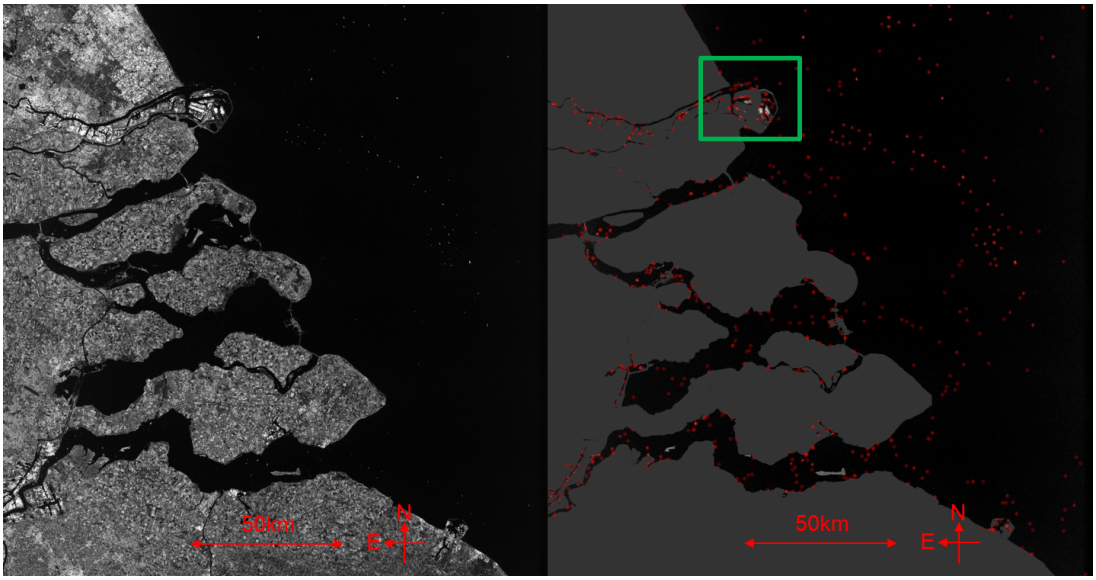


Fig. 5.1 Left: SAR image before CFAR algorithm. Right: CFAR algorithm predictions shown in red. Land was masked out before initiating CFAR to avoid CFAR predictions on pixels representing land.

As previously explained in Section 3, the CFAR algorithm bases its predictions on the comparison between the statistics of the target cell and background pixels(Adaptive Thresholding),

and filters out the predictions it their size is larger or smaller than the given tolerances (Object Discrimination). This approach relies on the assumption that background statistics do not vary too much. While this holds true for SAR images of the sea, the same can not be assumed for land, where adjacent objects might have vastly different radar reflectivities, making it harder to classify them under the same class ('land') using just the first and second order statistics. Land masking was performed using the SNAP toolkit by apply a geo-file containing the terrain information of the corresponding region and masking out points above the sea level. However, the land mask failed to exclude some parts of the port which inadvertently led to CFAR detecting false positives.

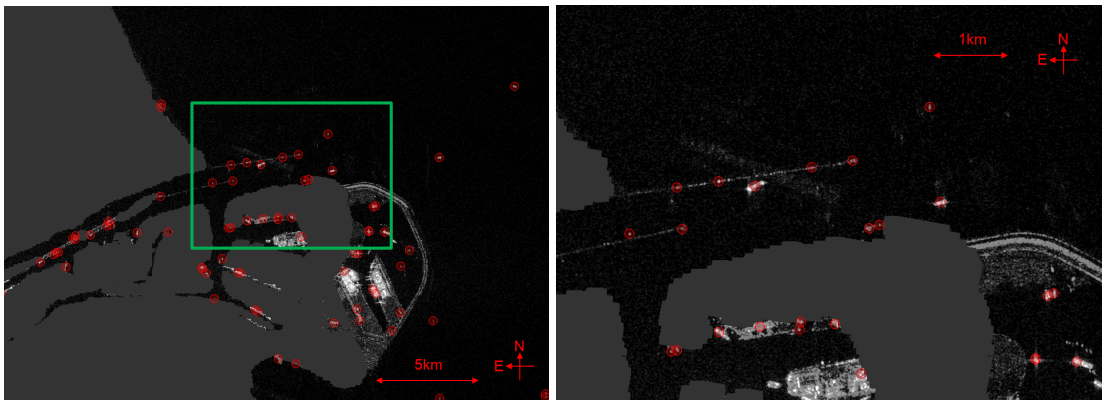


Fig. 5.2 Left: A closer look at the region bounded by green quadrilateral in Figure5.1 Left. Right: Further close-up reveals that the CFAR has detected several ships in the sea, however, it has also detected some false positives in strip of land that had not been masked out.

The confusion matrix given in Table 5.1 was calculated by comparing the number of ships detected by CFAR to those in the labelled dataset. Calculating pixel-wise accuracy does not really make sense here since the most of the land area had been masked out in the pre-CFAR stage. Even if we were to exclude the pixels belonging to the land-class in the labelled dataset, the land masking itself is not perfect and would impact any pixel-wise accuracy metric that we may use. Instead, we simply calculate an accuracy metric (call it α) based on the ratio of true positives p_t to the sum of true positives p_t, false positives p_f and false negatives n_f, given in Equation 5.1. For CFAR results, this metric evaluates to 0.610. In addition to using more sophisticated accuracy metrics on our subsequent models, we will also evaluate this metric on their results for comparison with the results from CFAR.

$$\alpha = \frac{p_t}{p_t + p_f + n_f} \tag{5.1}$$

Table 5.1 Confusion matrix for ship detections based on CFAR predictions.

	True	False
Detected	36	22
Not Detected	-	1

5.2 Pixel-wise Classification using CNN

In Section 4 we had introduced two types of model architectures: the Binary Classifier and the Multi-Class Classifier, which had largely the same architecture expect for a few differences in the output and training criterion. We have used these models in the following three classification exercises; more specifically, Binary Classifiers were used for the binary classification task in Section 1.2.1 and also in Section 1.2.2 where the multi-class classification task was reduced to three separate one-vs-all binary classification tasks. A Multi-Class Classifier was also trained in Section 1.2.2 and its performance compared to the One-vs-All approach.

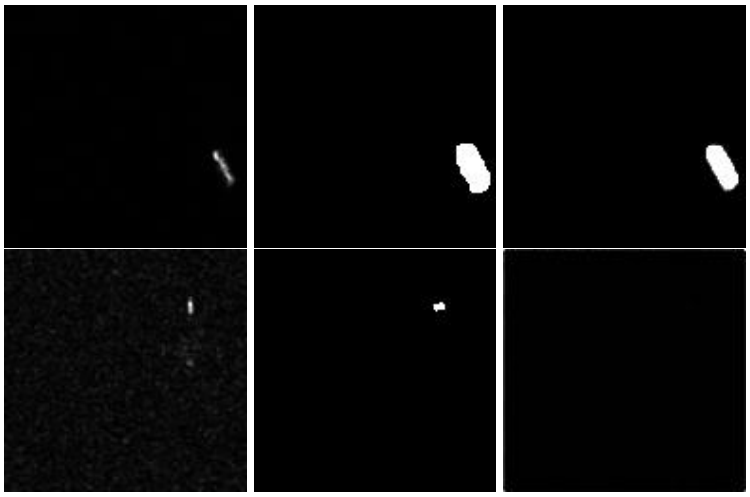
5.2.1 Binary Classification: Ship vs Sea

A Binary Classifier was trained to discriminate between the pixels belonging to the sea-class and pixels belonging to the ship-class. The training and test data consisted only of images with pixels representing sea or ship. Images with pixels belonging to the land-class were not shown to the network during training. The network was trained on 1440 images and tested on 160 images (90:10 split). The network predictions from a subset of the test images are illustrated in Figure 5.3 and the number of ship detections are given in Table5.2. Of the 160 test images, 6 had instances of ship and the Binary Classifier detected them correctly. For the other 154 test images, the Binary Classifier did not predict a single ship, yielding an α accuracy of 100%.

Table 5.2 Confusion matrix of ship detections. The Binary Classifier was fed with test data representative of only two classes: sea and ships.

	True	False
Detected	6	0
Not Detected	154	0

The next step was to run this trained Binary Classifier on test data containing land-class pixels. The network predictions from a subset of the test images are illustrated in Figure 5.4 and the number of ship detections are given in Table5.3. Interestingly, the classifier produced a false positive when it encountered an image with pixels corresponding to land-class. This case is illustrated in the second row in Figure 5.4. There were false negatives as well: cases where the ship pixels were classed as sea, illustrated in fourth row in Figure 5.4.



(a) (b) (c)

Fig. 5.3 Predictions from Binary Classifier: $p(x) \approx 1$ corresponds to ship-class and $p(x) \approx 0$ corresponds to sea-class. (a) Intensity_VH feature map of the input SAR image. (b) True labels in the form of a binary mask. (c) Predicted labels in the form of a greyscale image.

Table 5.3 Confusion matrix of ship detections. The Binary Classifier was fed with test data representative of all three classes: land, sea, and ships.

	True	False
Detected	12	1
Not Detected	17	4

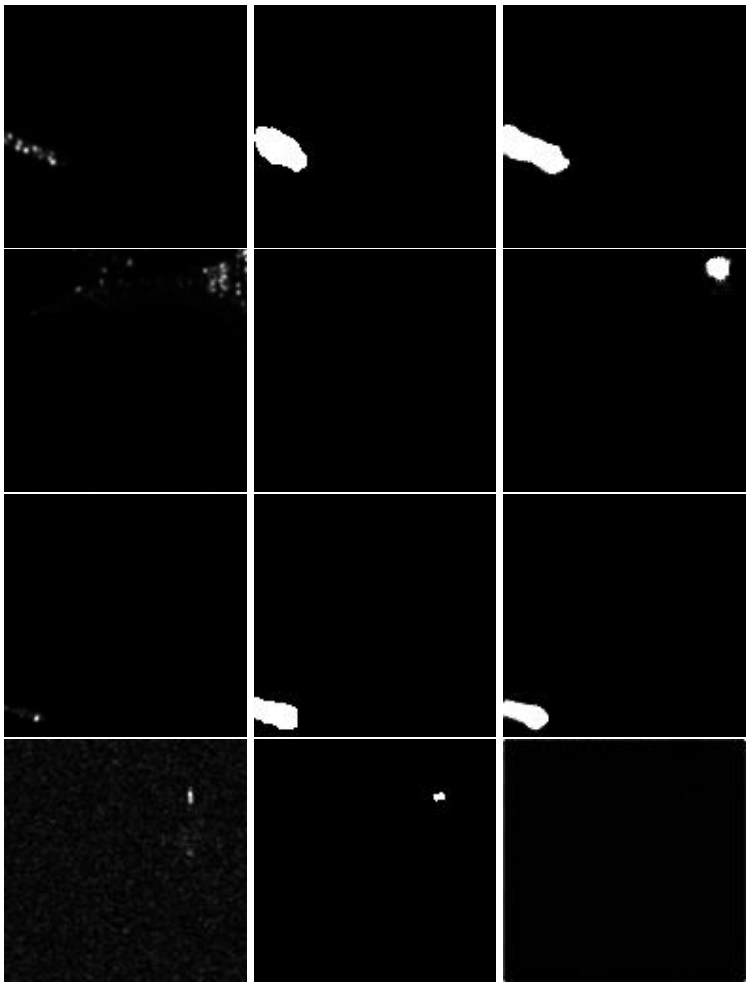
5.2.2 Multi-class Classification: Land vs Sea vs Ship

Next, we looked at training our model to perform multi-class classification rather than just binary classification as in the previous section. There were two approaches we considered: 1) training 3 individual one-vs-all classifiers for each class separately and combining their outputs on test data, and 2) training a single multi-class classifier.

The classifiers were tested on the same test data of 34 images as the previous Binary Classifier was tested on to allow us to compare the results and prediction accuracy. The accuracy metrics used, in addition to α are the binary accuracy for the one-vs-all classifier and the categorical accuracy for multi-class classifier. The binary accuracy rounds the predicted mask pixel values to 1 or 0 (threshold of 0.5) and finds the mean of the intersection of with the true labels, given in Equation 5.2.

$$\text{binary\_accuracy} = \text{mean}(y_{true} \ \ \text{AND} \ \ \text{round}(y_{pred})) \tag{5.2}$$

The categorical accuracy finds the class with the highest probability for each pixel and evaluates the mean of the intersection with the label mask, given in Equation 5.3. The -1 axis refers the last axis in the output image which is the feature axis.



(a) (b) (c)

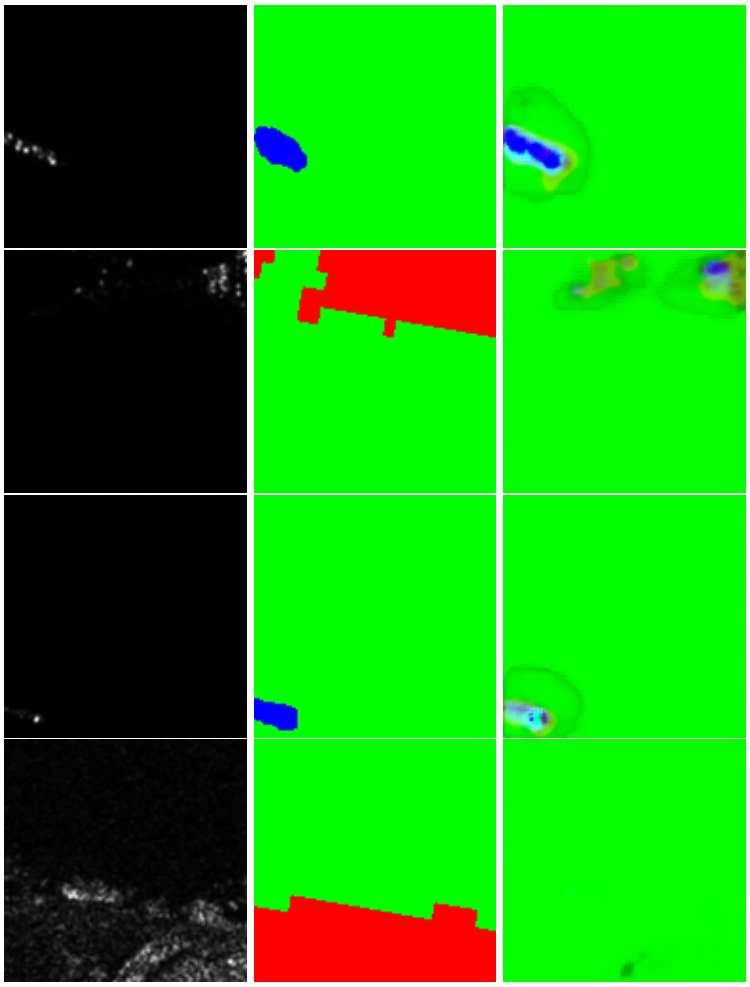
Fig. 5.4 Predictions from Binary Classifier for test data containing land-class pixels as well, in addition to sea-class and ship-class pixels as before. (a) Intensity_VH feature map of the input SAR image. (b) True labels in the form of a binary mask: mask pixel value of 1 represents a ship, and mask pixel value of 0 represents 'not ship'. (c) Predicted labels in the form of a greyscale image.

$$\text{categorical\_accuracy} = \text{mean}(\text{argmax}(y_{true}, \text{axis} = -1), \ \ \text{AND} \ \ \text{argmax}(y_{pred}, \text{axis} = -1)) \tag{5.3}$$

One-vs-all Classifier

The one-vs-all is essentially a binary classifier. The only difference is that the training labels for each classifier are different, e.g. the classifier for land class is trained on a binary mask representing land and not-land classes, and the binary masks for the other two classes will be differ likewise. Each test images is fed separately to each of the three classifiers and their output

is concatenated into a three-channel array. The value $p_k(\mathbf{x})$ represents the probability that the input image pixel at location $\mathbf{x}$ belongs to class k.



(a) (b) (c)

Fig. 5.5 Combined test predictions from the one-vs-all classifiers for test data containing all three classes of pixels. (a) Intensity_VH feature map of the input SAR image. (b) True labels in the form of an rgb mask: red represents pixels belonging to land-class, green represents pixels belonging to sea-classand and blue represents pixels belonging to ship-class. (c) Predicted labels also in the form of an rgb mask with same colour coding.

The network predictions from a subset of the test images are illustrated in Figure 5.5. The classifier correctly classifies ships in the input image but struggles with classifying land, as evident in rows 2 and 4 of Figure 5.5.

The binary accuracy achieved during training and testing for each individual classifier is given in Table 5.4. The confusion matrix for the combined test predictions is given in Table 5.5. The α accuracy metric evaluates to 0.889 for the test data, which is higher than both CFAR and Binary Classification considered previously.

Table 5.4 One-vs-All Classifiers for each of the three classes: land, sea, and ship. The classifiers were trained and tested separately. The accuracy metric was evaluated using binary_accuracy.

		Accuracy	
		Train	Test
One-vs-All	Land	0.9747	0.9773
	Sea	0.9743	0.9730
	Ship	0.9971	0.9970

Table 5.5 Confusion matrix for ship detections based on the combined test predictions from the One-vs-All Classifiers.

	True	False
Detected	16	1
Not Detected	16	1

Multi-Class Classifier

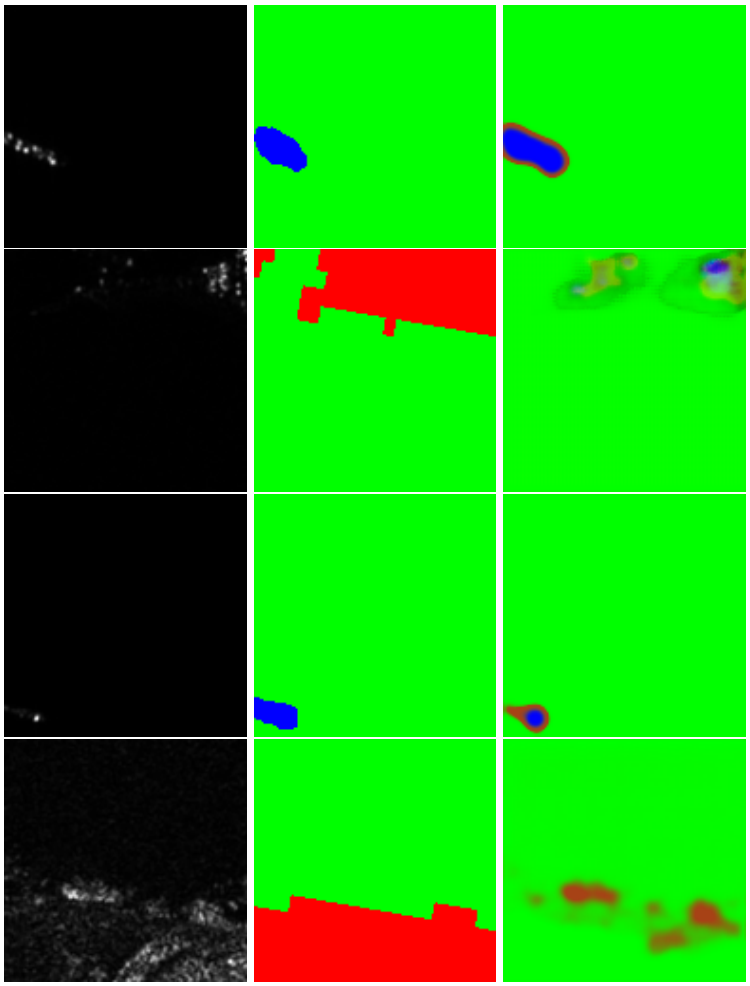
We next trained the multi-class classifier. This classifier is different from the previous binary classifiers we have evaluated so far in the sense that it directly outputs a three-channel prediction mask where the mask value $p_k(\mathbf{x})$ represents the probability that the input image pixel at location $\mathbf{x}$ belongs to class k. It does so by incorporating a softmax function at the output layer (refer to model architecture details in Section 4).

The network predictions from a subset of the test images are illustrated in Figure 5.6. The multi-class classifier seems to be less confident than the one-vs-all classifiers when classifying ships, as evident from the darker shade of blue in the predicted masks (lighter blue represents higher probability). However, it is seems to be better at classifying pixels that correspond to land, as evident from the lager patches of red in the predicted mask at pixel locations where land-class is expected.

Table 5.6 Summary of accuracy achieved with during training and testing, and the α metric applied to each model.

		Accuracy		α
		Train	Test	
CFAR		-	-	0.610
Binary		0.9996	0.9995	0.706
	Land	0.9747	0.9773	
One-vs-All	Sea	0.9743	0.9730	0.889
	Ship	0.9971	0.9970	
Multi-Class		0.9857	0.9766	0.706

The categorical accuracy achieved during training and testing for the multi-class classifier is given in Table 5.6, along with a summary of the accuracies of all the prior models. While the training and testing accuracies of the multi-class classifier are slightly lower than those of the



(a) (b) (c)

Fig. 5.6 Test predictions from the multi-class classifiers for test data containing all three classes of pixels. (a) Intensity_VH feature map of the input SAR image. (b) True labels in the form of an rgb mask: red represents pixels belonging to land-class, green represents pixels belonging to sea-classand and blue represents pixels belonging to ship-class. (c) Predicted labels also in the form of an rgb mask with same colour coding

one-vs-classifier for ships, they are still higher than those of one-vs-classifier for land and sea. This confirms our observation the multi-class classifier performs better than one-vs-all approach for the land-class but slightly worse than for the ship-class.

The confusion matrix for the test predictions is given in Table 5.7. The α accuracy metric evaluated based on this matrix is 0.706. This is lower than the metric we achieved in the one-vs-all approach and equals that of the binary classification approach, but it is still better than CFAR.

While the one-vs-all approach results in better metrics, the multi-class classifier has almost the same metrics and in fact performs better if we consider accuracy on all three classes not just on ship-class; our metric α has been biased towards accurate detection of ships (and rightfully so

Table 5.7 Confusion matrix for ship detections based on test predictions from the Multi-Class Classifier.

	True	False
Detected	12	0
Not Detected	17	5

for this project), but if our aim was to come with an optimum classifier for a three class problem, the multi-class classifier would be the preferred option.

Chapter 6

Conclusions and Future Work

The primary objective of this project was to develop and evaluate the performance of Neural Network models on marine vessel detection from SAR data. In pursuing this objective, we were faced with the challenge of having to create our own labelled SAR dataset, and it was through our investigations that we found that while the AIS data could be an effective ground truth, it can not be used in its raw form and we need to develop a model first that addresses the unreliability with the AIS data timestamps. Perhaps a probabilistic model that takes the marine vessels heading and trajectory into account and infers the approximate position of the vessel at each time step might provide some breakthrough in this regard. The effectiveness of such a probabilistic approach in fusing the AIS and SAR data is difficult to judge at the moment and requires further exploration.

The CNN architecture that we developed performed really well in both binary and multi-class classification tasks; consistently achieving over 97% accuracy for both training and testing data. So far we have considered pixel-wise classification for ship detection. To extend this further to ship identification would require the model to contextualize the pixel-wise predictions and would need to train on supplementary data such as ship dimensions, material reflectivity, etc..

References

[1] Ahmed Hassanin, Francisco Lazaro, and Simon Plass. An advanced AIS receiver using a priori information. In *MTS/IEEE OCEANS 2015 - Genova: Discovering Sustainable Ocean Energy for a New World*, pages 1–7, 5 2015.

[2] Fabio Mazzarella, Michele Vespe, and Carlos Santamaria. SAR Ship Detection and Self-Reporting Data Fusion Based on Traffic Knowledge. *IEEE Geoscience and Remote Sensing Letters*, 12(8):1685–1689, 8 2015.

[3] Stephan Brusch, Susanne Lehner, Thomas Fritz, Matteo Soccorsi, Alexander Soloviev, and Bart Van Schie. Ship surveillance with TerraSAR-X. *IEEE Transactions on Geoscience and Remote Sensing*, 49(3):1092–1103, 5 2011.

[4] R. Grasso, S. Mirra, A. Baldacci, J. Horstmann, M. Coffin, and M. Jarvis. Performance assessment of a mathematical morphology ship detection algorithm for SAR images through comparison with AIS data. In *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, pages 602–607, 2009.

[5] Wolfgang Middelmann, Alfons Ebert, and Ulrich Thoennessen. Automatic Target Recognition in SAR Images Based on a SVM Classification Scheme. *Framework*, pages 492–499, 2007.

[6] Orsan Aytekin, Mehmet Koc, and Ilkay Ulusoy. Local primitive pattern for the classification of SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 51(4):2431–2441, 4 2013.

[7] Seisuke Fukuda and Haruto Hirosawa. A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5 I):2282–2286, 1999.

[8] Xin Zhan, Rong Zhang, Dong Yin, and Chengfu Huo. SAR image compression using multiscale dictionary learning and sparse representation. *IEEE Geoscience and Remote Sensing Letters*, 10(5):1090–1094, 9 2013.

[9] Shuyuan Yang, Yueyuan Zhang, and Yue Han. Speckle reduction of SAR image through dictionary learning and point target enhancing approaches. In *Proceedings of 2011 IEEE CIE International Conference on Radar, RADAR 2011*, volume 2, pages 1926–1929. IEEE, 10 2011.

[10] Ganggang Dong, Gangyao Kuang, Na Wang, and Wei Wang. Classification via sparse representation of steerable wavelet frames on grassmann manifold: Application to target recognition in SAR image. *IEEE Transactions on Image Processing*, 26(6):2892–2904, 6 2017.

[11] Corneliu Octavian Dumitru and Mihai Datcu. Information content of very high resolution SAR images: Study of feature extraction and imaging parameters. *IEEE Transactions on Geoscience and Remote Sensing*, 51(8):4591–4610, 8 2013.

[12] Alex Krizhevsky, Ilya Sutskever, and G E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS 2012)*, NIPS'12, page 4, USA, 2012. Curran Associates Inc.

[13] Vasuki Perumal, P Vasuki, S Mohamed, and Mansoor Roomi. Automatic Target Classification in SAR Images by Multilayer Back propagation Neural Network. *Research Journal of Applied Sciences, Engineering and Technology*, 4(24):5510–5514, 2012.

[14] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 9 2014.

[15] MSTAR Overview.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. 5 2015.

[17] Fei Gao, Zhenyu Yue, Jun Wang, Jinping Sun, Erfu Yang, and Huiyu Zhou. A Novel Active Semisupervised Convolutional Neural Network Algorithm for SAR Image Recognition. *Computational Intelligence and Neuroscience*, 2017:35–45, 2017.

[18] D Malmgren-Hansen and M Nobel-J???rgensen. Convolutional neural networks for SAR image segmentation. In *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 231–236, 12 2015.

[19] M Gong, J Zhao, J Liu, Q Miao, and L Jiao. Change Detection in Synthetic Aperture Radar Images Based on Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 27(1):125–138, 1 2016.

[20] Sizhe Chen, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Target Classification Using the Deep Convolutional Networks for SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4806–4817, 8 2016.

[21] Open Access Hub.

[22] Gregory J Power and Robert A Weisenseel. ATR subsystem performance measures using manual segmentation of SAR target chips. *Proc. SPIE 3721, Algorithms for Synthetic Aperture Radar Imagery VI, 685*, 1999.

[23] Elmehdi Aitnouri, Shengrui Wang, and Djemel Ziou. Segmentation of Small Vehicle Targets in SAR Images. *International Society for Optics and Photonics*, 4726:35–45, 2002.

[24] Synthetic Aperture Radar - Space Program - Bedford Astronomy Club.

[25] p46.jpg (616×810).

[26] User Guides - Sentinel-1 SAR - Acquisition Modes - Sentinel Online.

[27] Francesco De Zan and Andrea Monti Guarnieri. TOPSAR: Terrain observation by progressive scans. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2352–2360, 9 2006.

[28] International Convention for the Safety of Life at Sea (SOLAS) Convention Chapter V, Regulation 19 - Carriage requirements for shipborne navigational systems and equipment, International Maritime Organization. 2004.

[29] Constant False Alarm Rate (CFAR) Detection - MATLAB & Simulink - MathWorks United Kingdom.

[30] CFARGaussian.png (561×420).

[31] Copernicus Open Access Hub.

[32] Research and User Support.

[33] SNAP | STEP.

[34] Danish Maritime Authority-AIS.

[35] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014.

Appendix A

GitHub Repository

Please visit the following GitHub repository for the project source code and electronic copy of the logbook. https://github.com/saaw2/IIBProject_1718

Appendix B

Risk Assessment

The Risk Assessment Form was submitted to the Department Safety Office in week 1 of Michaelmas term 2017, before the project started. Given the nature of the project, the only potential hazard identified on the Form was computer use:

'Potential ergonomic concerns from extensive use of computers. Care should be taken to use correct posture and to take breaks regularly.'

This was indeed the main health and safety issue during the entire course of the project. The author has utilised the hot desks in the Signal Processing lab where adjustable LCD monitors and external keyboards were available. The author has also made use of the computers at Dyson Centre which have wider extended LED screens.

There were no other health and safety issues.