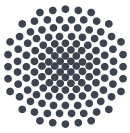


Universität Stuttgart

masterarbeit am ifp

Steffen Merseburg

Convolutional Neural Networks for tiny ship detection in Sentinel 1 SAR images towards their application in civil Search and Rescue

Betreuer: Prof. Dr.-Ing. Uwe Sörgel
Prüfer: Prof. Dr.-Ing. Uwe Sörgel

Erklärung

Hiermit versichere ich, Merseburg, Steffen, dass ich diese Masterarbeit selbstständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit oder wesentliche Bestandteile davon sind weder an dieser noch an einer anderen Bildungseinrichtung bereits zur Erlangung eines Abschlusses eingereicht worden. Ich erkläre weiterhin, bei der Erstellung der Arbeit die einschlägigen Bestimmungen zum Urheberschutz fremder Beiträge entsprechend den Regeln guter wissenschaftlicher Praxis1 eingehalten zu haben. Soweit meine Arbeit fremde Beiträge (z.B. Bilder, Zeichnungen, Textpassagen etc.) enthält, habe ich diese Beiträge als solche gekennzeichnet (Zitat, Quellenangabe) und eventuell erforderlich gewordene Zustimmungen der Urheber zur Nutzung dieser Beiträge in meiner Arbeit eingeholt. Mir ist bekannt, dass ich im Falle einer schuldhaften Verletzung dieser Pflichten die daraus entstehenden Konsequenzen zu tragen habe. Nach Abschluss der Arbeit werde ich zu diesem Zweck meinem Betreuer neben dem Prüfexemplar eine weitere gedruckte sowie eine digitale Fassung übergeben.

.....

Ort, Datum, Unterschrift

Zusammenfassung

Im Mittelmeer sterben jährlich tausende Menschen auf der Flucht nach Europa, somit ist diese Region eine der tödlichsten Grenzen der Welt. NGOs versuchen die humanitäre Krise abzumildern, indem sie mit Rettungsschiffen und Flugzeugen versuchen Menschen in Seenot zu finden und ihnen zu Helfen. Die NGO Space Eye e.V. arbeitet u.a. daran Satelliten Daten für die zivile Seenotrettung zu verwenden. Satelliten Bilder geben einen exzellenten Überblick über die Situation, allerdings entsteht durch das Versenden und Verarbeiten der Daten eine zeitliche Verzögerung. Dennoch nutzt die europäische Grenzschutzagentur Frontex zusammen mit der EMSA bereits erfolgreich Satelliten Bilder, um Menschen in Seenot zu finden. Viele Forscher haben Algorithmen und Frameworks entwickelt, um Boote verlässlich und schnell zu entdecken. Die Boote, die zur Flucht über das Mittelmeer zum Einsatz kommen, sind häufig klein und daher schwer zu entdecken. Diese Arbeit untersucht die Anwendungsmöglichkeiten von Sentinel-1 Bildern für die zivile Seenotrettung. Hierzu wird ein Datensatz mit einer neuen skalierbaren Methodik erstellt, eine spezielle Metrik eingeführt und spezialisierte Convolutional Neural Networks trainiert. Diese werden anschließend mit dem aktuellen Stand der Forschung verglichen. Die besten Modelle waren TinyNet3 und der traditionelle Algorithmus MMSE PWF. Diese wurden genauer getestet, um die Limitierungen für eine erfolgreiche Detektion von kleinen Schiffen zu untersuchen. Die beiden stärksten Faktoren waren dabei die Länge der Schiffe und die Wellenhöhe. Da der erstellte Datensatz einen Bias enthält, fiel der Zusammenhang zum Einfallswinkel deutlich geringer aus. TinyNet3 erreicht eine Detektierbarkeit von Schiffen zwischen 15 und 20 Metern von über 90%, was wesentlich höher ist als der bisherige Forschungsstand. Tests mit Bildern von bekannten Seenotrettungsfällen in der Vergangenheit ergaben, dass die Kombination von traditionellen Algorithmen mit Machine Learning Modellen brauchbare Ergebnisse liefert und ermöglicht, diese Fälle mit Satellitenbildern zu untersuchen. Darüber hinaus wurde getestet, ob die Algorithmen schnell genug sind, um komplette Bilder zu analysieren. TinyNet3 erreicht dabei 2586 Bilder pro Sekunde und MMSE PWF nur 96 Bilder pro Sekunde. Damit wäre TinyNet3 grundsätzlich gut geeignet um Sentinel-1 Bilder für die Seenotrettung zu untersuchen, jedoch muss mit einem größeren Datensatz trainiert werden um die Verlässlichkeit zu erhöhen.

Abstract

The Mediterranean Sea is known to be the deadliest border in the world, where thousands of people die every year. NGOs are trying to counter this ongoing humanitarian crisis with ships and airplanes equipped for Search and Rescue operations. The NGO Space-Eye e.V. is working on adding satellite images to the civil Search and Rescue efforts. Using satellite imagery gives an excellent overview of the situation but lacks real-time capabilities and detail. However, the European border and security agency Frontex is already using satellite images to spot people in distress together with the EMSA. Many researchers are tackling the issue of detecting particular weak targets like the small boats used by refugees and designing algorithms and frameworks to use them in Search and Rescue. This study explores using Sentinel-1 images and CNNs for their application in this context. A novel, scaleable classification datasets with AIS ship positions, a specialized metric and a series of specialized lightweight CNN architectures for tiny object detection are derived and tested against state-of-the-art algorithms. The best-performing models were the newly designed TinyNet3 and MMSE PWF. They were tested on the dataset to derive the limiting factors for the detectability of ships, with the most notable ones being the wave height and the ship's length. Because of a bias in the dataset, the incident angle appears not to play a noticeable role. The detectability of ships with a length between 15 and 20 meters was above 90%, which is noticeably higher than the current state-of-the-art. When testing the models on known past cases of refugee boats, the results are only conclusive when different models are combined. The CNN model outperforms the traditional algorithm in execution time with 2586 frames per Second against 96 frames per second. The initial results are promising for future deployment in Search and Rescue, but additional data and evaluation is recommended.

Contents

List of Figures

List of Tables

Acronyms

AIS	automatic identification system
AOI	area of interest
AUC	area under curve
CAM	class activation map
CFAR	constant false alarm rate
CNN	convolutional neural network
DPolRAD	dual-pol ratio anomaly detector
EMSA	European Maritime Safety Agency
FLOPs	floating point operations
FPR	false positive rate
FPS	frames per second
GRD	ground range detected
HH	horizontal-transmit-horizontal-receive
HV	horizontal-transmit-vertical-receive
IW	Interferometic Wide swath
MAC	memory access cost
ML	machine learning
MLP	multi-layer perceptron
NGO	non-governmental organization
NIS	normalized intensity sum
PMF	polarimetric match filter
PNF	polarimetric notch filter
PWF	polarimetric whitening filter
ROC	receiver operating characteristic
SAR	synthetic aperture radar
SLC	single look complex
SOG	speed over ground
SSDD	SAR Ship Detection Dataset
TNR	thermal noise removal
TPR	true positive rate
VH	vertical-transmit-horizontal-receive
VV	vertical-transmit-vertical-receive

1 Introduction

The Missing Migrants Project of the International Organization for Migration has counted 28106 dead or missing in the central Mediterranean since 2014, which is likely to be an undercount [1]. Since 2020, the numbers have risen again, with no end to the humanitarian crisis at Europe's border in sight [1]. That makes the Mediterranean Sea the deadliest border in the world. Refugees fleeing life-threatening conditions like war, discrimination, food shortage, or extreme poverty are forced to pay smugglers and human traffickers to get into a safe country. Often they pay for traveling in small (about 10m long), unseaworthy boats [1], like hand-made metal boats, rubber inflatable boats, coastal fishing boats, decommissioned fishing vessels, or decommissioned sailing boats.

Since 2015, the EU and its member states continued to withdraw from their Search and Rescue duty and funded the so-called Lybian Coastguard instead, which has repeatedly faced scrutiny for human rights violations [2]. Since then, non-governmental organizations (NGOs) like Sea-Watch reported many violent encounters between them and refugees fleeing. For example, some reports show that they rammed refugee boats or performed dangerous maneuvers nearby that caused panic and capsizing of the boats [3]. Civil actors like Sea Watch, RESQSHIP (the author is a member of RESQSHIP e.V.), and a few other organizations filled the Search and Rescue void left behind. These NGOs are present with vessels on the water and airplanes in the sky to spot and rescue people in distress. While these organizations have become very professional, they differ from what an EU-led Search and Rescue mission could accomplish.

Satellite images, for example, are used by the state actors but not yet by NGOs. In 2014, an EU project called SAGRES used Radarsat2 images to detect a 7m long rubber boat, leading to the rescue of 38 survivors [4]. Since then many researchers have been working on this problem using different approaches [5]–[9]. Meanwhile, the European Border and Coast Guard Agency (Frontex) is using the capabilities of the European Maritime Safety Agency (EMSA) and the Copernicus Border Surveillance program to spot refugee boats [10], [11]. Because Frontex does not share the information with the NGOs, the registered association Space-Eye plans to provide the civil actors with satellite

analysis in the future. To date, they only rely on optical images and do not have the capabilities to use synthetic aperture radar (SAR) data like EMSA does. Using the freely available Sentinel-1 images would significantly improve the efforts to support NGOs in their Search and Rescue operations. This study is moving toward this goal by investigating the limits and conditions of ship size detectability in Sentinel-1 images and how this technology can be applied in Search and Rescue.

2 State of the art

This chapter describes the state of the art of refugee boat and general ship detection in remote sensing and its unique challenges. It also covers the necessary basics of SAR and its preprocessing.

2.1 Refugee Boat Detection

As explained in Section 1, the EMSA uses its ship detection capabilities to detect refugee boats. Unfortunately, they do not share details about their technology. The SAGRES report [4], which was working with Frontex, claims to be capable of detecting small targets of 5-10 meters by processing single-channel images. They also refrain from sharing any details on the utilized detection methods. Instead, the report focuses on setup of tasking satellite images to deliver the analysis result to Frontex promptly, which then delimits the search area based on that knowledge. As mentioned, they successfully supported a Search and Rescue mission leading to the rescue of 38 survivors from a seven-meter-long rubber inflatable boat. The boat was spotted in a Radarsat2 image with 3m resolution less than 3 hours after acquisition and was later found by a rescue vessel 14.5 nm away from the location in the satellite image.

Topputo et al. [8] explored how to use satellites for Search and Rescue operations. They proposed a framework to task satellite images with different sensors from different satellite constellations. With such a system a vessel could be detected in more than one image at different points in time. They propose a minimum noise fraction algorithm for optical images to detect a ship followed by segmentation and a threshold rule set. Then, the ship and the wake are individually fitted with an ellipse to derive the vessel's size, course, and speed. For SAR, they proposed CFAR-based algorithms with clustering of the detection points. The ship parameters are then derived from sub-aperture processing and ship-wake displacement. They dismiss more complex approaches like polarimetry for being too costly or time-consuming. This statement might be outdated since computational power is higher nowadays compared to the paper's publication in 2015. The bottleneck for real-time applications is rather

the delivery time of satellite images [4]. The study does not give a detailed description of their performance or success.

Kanjir [9] explored the capabilities of optical Sentinel-2 images to detect refugee boats in the Mediterranean Sea. This study concludes that it will overlook smaller vessels of less than 20m in length but might be helpful when combined with other Search and Rescue methods. A modified Normalised Difference Water Index was used in combination with binominal logistic regression to detect vessels.

Melillos et al. [12] used an adaptive threshold algorithm on Sentinel-1 ground range detected (GRD) images off the coast of Cyprus to detect refugee boats. They compared their detection with reports from refugee boats collected from open-source data. They claim to successfully detect refugee boats without specifying the size of the boats. Their results show more detections than just the refugee boat, which the paper does not address further.

Lanz et al.[5]–[7] conducted a considerable in-situ study about the detectability of rubber inflatable boats on a lake in Germany. The boat was fixed at a specific position and orientation when different sensors and acquisition modes were tasked to take images of the lake. They tested different algorithms and designed new ones around their experiments. Furthermore, they simulated different wave conditions by pasting the boat in other scenes from the ocean. The study showed that detecting the 12 by 3.5 meter boat in different high-resolution images with a near-perfect area under curve (AUC) of 0.94 is possible. For lower-resolution Sentinel-1 Interferometic Wide swath (IW) images, they report a much lower performance and cannot detect the boat in all images. Their findings show a performance drop at 2.5-meter high waves [7]. Real images of boats in the open sea are not yet available.

2.2 Ship Detection

Besides Search and Rescue, ship detection is of interest for other reasons such as fishery control, pollution control or border surveillance, customs, and law enforcement [13]. Especially SAR images gathered special attention as most ships are made of metal and provide a clear signal. Hence, many researchers

addressed the broader term of ship detection in remote sensing. The following chapter picks up a few of them.

Torres et al. [14] published a paper about the Sentinel-1 mission stating that it is particularly suitable for an emergency response. However, they projected that the probability of ship detectability is 90% for ships between 25-34m without specifying the probability of false alarm. The limiting factors are the incident angle, clutter limit for the co-polarization channel and the noise floor in the cross-polarization channel. These theoretical values are based on algorithms that only consider intensity and dismiss geometric features or polarimetric methods.

Paolo et al. [15] used Sentinel-1 GRD images to find fishing boats and reveal industrial fishing activities worldwide. They used the CFAR algorithm as a first step and a convolutional neural network (CNN) as a second step to classify what kind of boat is present and if it is engaged in fishing. They report a detectability of 70% for vessels between 15-20 meters in length. However, the parameters needed to be tweaked for different time intervals to obtain this result. They do not provide a detailed description of the limiting factors nor the number of false alarms, as the focus was on revealing fishing activity that has not yet been tracked. Their findings are summarized in a publicly available map.

Bentes et al. [16] examined the theoretical limit of ship size detectability in TerrarSAR-X images. They concluded that wind direction and incident angle are the driving factors in ship detectability. They state that ships with half the length of the resolution cell should be detectable. However, this study is purely theoretical and focuses on the CFAR algorithm on X-Band radar images.

Different polarimetric algorithms were proposed to enhance the performance of the CFAR algorithm and improve the detection of ships [17], [18]. They utilize more information from the image but still do not take the geometric features into account. Some of these traditional methods are explained in greater detail in Section 3.3.

With the rise of machine learning (ML) and CNNs, many researchers have adopted the techniques to detect ships in remote sensing images. Li et al. [19] summarize the effort to use ML for SAR ship detection in a literature review

of 177 papers. They point out that the traditional algorithms might be robust when well-tuned to the ship size and number of ships in a scene but cannot generalize well. On the other hand, ML methods achieve better performance and do not need much preprocessing like land masking. They also looked at real-time algorithms and found high-performing examples with an average precision of 97.2% at 50% intersection over union. For future research, they are pointing out what the next challenges might be; among them are:

- training from scratch
- small ship detection
- real-time detection

The challenges of detecting small targets and being able to compute in real-time are general problems in remote sensing and ML.

Pawlowski et al. [20] created a dataset with tiny objects based on MNIST and explored what factors influence the performance of ML methods. They found that the dataset size scales with the inverse of the object-to-image ratio. The less space the object occupies, the bigger the dataset has to be. Also, the global pooling methods play a significant role in the performance with very low signal-to-noise ratios, with max pooling being more robust than average pooling. Adapting the receptive field to the object size increased the performance further, which was also found by Pang et al. [21]. The last thing they found is that larger capacity models exhibit better generalization. This is in conflict with the need for real-time algorithms.

Pang et al. [21] are addressing the need for real-time algorithms and propose a new backbone model called TinyNet which in combination with a global attention block showed better performance in optical images compared to state-of-the-art detectors.

Radosavovic et al. [22] worked on lightweight model design and methods to find suitable architectures. They narrowed down the design space to make the search more efficient and came up with a series of well-performing networks called RegNet. They summarized their findings and derived a set of guidelines for architectural design spaces.

Ma et al. [23] examined what factors influence the computation time of a model beyond the floating point operations (FLOPs). They conducted a series of experiments and derived guidelines. With those guidelines in mind, they proposed a new architecture called ShuffleNet V2.

2.3 Dataset

Suitable datasets are crucial for ML, and more high-quality datasets for ship detection in SAR images have been published over the last eight years. They are often built using automatic identification system (AIS) data [24]–[26]. AIS is a system ships use to communicate static information, like their IMO registration number, dynamic information like navigational status, and voyage-related information like position, speed over ground (SOG), and heading to other ships in the area for safety reasons [27]. Only passenger ships and cargo ships of a specific size are required by the International Maritime Organization to carry this system. However, many boats still have AIS because of its safety benefits, even though they are not required to. The signal is broadcast via radio and can be picked up by terrestrial or spaceborne receivers. Many sensors pick up these signals and track the boats, but the information is often incomplete because of reception problems over large distances. This can lead to a mismatch between the date and time of image acquisition and the reported AIS signal that needs to be compensated. The position will often still not match perfectly, making manual labeling necessary [24]. Usually, these datasets come with bounding boxes [26] as labels, but higher quality labels like rotated bounding boxes or polygon labels are also found [24].

The SAR Ship Detection Dataset (SSDD) was one of the first publicly available ship detection datasets and was used in many studies [19], [28]. The official release comes with a bounding box, rotated bounding box, and polygon pixel-level annotations in over 1000 images with over 2000 ships. The images are in different polarizations and resolutions from different sensors, all at the GRD processing level. Therefore, no phase information is given. The SSDD dataset contains many ships with an average size of 35x35 pixels, which is still relatively large.

Because detecting small objects is a scorching topic Zhang et al. [24] published a second dataset, called LS-SSDD, addressing this problem. It lowers the average size to 20x20 pixels. Small in this sense means small in scale and not absolute measures; therefore, the objects might be "big" but appear small in Sentinel-1 images. 15 large scene Sentinel-1 images are used, which contain 6015 labeled ships. These images are delivered at the GRD processing level with co-polarization only. However, there is no detailed description of the ship parameters, only the number of pixels the ship occupies [24].

There is much more related work that cannot be covered in this study, but the following chapters will discuss some of them in greater detail.

2.4 Synthetic Aperture Radar

SAR uses microwave signals to build two-dimensional images. The used frequencies are between 1 and 10 GHz, interacting more with objects with a high relative permittivity (dielectric constant). Microwaves have the advantage of being mostly independent of atmospheric conditions and being active sensors; they are also independent of sunlight to illuminate the scene. The technology advanced from the airborne side looking radars and achieves a higher spatial resolution than stationary radars by sending and collecting the signal sequentially over the flight path. The echos are recorded at different times, giving the target information at slightly different positions. This strategy is eponymous as it creates a synthetic aperture along the flight path to record the echo and provides a high spatial resolution in the Azimuth direction. For a high resolution in Range direction, the signal time-of-flight is used to calculate the distance to the sensor. The time length of the signal pulse limits the smallest separable time difference that can be measured and, therefore, the range resolution. The imaged scene is assumed to be stationary relative to itself, which is not always the case since many targets, like boats, are moving [29]. These principles of SAR are illustrated in Figure 1.

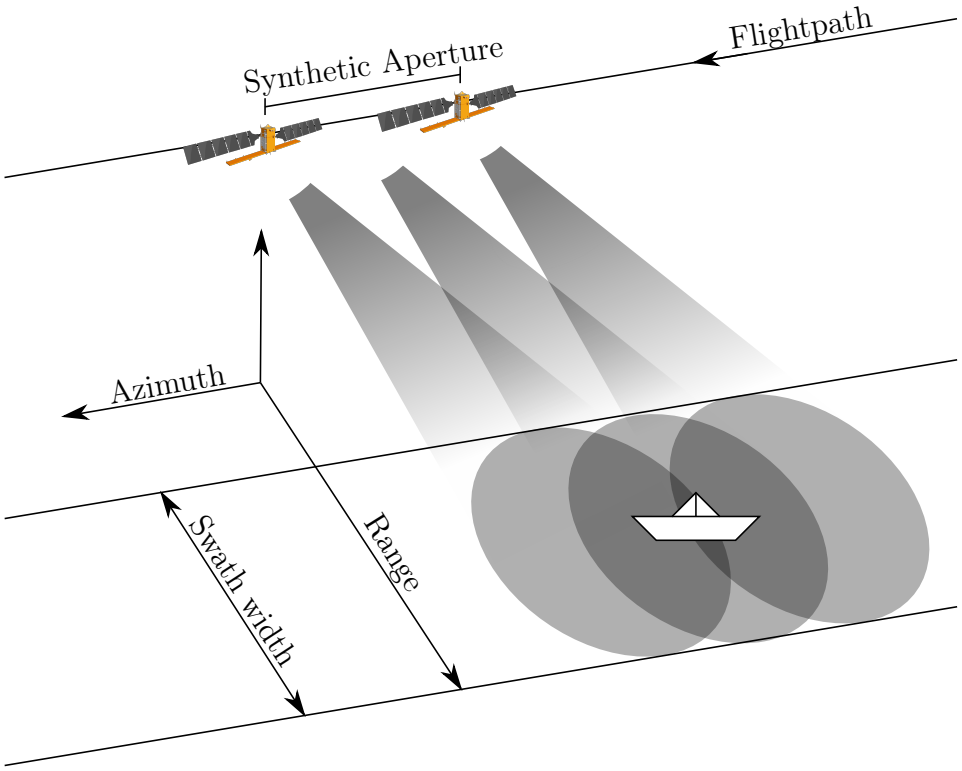


Figure 1: Principles of SAR

Currently, there are several satellite-based SAR instruments in orbit. Table 1 compares different instruments concerning frequency, swath width, resolution, launch year, and access. Two things stand out. First, most satellite data providers do not have an open-access policy. However, most of them, like TerrarSAR-X, might have research access or open-source archives. Second, the swath width and resolution are always a trade-off; either the spatial resolution is high, or the swath is wide.

A vital property of SAR, polarization, has yet to be mentioned. Besides phase and amplitude, the horizontal and vertical polarization of the outgoing and incoming signals can be recorded. This leads to four different polarizations: The co-polarizations, horizontal-transmit-horizontal-receive (HH) and vertical-transmit-vertical-receive (VV), and the cross-polarizations with horizontal-transmit-vertical-receive (HV) and vertical-transmit-horizontal-receive (VH). Not all instruments have four polarizations available in all imaging modes. Sentinel 1, for example, only provides VV and VH as dual-polarisation in the

Table 1: Comaparison of different satellite SAR platforms

Name	Frequency [GHz]	Swath width [km]	Resolution [m· m]	Year	Access
Sentinel 1	5.405	250	5·20	2014	Open
Gaofen-3	5.4	100	10·10	2016	Paid
TerrarSAR-X	9.65	30	3·3	2007	Paid
COSMO-SkyMed	9.6	40	3·3	2007	Paid
CSG	9.6	100	4·20	2019	Paid
Radarsat 2	5.405	150	8·8	2007	Paid

IW mode over the Mediterranean Sea. Polarization provides valuable information about the scattering effects of the target. Figure 2 shows different scattering effects, which may overlay each other in one resolution cell. Only the coherent sum signal of these scatterers can be measured, leading to the characteristic speckle in SAR images. The strongest scatterer in a resolution cell may dominate the measurement, so targets much smaller than the actual resolution cell can be visible in the image if they provide strong backscatter. Much like light reflectors appear bigger in the beam of a flashlight, an object that reflects the radar signal well appears bigger in the image. Radar corner reflectors, used for external sensor calibration of Sentinel-1, are only 2.8m on each leg but appear very bright if oriented correctly [30]. The double bounce and rough surface scattering lead to a strong signal in co-polarization, while the volumetric scattering gives a stronger signal in cross-polarization. Mirroring surfaces, like flat water, reflect the signal away from the sensor and give an overall low backscattering intensity [31].

2.4.1 Preprocessing of SAR images

Radar images are delivered in different processing levels. Standard options include raw, single look complex (SLC) and GRD with multilooking. Raw images are unprocessed data from the instruments and are not part of this study. SLC Images are the next processing level; the raw signal is focused and combined into a 2D image of complex values representing the amplitude and phase of the signal. GRD is a higher processing level in which the SLC

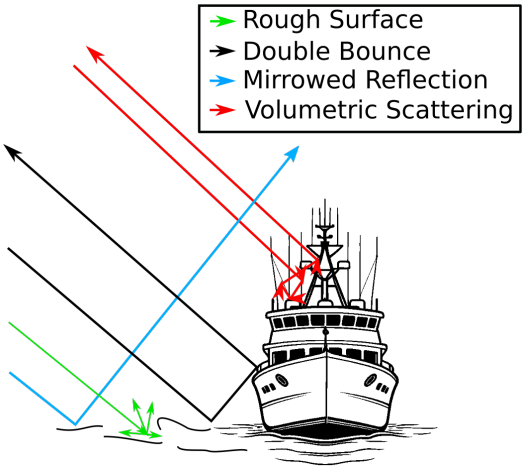


Figure 2: Different scattering effects that may overlay each other in one resolution cell.

data is projected from the slant range into the ground range direction and converted to dB values. The phase information is lost that way. Often, the GRD is combined with multilooking to reduce the speckle at the cost of spatial resolution. Originally, multilooking is achieved by splitting and merging the raw data, but mean filtering gives similar results and is often used instead [32].

For polarimetric methods, the SLC image is used to form polarimetric matrices. The complex scattering matrix [33] in Equation 1 is used as a basis. E denotes the electromagnetic wave, h horizontal, v vertical, s the scattered wave, i the incident wave, r the distance to the receiver, and k the wavenumber.

$$\begin{bmatrix} E_H^s \\ E_V^s \end{bmatrix} = \frac{e^{-jkr}}{r} \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \begin{bmatrix} E_H^i \\ E_V^i \end{bmatrix} \tag{1}$$

According to Equation 2, the covariance matrix can be formed from the scattering matrix [34]. Here, k denotes the target vector consisting of the elements of the scattering matrix, T denotes the transpose, and $\overline{overline}$ the complex conjugate.

$$
\begin{aligned}
k &= [S_{HH}, S_{HV}, S_{VH}, S_{VV}] \\
C &= \overline{kk}^T \\
&= \begin{bmatrix} |S_{HH}|^2 & \sqrt{2}S_{HH}\overline{S}_{HV} & S_{HH}S_{VV} \\ \sqrt{2}S_{HV}\overline{S}_{HH} & |S_{HV}|^2 & \sqrt{2}S_{HV}\overline{S}_{VV} \\ S_{VV}S_{HH} & \sqrt{2}S_{VV}\overline{S}_{HV} & |S_{VV}|^2 \end{bmatrix}
\end{aligned} \tag{2}
$$

For dual-polarisation with VV and VH this is reduced to Equation 3.

$$
\begin{aligned}
k &= [S_{VV}, S_{VH}] \\
C2 &= \overline{kk}^T \\
&= \begin{bmatrix} |S_{VV}|^2 & S_{VV}\overline{S}_{VH} \\ S_{VH}\overline{S}_{VV} & |S_{VH}|^2 \end{bmatrix}
\end{aligned} \tag{3}
$$

There are more methods for polarimetric analysis, like coherency matrices or different compositions [33], which are not part of this study.

3 Methodology

This study empirically examines the limits of small vessel detectability in Sentinel-1 SAR images to determine how suitable these images are for Search and Rescue. Therefore, a dataset of such vessels is needed. Then, different algorithms can be tuned, trained, and tested to detect the vessels in the images with a special focus on CNNs. The Hypothesis is that even with tiny targets of only a few pixels, the convolution-based methods outperform the adaptive-threshold-based methods because they can take the surrounding area into account. State-of-the-art models are trained to create a baseline for the dataset and determine favorable architectural characteristics. Also, new model variations are tested, both newly designed and reimplemented from the literature. Two novel designs are contributed, the use of max pooling for downsampling the residuals and the Reception block described in Section 3.5.1. The best-performing model is then selected and further analyzed. All experiments are carried out in Python with the PyTorch framework and the Python library Ray. The following chapters discuss the methods in greater detail, starting with the necessary dataset and ending with the applied ML methods.

3.1 Dataset

As stated in Section 2.3, datasets are a crucial part of all ML approaches. Especially supervised learning requires high-quality datasets with labeled data. For the specific task of detecting tiny boats, we need a dataset with as little preprocessing as possible so as not to blur out the target. Furthermore, we want to focus on Sentinel-1 IW mode since they are publicly available and can be delivered timely, which is crucial for the application in Search and Rescue. Since many researchers have pointed out the usefulness of polarimetric information, we want to have both channels available. The size needs to be sufficient for ML; since SSDD is a popular choice [19], we want at least 2456 vessels. Many images should be acquired at different times of the year to capture different sea states. To derive the limiting factors, we need to gather the environmental conditions, the image properties such as the incidence angle and the vessel parameters, such as type and length. The minimal requirements

can be summarized as follows:

- low-level preprocessing
  - SLC
  - no geometric correction
  - no multilooking
- minimum 2456 vessels
- Sentinel-1 IW mode only
- both polarizations
- vessel type and length
- incidence angle
- wave height
- wind speed
- wind direction

Even though ML became a research hotspot in SAR ship detection, there is still a lack of SAR ship detection datasets [24]. Table 2 compares five different datasets about the SAR platforms used, the number of scenes, and the number of ships. Compared to modern ML datasets like Imagenet, which consists of 14197122 images, all SAR datasets are relatively small [35]. The second thing that stands out is that, apart from the SSDD, all datasets are derived from relatively few scenes and, therefore, probably do not depict a diverse spectrum of environmental conditions. For the specific task of detecting small targets, the LS-SSDD dataset might be most suitable. However, it only comes with the GRD preprocessing level with georectification and multilooking. The accurate vessel description is also not available.

We have to conclude that, to the author's best knowledge, no suitable dataset exists to this point, and a new dataset needs to be designed for the

Table 2: Comaparison of SAR ship detection datasets

Name	Year	Platform	Number of images	Number of ships
SSDD [36]	2017	RadarSat-2, TerrarSAR-X, Sentinel-1	1160	2456
AIR-SARShip-1.0 [37]	2019	Gaofen-3	31	461
LS-SSDD-v1.0 [24]	2020	Sentinel-1	15	6015
SAR-Ship-Dataset [26]	2019	Gaofen-3, Sentinel-1	210	59535

specific needs of the task. The goal is to have a balanced classification dataset with images of vessels under 30 meters long and corresponding images with no vessels. That means, for simplicity, no bounding boxes or higher-quality annotations are used. Higher-quality annotations would be beneficial, but image-level annotation is sufficient for answering if a ship is detectable. Furthermore, different processing levels should be included for easy access and to measure their effect on classification. Nine different preprocessing levels are chosen: GRD with thermal noise removal (TNR), SLC and the covariance matrix $C2$, each as a multilooked image, a larger cutout that covers about the same area as the multilooked image and a single looked image. Preprocessing is done with the SNAP Toolbox; the different steps with the different parameters are described in Figure 3. Georeferencing is not used to avoid interpolation.

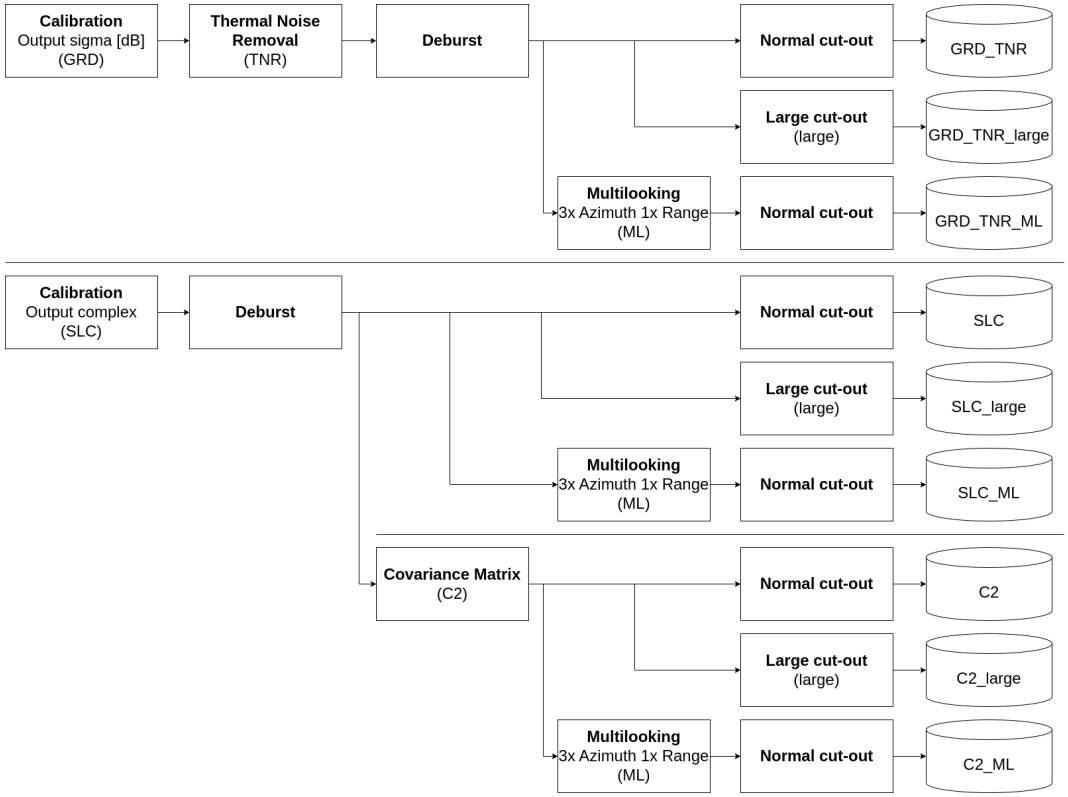


Figure 3: Preprocessing pipelines

We also use AIS, gathered from UNBigData [38], to obtain the ship locations. As explained in Section 2.3, only some ships are equipped with AIS, and the coverage varies. Especially in the area of interest (AOI), the central Mediterranean Sea between Libya and Lampedusa, the AIS coverage could be better [15]. To be able to collect enough data for this dataset, two more areas in the Mediterranean Sea that provide good AIS coverage [15] were chosen. The Ligurian Sea in the south of Genoa, Italy and the Gulf of Lion in the south of France. Figure 4 shows these areas. First, we gather all Sentinel 1 images between January 2021 and November 2023 that intersect the AOI. Then, all AIS information is requested for the footprint within ± 20 minutes of the acquisition time of the image. The vessel's position is linearly interpolated to the exact acquisition time of the image. The position may be wildly inaccurate because of the interpolation but also because of poor georeferencing of the image and the Doppler shift of moving targets [39]. That means

the snippets of the images with a vessel visible need a large enough cutout around the suspected position. Furthermore, the object should always be in a different position in the image, not always close to the center. A cutout of 256 by 256 pixels around the reported location is chosen and randomly shifted by up to 53 pixels in each direction. This shift ensures a 75-pixel, up to 1500 meters, buffer zone at the edge of the image to account for the inaccuracy of the position. This setting was manually verified by the author for each ship and proved to be a good choice. The 256-pixel square corresponds to roughly 7240 meters on the diagonal and, therefore, less than a radius of two nautical miles. If the weather conditions allow, a Search and Rescue crew could see the vessel from the center of the cutout, which is essential to the application in Search and Rescue. Such a cutout is made for all boats of less than 30 m in length. Then, all the AIS positions within that cutout are transferred to the pixel position in the image, where the origin lies in the top left corner.

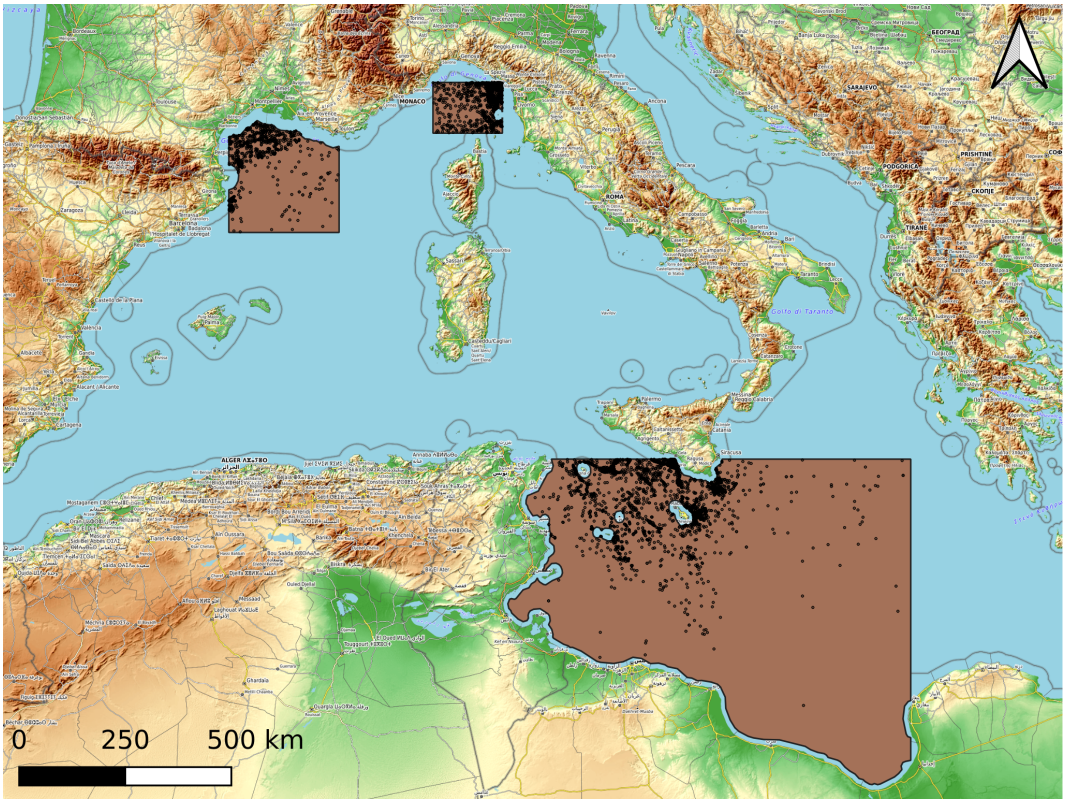


Figure 4: Acquisition areas and positions of ships with less than 30 m in length. Background map: OpenTopoMap®

A more complex approach is needed for the image that only shows water without any boat because there might be boats without AIS. First, a three times larger search area for a suitable cutout location is chosen. It must be relatively close to the vessel to show similar ocean state and environmental conditions. However, it should also be far enough not to be affected by secondary features of the vessel, which leads to the following approach: First, a search area is chosen, which lies ahead of the vessel's traveling direction. Then, to make it less likely that a vessel without AIS is in that area, the CFAR algorithm is used on GRD TNR multilooked images on both polarisations and combined with bitwise AND to find a spot where probably no boat is present. The AIS positions of the other boats are also added to this mask. If no cut out of 256 by 256 pixels can be found in that area, the next best search area is chosen, first 90 degrees to the left and right and then behind the vessel. If

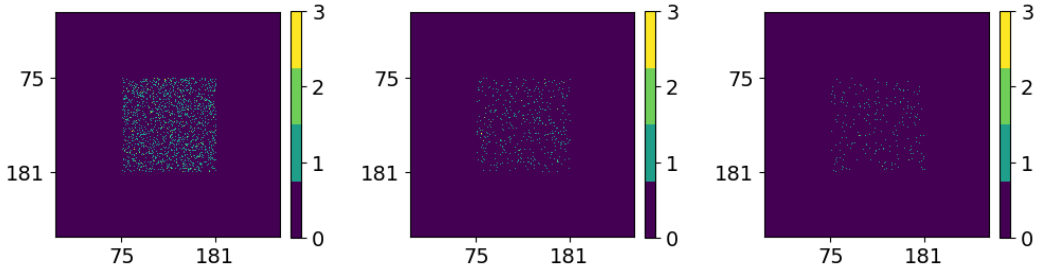
no cutout is found, the image is skipped.

The position of the cutouts is saved as metadata, and then the exact cutout is made for all nine different preprocessing levels. If no pair of vessel and water images are found or one preprocessing level is corrupted, the corresponding image pair is dropped. All remaining image pairs are manually checked in the GRD TNR processing level to see if they contain a vessel or not to verify this approach. The dataset is split into training, validation, and test datasets with the popular ratio of 70-20-10 for good model training. The dataset is binned according to its vessel length in 5-meter steps to achieve a similar vessel length distribution among all splits. Then, the split is made randomly for each bin and set. This is done once for every vessel and water image pair so that all sets contain the same geographic cutout of a vessel and water in all the different preprocessing levels.

The weather information is requested and averaged for each image and added to the metadata. The wave height is requested from the Copernicus Marine Service Mediterranean Sea Waves Reanalysis data [40]. The wind speed and direction are requested from the Global Ocean Hourly Sea Surface Wind dataset also provided by the Copernicus Marine Service [41]. If the wind data was not available from that source, the ERA5 hourly data from the Copernicus Climate Data Store is used instead [42].

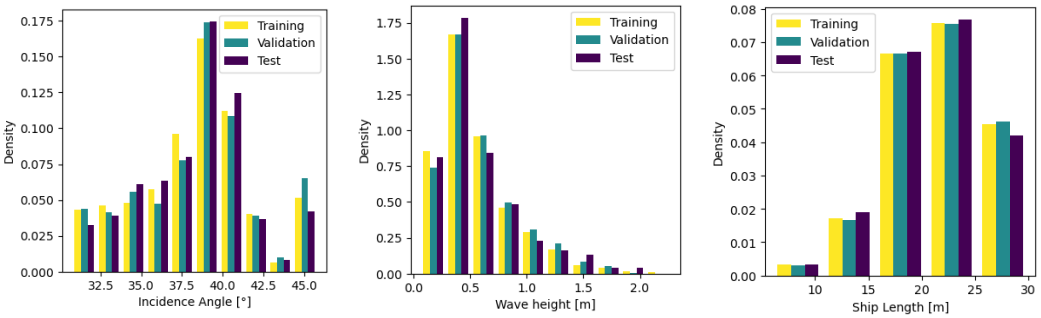
The approach led to a perfectly balanced, binary classification dataset consisting of 6192 cutouts from 1080 different scenes with rich metadata ready for ML. Unfortunately, the heading information is often incorrect; many AIS positions are reported with a heading of zero, as can be seen in Annex A. This is not seen as a probably for the water images, because we also use a sufficient distance to the ship's position and not just ahead of the ship's traveling direction. This lack of metadata is a problem for the analysis of the limitations but cannot be further addressed. Besides that, the dataset meets all the requirements stated before.

Figure 5 shows the distribution of the theoretical positions of the vessels in the cutouts with the 75-pixel buffer zone for each dataset split. It can be seen that they are well distributed within the allowed zone. To verify the diversity of incidence angle, wave height, and ship length are plotted for each dataset



(a) Training dataset (b) Validation dataset (c) Test dataset

Figure 5: Theoretical positions of the vessels in the cutout for each dataset split. The positions are accumulated over all respecting ship cutouts.



(a) Incidence Angle (b) Wave Height (c) Ship Length

Figure 6: The distribution of incidence angle, wave height, and ship length for each dataset split.

in Figure 6. The overall distribution of the different factors is not uniform but similar for each split. Annex A shows the parameter distribution in more detail in a scattering matrix. It needs to be mentioned that the dataset split is not visualized in the scattering matrix.

3.2 Metric

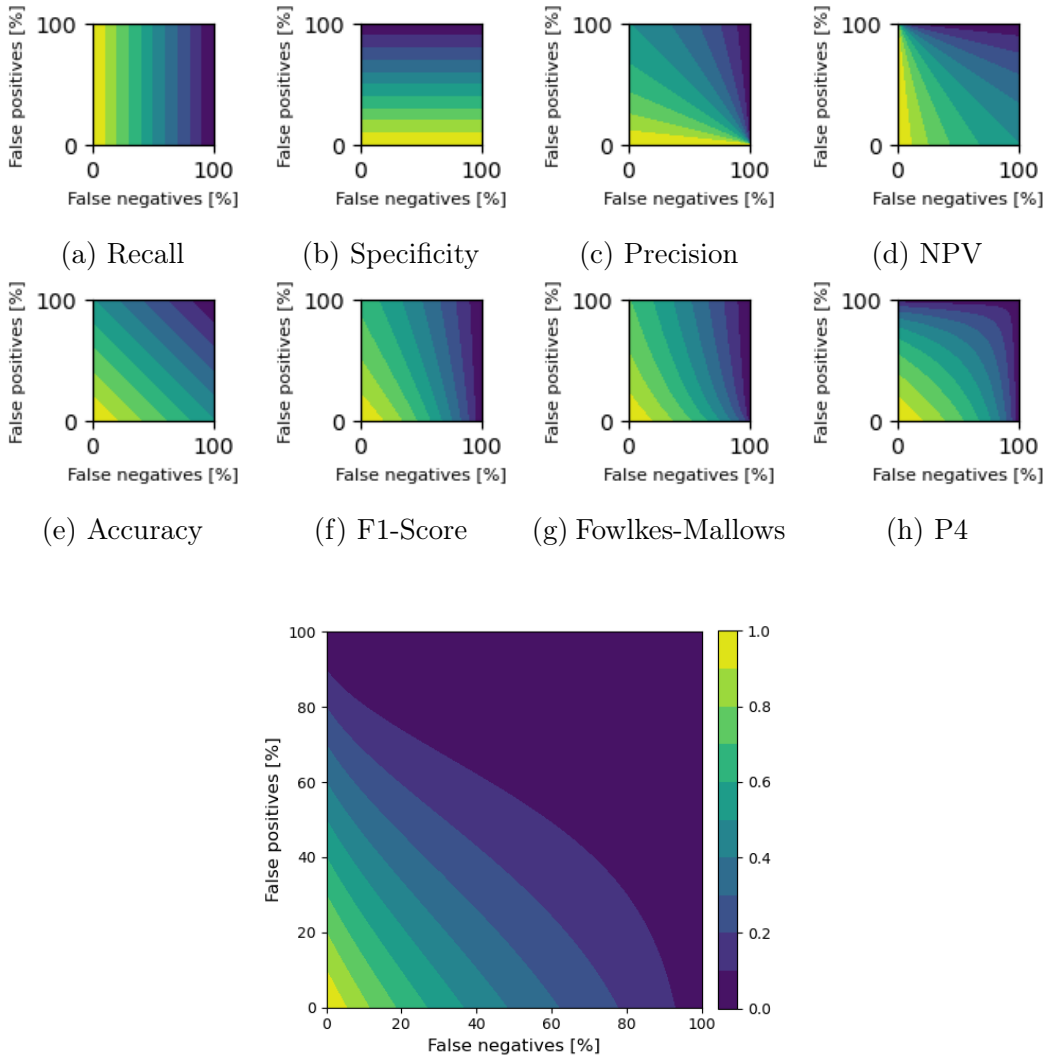
To evaluate the performance of the model different metrics exist for binary classification. The confusion matrix summarizes the quality of the prediction by showing the true positives, false positives, false negatives, and true negatives. Torres et al. [14], Paolo et al. [15], and Tings et al. [43] used the detectability in their papers, which is simply the ratio of true positives to false

Table 3: Binary confusion matrix and metrics

	Predicted True	Predicted False	
True	True Positive	false negative	Recall $\frac{TP}{TP+FN}$
False	False Positve	true negative	Specificity $\frac{TN}{TN+FP}$
	Precision $\frac{TP}{TP+FP}$	Negative Predicted Value $\frac{TN}{TN+FN}$	P4 $\frac{4 \cdot TP \cdot TN}{4 \cdot TP \cdot TN + (TP+TN) \cdot (FP+FN)}$
	Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$	F1-Score $\frac{2 \cdot TP}{2 \cdot TP+FP+FN}$	Fowlkes-Mallows index $\sqrt{recall \cdot precision}$

negatives. It, therefore, lacks the valuable information of false positives and true negatives. Table 3 shows the confusion matrix together with different ratios and metrics that are typically derived from the confusion matrix. These ratios and metrics are also displayed in Figure 7 for different false negative and false positive counts.

In Search and Rescue, it is more important not to miss a potential case than to have a false alarm. Therefore, the metric should punish false negative predictions harder than false positives. As can be seen from Figure 7, the F1-Score, the Fowlkes-Mallows index, the recall, and the NPV do that. Unfortunately, they do not punish trivial cases, always bedding on true or false or random bedding, as the P4 score does. We designed a new metric for this study to address this problem. Inspired by the Fowlkes-Mallows index, the square root of basic ratios is used as described in Equation 4. Figure 7i shows that it punishes trivial cases and mainly prefers a high number of false positives over false negatives. Additionally, it discriminates much finer if the score is above 50% accuracy and is close to zero for everything with less than 50% accuracy.



(a) Recall (b) Specificity (c) Precision (d) NPV

(e) Accuracy (f) F1-Score (g) Fowlkes-Mallows (h) P4

(i) FM3

Figure 7: Different metrics and ratios for binary classification over different false negative and false positive percentage. The FM3 score is a new metric designed to favor false positives over false negatives and punish trivial choices.

$$FM3 = \sqrt{recall \cdot specificity^2 \cdot NPV^3} \tag{4}$$

Introducing a new metric is problematic since it does not allow the direct comparison of the results of different models in the literature. However, this metric helps to compare different approaches tested in this study. To be able to

compare the results better and to have a complete picture of the performance for different thresholds can be drawn with the receiver operating characteristic (ROC) curve, which plots the true positive rate (TPR) over the false positive rate (FPR) as described in Equation 5 for the different thresholds.

$$\begin{aligned} Recall = TPR &= \frac{TP}{TP + FN} \\ FPR &= \frac{FP}{FP + TN} \end{aligned} \tag{5}$$

Figure 8 is an example of different ROC curves. The dotted line is a perfect classifier, while the dashed line is a random classifier. The solid line is the result of the CFAR algorithm on the dataset and lies between the other two lines. Additionally, the AUC is calculated with trapezoidal numeric integration. The closer the AUC value is to one, the better the model.

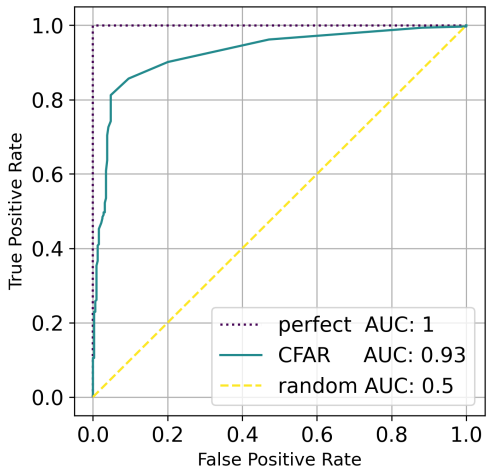


Figure 8: ROC curve examples.

3.3 Traditional Boat Detection

Over the past decades, many algorithms have been designed to detect ships. However, they are also suitable for small rubber inflatable boats in combination with traditional algorithms, as the research from Lanz et al. showed [5]–[7]. Probably the most common algorithm to this day is the constant false

alarm rate (CFAR) algorithm, which goes back to the year 1966 when it was first published [44] and is used in many research efforts to this day [5], [13], [15]. CFAR is based on the observation that ships provide higher intensity values than the water's backscatter. This backscatter is modeled with a suitable statistical distribution and fitted with samples from the image. Usually, these samples are taken from a guard window around the cell under test. When the distribution parameter is known, the probability of false alarm can be derived from the probability density function. Therefore, the algorithm can be tuned to have a constant false alarm rate [44] if the distribution is well-chosen and well-matched. However, the image might also have texture, which is not well represented in backscatter statistics. A simpler version of the CFAR algorithm only uses the expected value and variance of the pixel values and determines a threshold as a multitude of the variance. This approach assumes the backscatter to be normal distributed and allows for skipping more computationally costly distribution modeling. Since the normal distribution can not accurately model the backscatter, this method does not give a constant false alarm rate. For simplicity, that method is used and referred to as CFAR in this study according to Equation 6, where μ is the expected value, σ is the variance, Y is the pixel value, and T is a threshold value.

$$Y > \mu + T \cdot \sigma \tag{6}$$

This basic algorithm was often improved but requires tuning to specific images and scenes [15]. Filtering the image before detection can improve the signal-to-noise ratio. Since these traditional algorithms are fast, easy to implement, and still used in recent publications, they will also be tested on the dataset. Besides the CFAR algorithm, the polarimetric whitening filter (PWF), the polarimetric match filter (PMF), the polarimetric notch filter (PNF), normalized intensity sum (NIS), and the dual-pol ratio anomaly detector (DPolRAD) filtering techniques are investigated closer.

The PWF was introduced by Novak et al. in 1990 [45] for fully polarized SAR images. It tries to reduce the noise level by minimizing the ratio of the standard deviation over the mean pixel value. The optimal solution is found

to be the quadric with the inverse of the covariance matrix C as a weighting matrix and is described in Equation 7. Here, y denotes the optimally filtered image, C is the covariance matrix, and Y is the original pixel values of the image as a vector consisting of the different polarisations.

$$y_{PWF} = YC^{-1}Y \tag{7}$$

The PWF was further modified by An et al. [46] to compensate for the loss of power information in PWF filtering. Usually, the PWF is multiplied with the co- or cross-polarization intensity value to bring back the power information. An et al. [46] propose multiplying the weighting matric first with an optimal vector representing the power of the different channels. The optimal solution they found is minimizing the mean squared error between the original data X and the optimal data Y as described in Equation 8. X denotes the elements of the polarimetric scattering matrix.

$$y_{PWF_{MMSE}} = \frac{1}{m^2} Tr^2(C^{\frac{1}{2}}) X^{T*} C^{-1} X \tag{8}$$

Novak et al. also introduced the PMF in 1989 [47]. The goal is to maximize the target-to-clutter ratio used in the CFAR algorithm to detect targets. The optimal solution is to find the maximum eigenvalue of the covariance matrices of the clutter and target and use them as a filter. The filtered image is obtained by applying Equation 9. X denotes the intensity values of the image bands as a vector, W is the eigenvector corresponding to the maximum eigenvalue λ, and y is the optimally filtered image. The subscript t denotes target, while c denotes clutter. The clutter values are derived by spatial averaging around the target, similar to the guard window of the CFAR algorithm.

$$\begin{aligned} y_{PMF} &= \left| \overline{W}^T X \right|^2 \\ C_c^{-1} C_t W &= \lambda W \end{aligned} \tag{9}$$

The PNF was specially designed by Marino in 2013 with ship detection in mind [18]. The aim is to highlight the different polarimetric behaviors of a target area compared to the clutter around that area. A filtered image can be

calculated using Equation 10. y denotes the optimally filtered image, Tr is the Trace of a Matrix, ψ is a complete set of basis matrices under the Hermitian inner product, T_n is a threshold used for detection, and $RedR$ is a constant called Reduction Ratio and can be set individually. However, the formula for $RedR$ given in 10 has proven to be a good choice. The subscript c denotes clutter and is calculated by averaging over a large window area, while the subscript t denotes target. P_t^{min} is the minimum intensity of a target.

$$\begin{aligned} y_{PNF} &= \frac{1}{\sqrt{1+\frac{RedR}{t^{*T}t-\left|t^{*T}\hat{t}_c\right|^2}}} \\ t &= Tr(C\psi) \\ t &= \left[|S_{HH}|^2, |S_{HV}|^2, |S_{VV}|^2, S_{HH}^* S_{HV}, S_{HH}^* S_{VV}, S_{HV}^* S_{VV}\right] \\ \hat{t} &= \frac{t}{\|t\|} \\ RedR &= P_t^{min}\left(\frac{1}{T_n^2}-1\right) \\ P_t &= t^{*T}t - \left|t^{*T}\hat{t}_c\right|^2 \end{aligned} \tag{10}$$

In the dual polarimetric case, t reduces to Equation 11.

$$\begin{aligned} t &= Tr(C\psi) \\ t &= \left[|S_{VV}|^2, |S_{VH}|^2, S_{VV}S_{VH}^*\right] \end{aligned} \tag{11}$$

The NIS is based upon the PWF and states that the inter-channel correlation between co- and cross-polarization is low so that the PWF can be derived as the NIS [17]. The NIS can be calculated according to Equation 12. $R1$ and $R2$ are the intensity values of the respective channels.

$$\begin{aligned} y_{NIS} &= \frac{R1}{C11_c} + \frac{R2}{C22_c} \\ C11_c &= |S_{Co}|^2 \\ C22_c &= |S_{Cross}|^2 \end{aligned} \tag{12}$$

Marino et al. also developed the DPolRAD detector for sea ice detection, a

topic closely related to ship detection [48]. They call it a detector because it is meant to be used with a threshold to detect objects, but before thresholding, it is effectively a filter. As the name implies, it is based on the ratio of the polarisation channels according to Equation 13. The $\langle\rangle$ denotes a spatial average over a particular area. The subscripts c and t denote clutter and target, respectively.

$$y = \frac{\langle |S_{Cross}|^2 \rangle_t - \langle |S_{Cross}|^2 \rangle_c}{\langle |S_{Co}|^2 \rangle_c} \langle |S_{Cross}|^2 \rangle_t \tag{13}$$

This ratio was further improved upon by Lanz et al. [6] with the detection of particular small boats in mind. After thresholding the image, the developed detector combines 13 with 14 through bitwise OR operation after thresholding the image. This study will not use the bitwise operation but only the two filters separately as CrossDPolRad 13 and CoDPolRad 14.

$$y = \frac{\langle |S_{Co}|^2 \rangle_t - \langle |S_{Co}|^2 \rangle_c}{\langle |S_{Cross}|^2 \rangle_c} \langle |S_{Co}|^2 \rangle_t \tag{14}$$

3.4 Machine Learning and Convolutional Neural Networks

ML, specifically CNNs, showed excellent performance in image recognition [49]. At least since 2017, when the SSDD dataset was published, ML emerged in SAR ship detection [36], [50]. The main advantages of ML in SAR are better generalization, the option to do classification and regression in one step, and the performance of CNNs, which were reported to be much better compared to traditional methods [19]. The traditional methods need more pre- and post-processing to achieve the same results. As stated before, it is sufficient for this study to do binary classification and not do classic object detection as the term is often used in computer vision. An architecture sweep, in which many models are trained, and tested is used to find good-performing models. When introducing a new dataset it is helpful to establish a baseline with state-of-the-art models, which will be the starting point of the architecture sweep. For the use case in Search and Rescue, it is crucial to have fast computation of

large volumes of data to deliver the results timely. Furthermore, SAR comes with unique problems for CNNs, like strong speckle, but also offers new opportunities with complex-valued inputs and complex-valued CNNs. Therefore, this study will explore lightweight CNNs, tiny object detection, and CNNs for complex inputs. The following chapters describe the steps used in this study, from training to model architecture design.

3.4.1 Training

The training needs to be adjusted in a hyperparameter search for each model to achieve good performance. Statistical gradient descent combined with an exponential learning rate scheduler is used to update the weights and biases in the network. The statistical gradient descent can be controlled with the learning rate and momentum to adjust the stepsize after each iteration. The exponential learning rate scheduler reduces the learning rate after each epoch and is controlled through the gamma value. The gradients are computed with backpropagation through the cross entropy loss, a common choice for classification tasks [51]. The batch size can influence the training performance and must match the other parameters. The optimal choice of these hyperparameters can not be known in advance and must be tested for each trial. During the architecture sweep the ASHA scheduler, which uses parallelization and early stopping of unsuccessful trials, is chosen. The hardware setup with a single RTX 4090 does not allow massive parallelization of the training, but the ASHA scheduler chooses the training parameters and terminates trials early if they are not performing well concerning the FM3 score [52]. Each trial has a grace period of at least 10 iterations before it may be terminated. The parameters of the hyperparameter search are identical during the architecture sweep and are summarized in Table 4. The images are normalized with a mean of 0.5 and a standard deviation of 0.5. Different image augmentation techniques, like rotation, flipping, or cropping, can be applied to avoid overfitting and compensate for the small size of the dataset. These techniques may lead to unreasonable results for SAR images because of the characteristics of the side-looking sensor. For example, overlay and foreshortening will always be in the

same direction if georectification is not applied. The only standard methods that can be applied to keep these characteristics consistent are cropping and vertical flipping. Since the ship's exact location is unknown, random cropping may lead to images with false labels. That leaves random vertical flipping as the only option and is used with a probability of 50% in the training.

Table 4: Training parameter space.

Parameter	distribution	values
Learning Rate	loguniform	0.0001 - 0.5
Batch Size	choice	32, 64, 128, 256
Gamma	uniform	0.7 - 0.9
maximum epochs	choice	100
samples	choice	30
grace period	choice	10

3.4.2 Baseline

There is a wide variety of models to choose from. Table 5 compares a selection of different state-of-the-art models available from Torchvision regarding parameters and computational complexity in the form of giga FLOPs. Furthermore, their accuracy on the ImageNet dataset, the year of publication, and their size in MB are shown in Table 5. These specific models have less than 12 million parameters, an arbitrary limit chosen because of the limited hardware, dataset size, and training time. The selected models are known to be lightweight. The characteristics of SAR images described in Section 2.4 make the available pre-trained weights, which are based on Imagenet training, less useful. Each is trained from scratch on the dataset to create a baseline and determine which design choices may be favorable for this specific task. The hyperparameter search is repeated with different inputs from the different preprocessing levels of the dataset to see which preprocessing is favorable.

Table 5: Different Models available at PyTorch. The underlined values are the most favorable in each category.

Model	Accuracy	Parameters	GFLOPs	Size[MB]	year
ShuffleNetV2 0.5	60.552	1366792	0.04	5.3	2018
MNASNet 0.5	67.734	2218512	0.1	8.6	2019
small MobileNetV3	67.668	2542856	0.06	9.8	2019
MobileNetV2	71.878	3504872	0.3	13.6	2019
RegNetY_400MF	74.046	4344144	0.4	16.8	2020
EfficientNet B0	77.692	5288548	0.39	20.5	2020
large MobileNetV3	74.042	5483032	0.22	21.1	2019
MNASNet 1.3	76.506	6282256	0.53	24.2	2019
RegNetY_800MF	76.42	6432512	0.83	24.8	2020
GoogLeNet	69.778	6624904	1.5	49.7	2014
ShuffleNetV2 2.0	76.23	7393996	0.58	28.4	2018
EfficientNet B1	78.642	7794184	0.69	30.1	2020
Densenet-121	74.434	7978856	2.83	30.8	2018
EfficientNet B2	80.608	9109994	1.09	35.2	2020
ResNet-18	69.758	11689512	1.81	44.7	2015

3.4.3 Lightweight Models

Lightweight models are needed for real-time applications and platforms that do not have high computation power, like smartphones. So, the model should be as small as possible in memory space and as fast as possible by reducing memory access cost (MAC) and FLOPs. In recent years, new architectures have been designed that compromise as little as possible on performance while improving execution speed.

Rodriguez-Conde et al. [53] looked at different approaches to lightweight model design and reported that "the most critical point" in a detection model is the backbone architecture. The classification models tested in this study are similar to such backbones and can be seen as a step toward lightweight object detection. To achieve a lightweight model, modifications are made on a micro-level, like changing inner layers, or on a macro-level, like changing the depth and width of the model. To come up with a good macro-level design choice, many researchers rather define a design space and find an optimal

solution through different techniques for neural architecture search instead of handcrafting different designs. This approach requires a lot of computational power and time to complete and adds complexity to the methodology.

Radosavovic et al. [22] analyzed many design spaces and derived good performing choices from them, which helped reduce the size of the design space. They proposed a simple, straightforward architecture design by dividing the model into a stem, a body, and a head and derived a series of models called RegNet. The findings can be summarized as follows:

D1 The width increases by 2.5 per block.

D2 Do not reduce the resolution by more than a factor of 2 per block.

D3 Do not use more than 20 blocks in total.

D4 Do not group convolutions by more than 8 or 16.

Ma et al. [23] introduced guidelines for lightweight model architecture. Based on their findings, they derived a new architecture called ShuffleNetV2. The main points they derived are:

L1 Equal channel width reduces MAC

L2 Excessive group convolutional increases MAC

L3 Network fragmentation reduces the degree of parallelism

L4 Elemntwise operations are non-negligible

The authors also mention that the properties depend strongly on platform characteristics and that the actual speed should be considered for performance metrics instead of indirect metrics such as FLOPs.

3.4.4 Tiny Object Detection

Detecting Tiny Objects can be challenging for state-of-the-art CNNs since they are often designed around different problems, such as automated driving or quality control. Since remote sensing images often have relatively low spatial

resolution, tiny object detection has become an active research field in remote sensing and has led to specialized model architectures [21], [54], [55]. That often goes hand in hand with the need for fast, lightweight models so that large areas can be analyzed in a reasonable time.

Pang et al. [21] designed an object detection model for fast tiny object detection with a new lightweight backbone called TinyNet. It is based on residual blocks similar to [56] but has a lot fewer layers and parameters compared to ResNets. This allows to train TinyNet from scratch with limited dataset size and computational resources. The TinyNet architecture was reimplemented and tested for this study.

As explained in Section 2.2, Pawlowski et al. [20] experimented with different global pooling operations for tiny object detection. They found that global max pooling is more robust than global average pooling. This study will test global max pooling, global average pooling and fully connected layers before feeding them into the softmax activation.

3.4.5 3D CNNs

Usually, 2D convolutions are used in image recognition models to capture the geometric information. The channel information is processed by summing the same filter over all channels as described in Equation 15. The $\star$ denotes the 2D cross-correlation operator. The correlation between the input channels is weak, and the polarimetric information is not used properly [57], [58].

$$out(N_i, C_{out_j}) = bias(C_{out_j}) + \sum_{k=0}^{C_{in}-1} weight(C_{out_j}, k) \star input(N_i, k) \tag{15}$$

For polarimetric SAR images or optical hyperspectral images, the channel information can be just as important as the geometric information, which leads to using 3D convolutions instead of 2D convolutions to capture the channel information [57]–[61]. In that case, the third dimension is not geometric but the channels. Different variations and combinations of 2D and 3D convolutions have been tested by researchers over the years. Fully 3D convolutions showed

good performance [59], [60] but often struggle from structural redundancy, which makes alternating sequences of 2D convolutions and 3D convolutions beneficial [61]. Dong et al. used 2D convolution followed by 1x1xC 3D convolution, similar to Figure 9, which reduces the number of parameters and achieves better performance in their experiments [58].

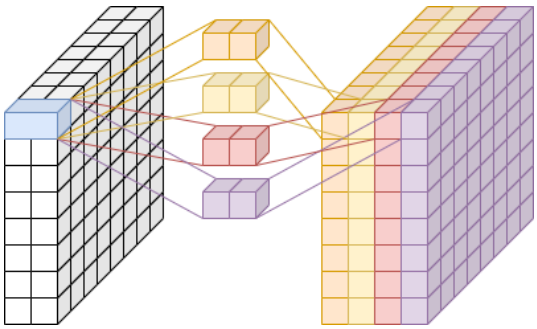


Figure 9: 3D 1x1xC convolution.

3.4.6 Complex-Valued CNNs

As described in 2.4.1, the polarimetric information can only be fully used when using complex-valued data. Tan et al. [57] designed a fully 3D CNN, which is wholly complex-valued. They tested it against complex-valued CNN and found it to have only 0.03% higher overall accuracy while taking three times longer to train and 1.5 times longer to test. The convolutions can handle complex-valued data well, but new methods need to be developed for activation, pooling, and batch normalization. For activation and pooling, Tan et al. [57] apply the function on the real part and the imaginary part separately. This is a common practice in complex-valued deep learning and is often used to feed complex values into a real-valued network [62]. Before softmax activation, the real and imaginary parts are split again and fed into fully connected layers. Splitting into real and imaginary parts performed noticeable better than splitting the complex value into amplitude and phase [57].

3.5 Model Architecture Design

Based on the literature findings above, two major points for improvements are identified: increasing the receptive field and carrying the weak signal of tiny objects along in the network. Therefore we introduced a new block design with dilated convolutions and made changes on the microlevel architecture design, which are explained in Section 3.5.1 and 3.5.2. We test these new designs and the design choices described above in a series of handcrafted models in another architecture sweep. The design terminology is adopted from [22], where the model is divided into stem, body, and head. The stem is the first stage of the model and prepares the input for the body. The body is the biggest part of the model with many blocks of operations. The head is last and outputs a vector resembling the result. The last layer always uses softmax activation to get pseudo probabilities of the prediction. The design strategy is split accordingly into three iterations. First, different blocks and depths of the body are tested with RegNet [22] and TinyNet [21] architecture to find a suitable configuration regarding the FM3 score. The best-performing architectures are then further tested with different heads. The last step is to test different stems and use the complex inputs for some of them. The following Sections go into greater detail about the different architectures.

3.5.1 Reception Block

It is beneficial to increase the receptive field and consider the surrounding area to classify tiny objects correctly [21]. This can be done in different ways, for example, with bigger kernels or dilated convolutions. The latter has the advantage of not adding too many parameters or FLOPs to the network. We designed a new block with three dilated convolutions with different dilation rates, which are combined to increase the receptive field within one operation. Figure 10 shows this novel design that we call Reception block. The first convolution has a dilation of zero, the second has a dilation of one, and the final convolution uses a dilation of two. The outputs of each convolution are then concatenated to get the final output. Accordingly, the output channels will always be a multiple of three.

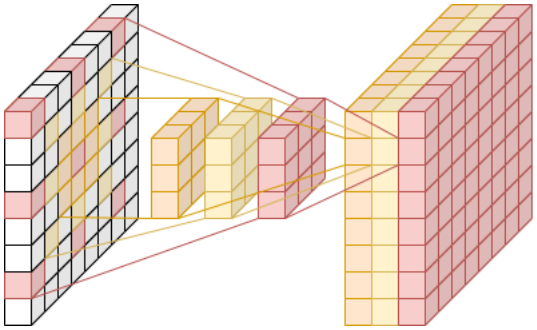


Figure 10: Dilated convolution in the Reception block.

3.5.2 Block Design

In the context of this study, a block is a combination of different operations like convolutions, activations, or normalization. Four block designs were used: the Residual block [56], the xBlock, the yBlock [22], and the TinyBlock similar to [21]. Figure 11 shows the details of each block. It can be seen that their designs are very similar, as they all use skip connections as ResNet uses them [56], and each convolution is followed by batch normalization and ReLU activation. The Residual block consists of one 1x1 convolution with increasing width. The second convolution uses a 3x3 kernel and equal numbers of channels. This combination of 1x1 and 3x3 convolution reduces the number of parameters and MAC according to L1. If the block is used for downsampling the second convolution is strided. The residual is formed by 1x1 convolution to adapt the width; if downsampling is applied, this convolution is also strided. The residual is added to the feature map before ReLU activation. The xBlock is similar to the Residual block, but the 3x3 convolution is followed by another 1x1 convolution. We changed the downsampling of the residual and used a max pooling followed by a 1x1 convolution instead of a strided 1x1 convolution, which is in conflict to L4 but expected to be an improvement. This novel design should conserve the weak signal of tiny objects, inspired by Pawlowski et al. who looked at global pooling operations [20]. The yBlock is similar to the xBlock but adds a squeeze excitation after the 3x3 convolution. Hu et al. [63] introduced squeeze excitation, which uses global average pooling followed by a fully connected layer with fewer channels and ReLU activation, as well

as a fully connected layer with the original number of channels and sigmoid activation. The resulting vector is then multiplied with the feature map. Last, the TinyBlock is also similar to the xBlock but inspired by [21], which uses two 3x3 convolutions to capture the surrounding information better. This is equivalent to one 5x5 convolution [64].

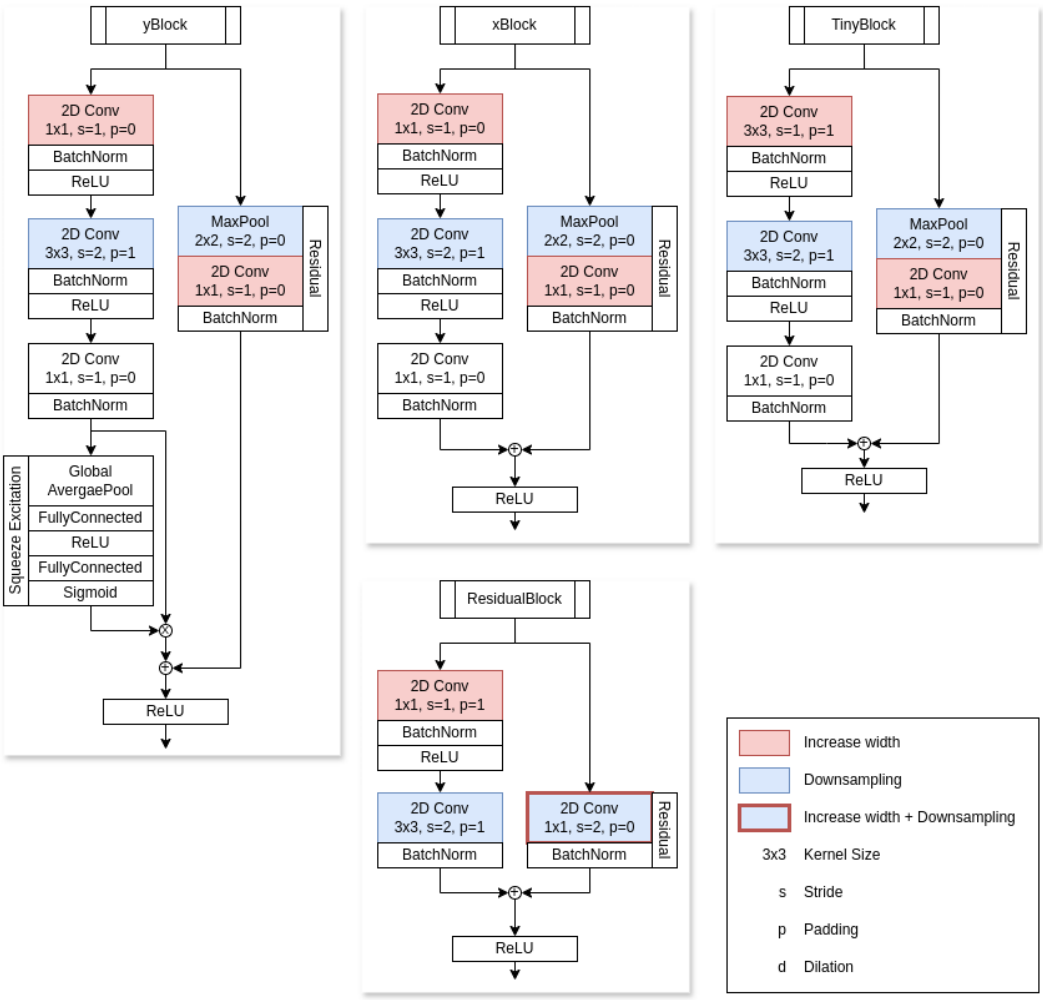


Figure 11: Design of the Residual block, the xBlock, the yBlock, and the TinyBlock.

3.5.3 First Iteration

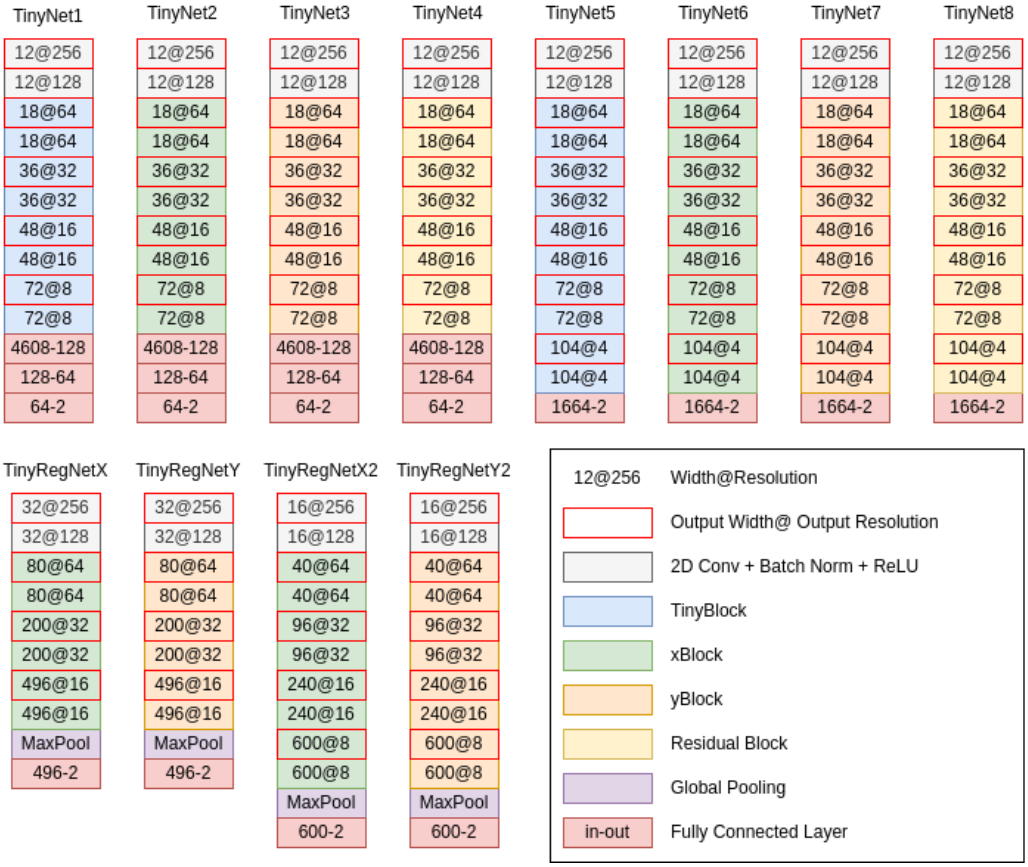


Figure 12: First design iteration with different numbers of downsamplings, width and blocks.

First, the TinyNet backbone from Pang et al. [21] is implemented with slight variations. The original architecture is based on residual blocks similar to ResNet, but the characteristic of TinyNet is the smaller width of only 72 layers maximum. Since the original TinyNet is used as a detection backbone, the classification head is missing and needs to be added to predict the output class. In this first iteration, that will be a multi-layer perceptron (MLP) with three layers. Following D1 to D4 from Radosavovic et al. a series of so-called TinyRegNets is designed. The main macro-level difference from TinyNet is the broader width of the network. Because of that bigger width, the depths can not be increased, similar to the TinyNet. Also, a 3-layer MLP cannot be

used as a head since it would require too many parameters. Instead, a max pooling head is used in which a global max pooling operation is followed by a fully connected layer. These two architectures are tested with different blocks in the body, depths, downsampling, and width. Figure 12 gives a detailed description of the tested architectures. The TinyNets used all different kinds of blocks with a width between 72 and 104 and six to seven downsamplings. The TinyRegNets, on the other hand, only used the xBlocks and yBlocks in the body with a downsampling of four to five stages and a maximum width of 600.

3.5.4 Second Iteration

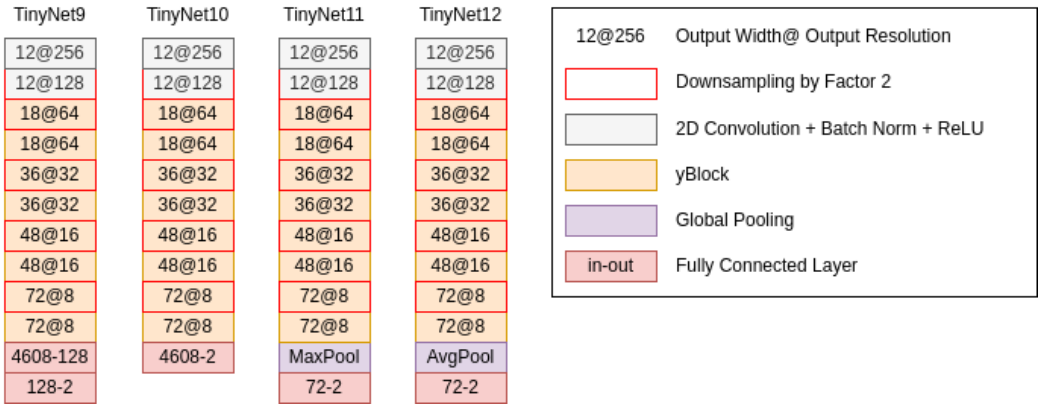


Figure 13: Second design iteration with different head designs.

In the second iteration, three types of heads are tested: a global average pooling head, a global max pooling head, and an MLP head. The pooling operations have the advantage of drastically reducing the number of parameters and FLOPs depending on how strongly the image resolution was downsampled. The pooling layer is followed by a single fully connected layer to predict the class. The MLP head might consist of up to two fully connected layers of different widths with ReLU activations in between. In the previous iteration, the MLP with three layers was already tested. The final layer is a softmax layer to get pseudo probabilities of the prediction. The details of the different architectures can be seen in Figure 13.

3.5.5 Third Iteration

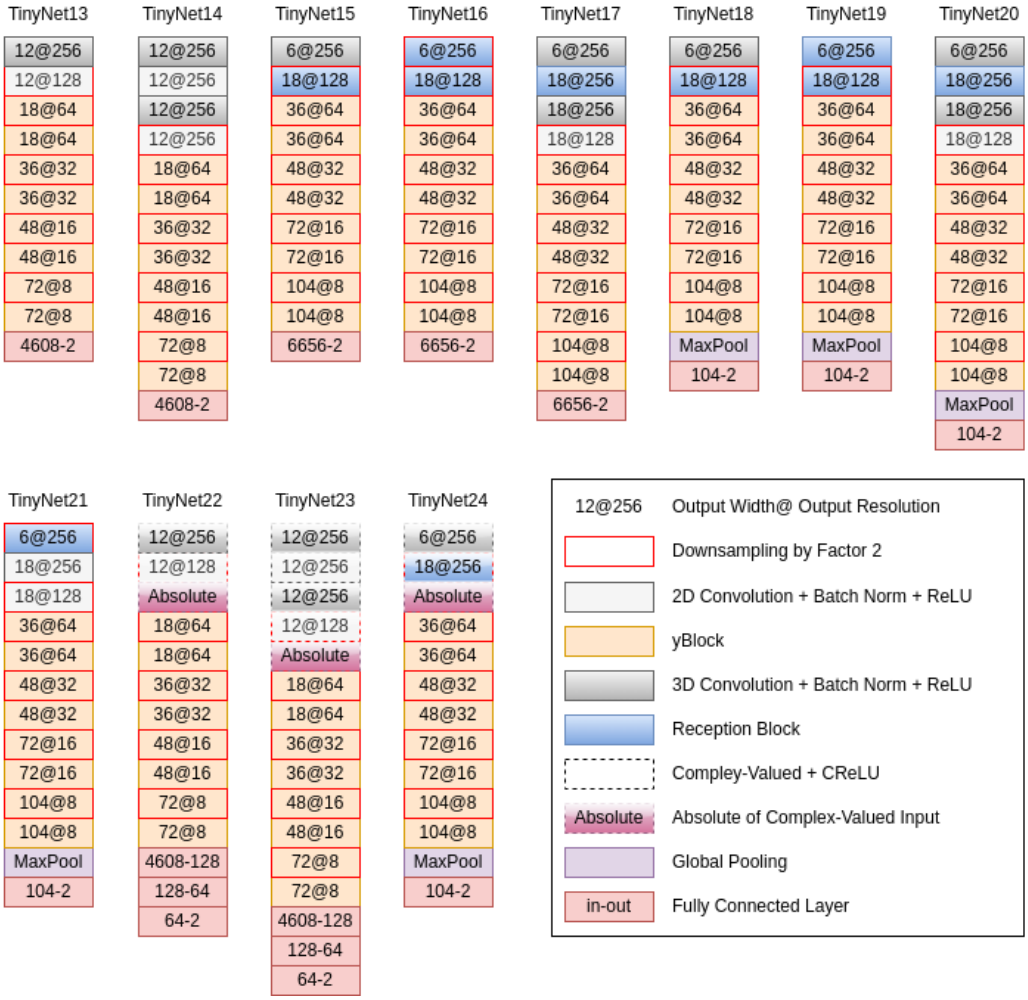


Figure 14: Third design iteration with different stem designs.

A basic stem with two 3x3 convolutions with batch normalization and ReLU similar to Pang et al. [21] was used in the previous iterations. In the last iteration, these are changed to various combinations of 3D convolutions, Reception blocks, and standard 2D convolutions, as described in Figure 14. Because of the large amount of parameters in the head, TinyNet15 to TinyNet17 are repeated with global max pooling heads. The last three trials are with complex-valued stems and complex-valued inputs. Here, no batch normalization was implemented, and instead of ReLU activation, CReLU activation is used. Before

entering the body, the absolute value of the complex-valued feature maps is computed and passed on.

3.6 Analysis

As the first step, the different tuning and training results of the traditional and ML models will be analyzed in terms of their parameters, training time, and FM3 score. Then, the best-performing model architecture and the best-performing traditional algorithm are compared to one another in four different ways. First, their performance on the test set is analyzed in depth to find out more about the limiting factors of the detection task. Secondly, the ML model is analyzed using the Grad-CAM ++ method to explain the predictions. For the traditional algorithms, we will look at the same image after filtering and locate the maximum value. Since the goal is to develop a model that can be used in Search and Rescue, the third analysis will investigate past cases, and lastly, a complete Sentinel-1 scene will be analyzed. The following Sections describe the methods in greater detail.

3.6.1 Training and Tuning

The training and tuning are analyzed with box plots for each design choice or tuned parameter. We have 30 training runs for each ML model and, accordingly, 30 results on the validation set. Additionally, each best-performing trial is evaluated once on the test set. During training, the model is evaluated against a threshold of 0.5 for simplicity and fast computation. On the test set, the prediction is tested against 100 thresholds between 0 and 1 to find the best threshold, which strongly influences the scores. All these results are then aggregated towards their common design choices, like width, downsamplings, block type, head, and stem. This will tell us how well the model performs on the test set and how easy it is to train the model. Additionally, the training time is evaluated with boxplots, which is a proxy for the model's complexity. For the traditional algorithms, the boxplots show how well the chosen parameters, inputs, and filterings perform on the test set.

3.6.2 Test Set

The results of the two models on the test set are compared concerning the different conditions in the image to analyze the driving factors. These conditions are ship length, SOG, heading, incidence angle, wind speed, wind direction, and wave height. As a first step, they are compared against one another in a partial correlation matrix. This matrix is extended by the partial correlation to having a correct prediction. These partial correlations only capture linear relationships, so to analyze the interesting factors further, they will be visualized as binned and plotted against their detectability or FM3 score. As explained in Section 3.2, the detectability alone is a weak measure and is only picked up because of its use in literature [14], [15], [43]. The FM3 score is used because it discriminates much finer. We keep a table of the test set results in the Annex to make this study comparable to other research.

3.6.3 Grad-CAM ++

Grad-CAM++ is used to analyze how the model makes the decisions and to verify that the actual boat leads to the decision. This method was designed in 2018 by Chattopadhyay et al. [65] to "provide better visual explanations of CNN model predictions." Grad-CAM++ was chosen over Grad-CAM because it uses pixel-wise weighted gradients instead of the average gradient. This proved beneficial for small target areas, which are the main focus of this study. The feature maps of the body's final layer will be used as input to derive the class activation map (CAM). Through the strong downsampling of the input, this CAM will only give a rough localization of what area of the image influenced the prediction. This will also help answer the hypothesis that using CNNs over adaptive-threshold-based methods is beneficial as they incorporate secondary features around the ship.

3.6.4 Search and Rescue Cases

Past Search and Rescue cases are analyzed to test if the algorithm is capable of detecting actual refugee boats. The rubber inflatable used by Lanz et al. could not be used because the lake is too small and crowded to be used with

the 256 by 256 cut-outs which TinyNet3 needs as an input. It can not be garunteed that the model will react to the rubber inflatable or other objects in the vicinity. Instead, the NGOs, Resqship, Sea Watch, and Pilotes Volontaires provided the author with an archive of past cases spotted by airplanes or Search and Rescue vessels. The data is considered sensitive, so no case details can be published. The case time and positions are matched with Sentinel-1 images, and three cases within ±20 min of the acquisition time are analyzed. The image is preprocessed like the dataset's images and split into 13 by 13 patches around the reported location. Each patch is 256 by 256 pixels in size and collected with a stride of 128 pixels, which leads to a 50% overlap with neighboring patches. This creates a buffer zone of roughly 26880 by 8960 meters around the reported location and should account for the inaccuracy of the reported position and the mismatch between acquisition time and the time the case was reported. If the case is in the area, it should be detected in the overlapping area of four images. This overlap has a diagonal of 1.88km and is, therefore, as good as reportings from airplanes that are precise to the one nautical mile.

3.6.5 Complete Sentinel-1 Image

As mentioned before, the timely delivery of the results is a crucial factor for the application in Search and Rescue. To benchmark the algorithms regarding execution time, they are tested on a complete Sentinel-1 scene. An image acquired on the 7th of January 2024 at 05:05:59 UTC over the central Mediterranean Sea without any landmass is chosen. Similar to Section 3.6.4, the image is preprocessed and split into 256 by 256 pixel patches with a stride of 128 pixels. In total, 55284 image patches were created for a single Sentinel-1 IW image.

4 Results and discussion

This Section presents the findings of our experiments starting with the baseline for the dataset and the newly designed architectures. Then the limits of detectability are explored and finally, the methods are tested for their use in Search and Rescue.

4.1 Problems

The training of many models struggled to converge. This is a common problem in machine learning, and the reason needs to be investigated more closely in future work. However, the problem might be connected to the small size of the dataset, the inherently low signal-to-noise ratio, and the choice of loss function. All training runs with an FM3 score of zero are filtered out so as not to clutter the following analysis.

4.2 Tuning and Training

Figure 15 compares the traditional algorithms with CNNs regarding their FM3 score on the test dataset. What stands out is that the MMSE PWF algorithm outperforms all others, followed by the ResNet-18 architecture, the RegNetY 800MF, and the Densenet121. The dataset is biased towards the adaptive-threshold-based algorithms, especially towards the CFAR algorithm, since they were used to determine the water cut-outs. The CFAR algorithm still produces false positives, which can be explained by the fact that in the creation of the dataset, a bigger area of the image was used to determine the clutter statistics, and in the test, there were only smaller cut-outs.

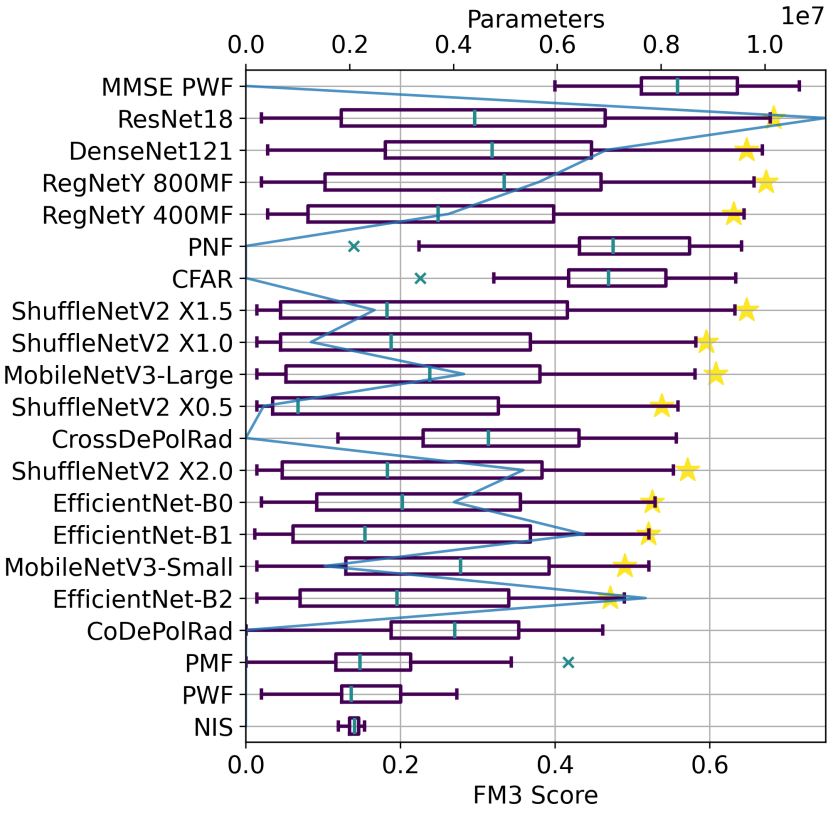
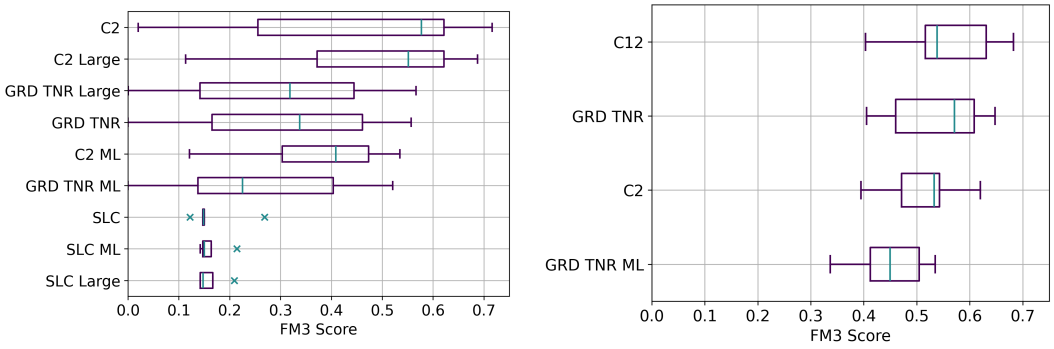


Figure 15: Boxplot of the FM3 score of the trials with traditional algorithms and the CNNs. The results of the traditional algorithm were achieved on the test set while the results of the CNNs were achieved on the validation set, except for the star which marks the highest score on the test set. The line indicates the number of parameters.

4.2.1 Tradtional Algorithm

The boxplot in Figure 15 shows that the traditional algorithms can compete well with the ML methods. This is unexpected but showcases the importance of these algorithms and their usage to this day. The PMF, PWF and NIS performed a lot worse compared to the other traditional algorithms. These filtering techniques lose the intensity information which needs to be fed back by multiplying the filtered image with one of the intensity channels. This was not done in this experiment. The MMSE PWF on the other hand does feed back the intensity in an optimal way with clear success. Figure 16a shows the

input types used for the traditional algorithms. As expected, the multilooked images perform much worse than their single-look counterparts. Furthermore, the ground range images performed much better than complex-valued slant range inputs, which might be caused by the methods used rather than by the projection to ground range or scaling to dB. This theory is backed up by the fact that the methods using the $C2$ matrix performed best, which is directly derived from the complex-valued slant range input.



(a) Traditional algorithms.

(b) Deeplearning algorithms.

Figure 16: Boxplot of the FM3 scores achieved for different input types by a) the traditional algorithms on the test set and b) by the ML algorithms on the validation set.

4.2.2 CNNs

When comparing the baseline algorithms in Figure 15, the ResNet, RegNetY, and Densenet architecture performed best. RegNet had a better median performance and might be easier to train for this task. A clear dependency on trainable parameters can not be drawn. The best-performing models tend to have more parameters but overall the specific design choices seem to play a more noticeable role. Inverted bottlenecks used by EfficientNet and MobileNet as well as channel shuffle used by ShuffleNetV2 seem to be less useful. Residuals, squeeze excitation and dense layers used by ResNet, RegNetY and Densenet respectively performed well. When concerning the time used for training, Figure 17 shows that the Resnet is found in the middle field, which is outstanding for the good performance it showed on the test dataset. Densenet,

on the other hand, took much longer to train. We can also see that the best model weights marked with the star are often close to the beginning of the training and not at the respecting maximum training time, showing that the models struggled to converge and had a tendency to overfit. ResNet, RegNetY 400MF and ShuffleNetV2 seemed to have better converging training, Densenet an EfficientNet and MobileNet less. For Training time the amount of parameters did matter with lighter models being faster to train. ResNet18 seems to be an exception to this rule. However, we need to keep in mind that training time and inference time are different from one another and should be evaluated separately. When looking at the inputs used in Figure 16, the C12 input was the best, which was expected since this input also showed great performance in traditional algorithms and has favorable theoretical properties as it correlates the cross- and co-polarization [66]. Using the complete C2 matrix, on the other hand, showed not as good results, which might be caused by the fact that the channels correlate with each other, which makes it harder for the network to learn meaningful features. However, as we can see in Annex C there are exceptions to this rule, for example with the EfficientNet-B2. From that Table, we can also see that for GRD TNR images using both channels usually worked best and for the single-channel inputs the VH channel was preferred. This can be explained by the fact that cross-polarization is less dependent on the incidence angle, which results in a uniform performance across the image swath [14], [24]. In Annex C we can see that all models were able to reliably detect water with only very view false positives. Detecting the boats on the other hand was much harder as the high number of false negatives tells us. The tiny features are not well captured by these state-of-the-art models. It also shows the importance of having a suitable metric, because the MMSE PWF and the ResNet18 do reach the same accuracy, but the MMSE PWF has fewer false negatives at the expense of more false positives. This difference is reflected in the new FM3 score.

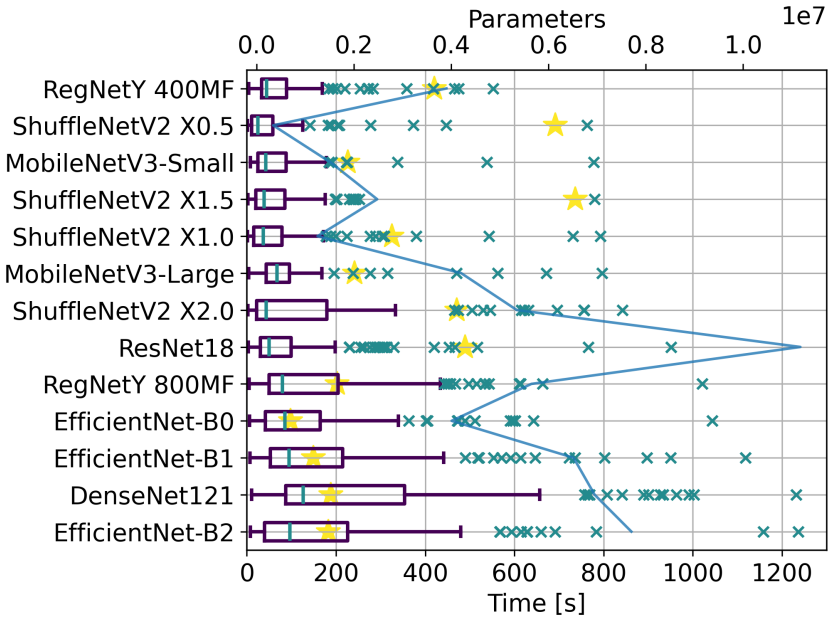
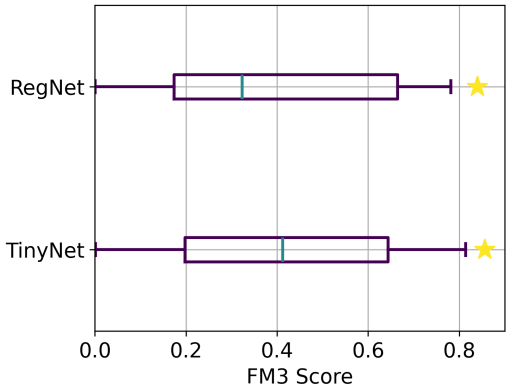


Figure 17: Time in seconds used for training until the best FM3 score is reached. The line indicates the number of trainable parameters in the model. The star marks the best FM3 score on the test set.

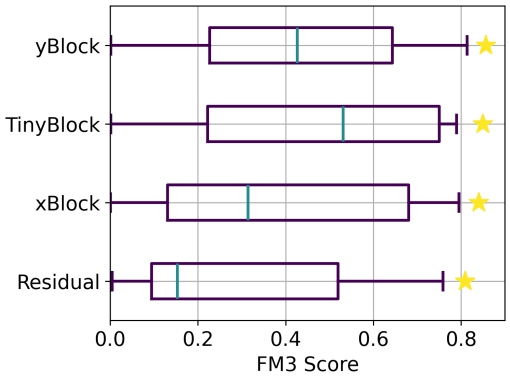
4.2.3 Model Architecture Design

Based on the findings of the baseline training, ResNet and RegNetY architectures were chosen for further study. As input only the GRD TNR images without multilooking were used since they showed consistently better performance in Figure 16. The FM3 scores of the training trials are drawn with boxplots, while each best result on the test set is marked with a star.

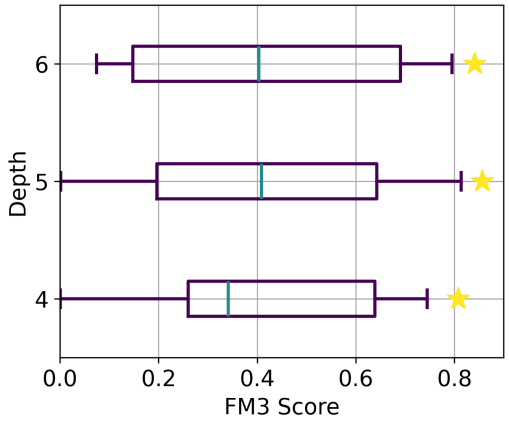
In the first iteration, the two different architectures were tested with different depths, widths, and blocks. Figure 18 shows the results for these four factors. From Figure 18a, it can be seen that both architectures perform well regarding the maximum FM3 score on the validation and test set, but the lighter TinyNet is slightly better. The Median FM3 score is noticeably higher for the TinyNet architecture. This behavior is also seen in Figure 18d for the width of the network. The two much wider networks reach lower scores, while the two slim versions reach higher maximum and median scores. When concerning the network depth in Figure 18c, four downsampling operations



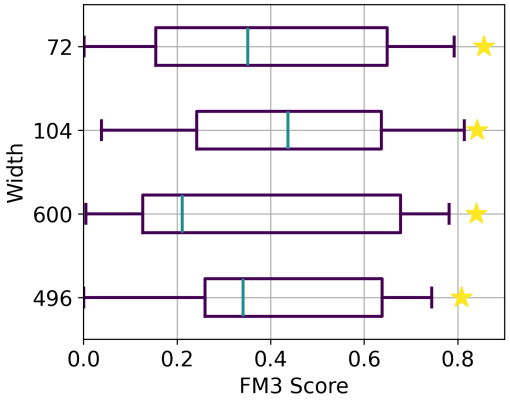
(a) Archicteture



(b) Block Design



(c) Model Depth



(d) Width of the Final Convolutional Layer

Figure 18: Results on the validation set of the first architecture sweep iteration. Each factor is plotted against the FM3 score. The star indicates the best FM3 score on the test set.

seem to be not enough while the difference between five and six downsamplings is not as pronounced. Five downsamplings performed best concerning the maximum score. When looking at the block design in Figure 18b, the two 3x3 Convolutions of TinyBlock seem beneficial, as well as the squeeze excitation of the yBlock. The xBlock and Residual Block showed noticeably lower Median performance. When comparing the xBlock to the Residual block, it seems to matter that instead of a 1x1 convolution with a stride of two, first a MaxPooling operation is performed and followed by a 1x1 convolution with stride 1 in the xBlock. Therefore we conclude that our new design choice is

successful. Overall, the best architecture of the first iteration was TinyNet3 with five downsamplings, a width of 72 layers, and yBlocks.

In the second iteration, the TinyNet3 design was tested with different heads. As can be seen from Figure 19, the MLP heads showed the best performance, while the average pooling head showed the worst performance. It seems to wash out the weak signal of the vessels. The MaxPooling head showed the best performance on the validation dataset and a better median performance for the training runs but had a slightly less high score on the test set. This can be explained by the fact that the MLP requires noticeably more parameters than the MaxPooling head and is, therefore, harder to fit. This can also be seen when comparing the median FM3 score of the single-layer perceptron MLP1 and the MaxPooling head to the other methods. They both have noticeably better median scores, as they both have the least amount of parameters. However, the best result on the test set was achieved with the MLP with three layers; the second best result was with the MLP with one layer, and therefore, these variations are prioritized.

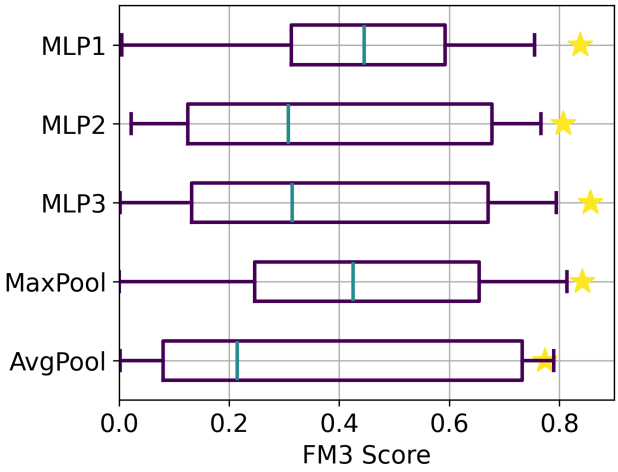


Figure 19: Results on the validation set of the second architecture sweep iteration. Each head variant is plotted against the FM3 score. The star indicates the best FM3 score on the test set.

The final step is to test different stems and complex-valued inputs. Surprisingly, the complex-valued inputs performed noticeably worse compared to

the real-valued stems, as can be seen in Figure 20. This might be influenced by the 3D convolutions that were used in all complex-valued stems, which did not perform as well as their 2D counterparts. However, that does not fully explain the noticeably lower performance. The lack of normalization probably also played a role but further research needs to be done to explore the complex-valued inputs. It is worth noticing that the best result on the test set was achieved with the 2D convolutions, and the best result on the validation set with stems that use the Reception block. The highest median can also be found in these stems, and the difference between the score on the validation set and the test set is smaller. This might point to a better separation of classes when using dilation.

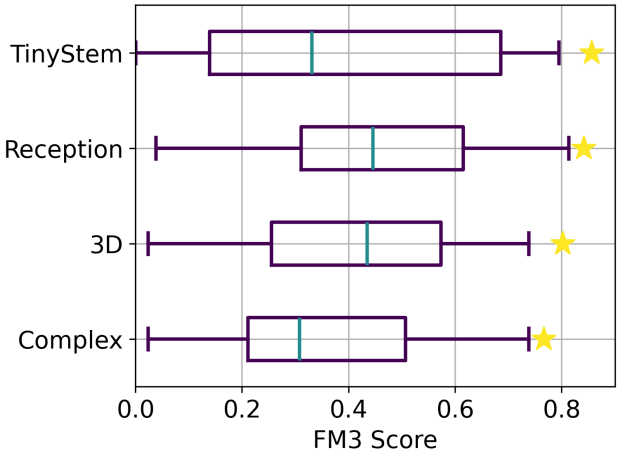
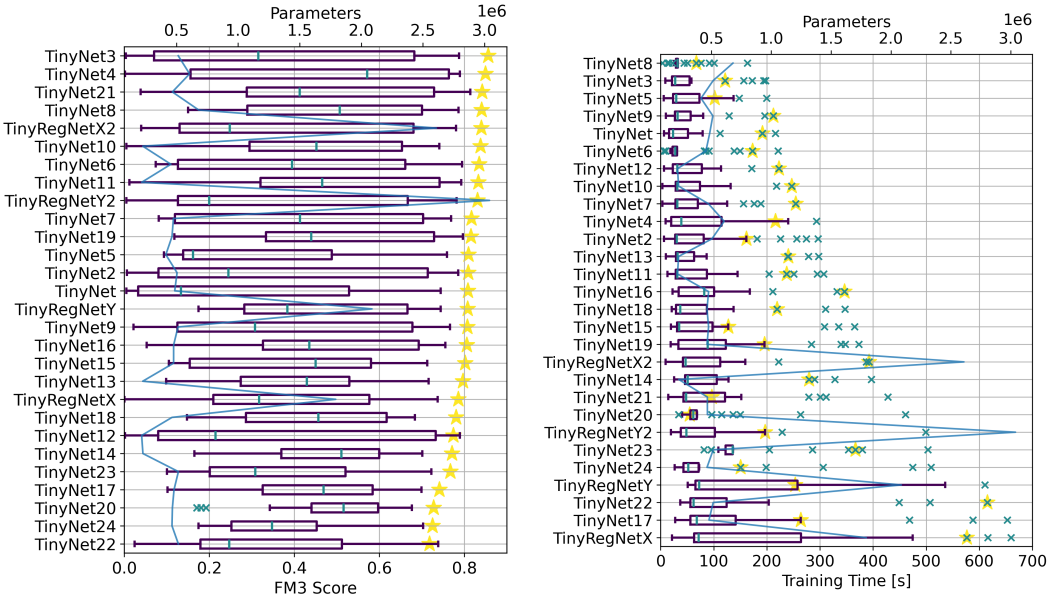


Figure 20: Results on the validation set of the training runs with different stems. Each factor is plotted against the FM3 score. The star indicates the best FM3 score on the test set.

To summarize the architecture sweep, we can look at Figure 21. A strong correlation to the number of trainable parameters can not be seen in the FM3 score nor in the training time, except for the TinyRegNet architectures, which have noticeably more parameters and usually take longer to train. The overall best-performing model on the test set was TinyNet3, with an FM3 score of 0.86, as can be seen in Figure 21a. This architecture uses 2D convolutions in the stem, yBlocks in the body with five downsamplings and a width of 72, finishing by an MLP with three layers. The score on the validation set was

noticeably lower, with a difference of 0.06 points. This can be explained by the fact that the score on the validation set was evaluated against a threshold of 0.5, while the score on the test set was achieved against a threshold of 0.16. The highest score on the validation set was achieved with the TinyNet21 and a score of 0.81. This architecture starts with a Reception block in combination with standard 2D convolutions followed by yBlocks and a MaxPooling Layer. Again, five downsamplings are used, and a width of 104 because the Reception block creates a wider network at the start. The gap between the best result on the validation set and the test set is much smaller, with a difference of 0.03 points, which again might show a better separation of classes with MaxPooling heads. When comparing the training time in Figure 21b, we can see that TinyNet3 took noticeably less time to train than TinyNet21.



(a) FM3 score

(b) Training time

Figure 21: Results of all different models regarding the FM3 score and the training time. The star marks the best trial that is then used on the test set. The line shows the number of trainable parameters in the model.

When we compare the performance of these newly designed CNNs to the baseline we can see that our specialized architecture performed noticeably better in all regards. The tables in Annex B and C show that especially

the amount of false negatives went down drastically, while the amount of false positives went up far less. Figure 21b and Figure 17 can see that the models in the architecture sweep took less time to train compared to the baseline models, therefore the models not only perform much better but are also very lightweight. The best-performing models lie also closer to the higher training times hinting at a better convergence during training.

4.3 Analysis on Test Dataset

The two best-performing algorithms are the MMSE PWF for the traditional algorithms and the TinyNet3 for the ML models. Their performance on the test set is closely analyzed in this Section to find out more about the limitations of the detection task. To start off we first compare the overall performance of the algorithms in Figure 22. We can see that the TinyNet3 has a much stronger and sharper discrimination compared to the MMSE PWF. This means that the highest FM3 score is achieved with fewer false positives. TinyNet3s AUC of 0.96 is higher than the reported AUC of Lanz et al. [67], while the MMSE PWFs AUC is similar. This is an uneven comparison since we tested real data of bigger boats while Lanz et al. simulated data by combining real scenes of smaller boats. The best FM3 scores were reached with a threshold of 0.16 for TinyNet3 and 4.59 for MMSE PWF. These thresholds are used from now on in the analysis.

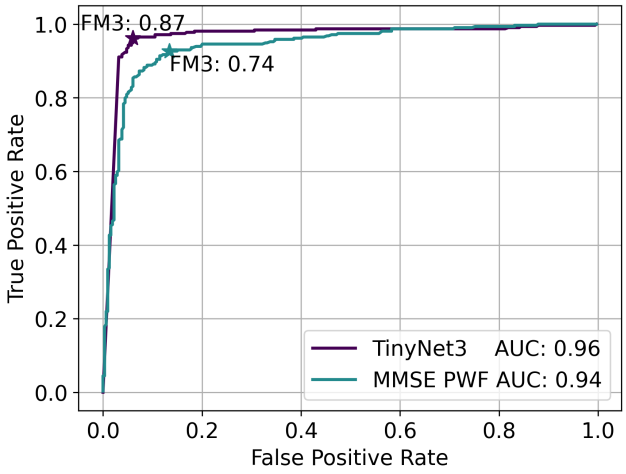


Figure 22: ROC curve for the MMSE PWF and TinyNet3 on the test set. The star marks the position of the highest FM3 score on the ROC curve

To get a feel for the data, we look at the partial correlation between the different factors and their correlations to the algorithms' correct predictions in Figure 23. It needs to be mentioned that these correlations only pick up linear relationships and that this can not be expected for the heading or wind direction. The first thing that stands out is that all factors are not strongly correlated because they have a partial correlation of less than 0.7. By far, the highest correlation is between the wave height, and the wind speed, which is expected. It is also worth noticing that there is a weak correlation between length, wave height and wind speed. This is also expected, as most larger vessels handle rough seas better.

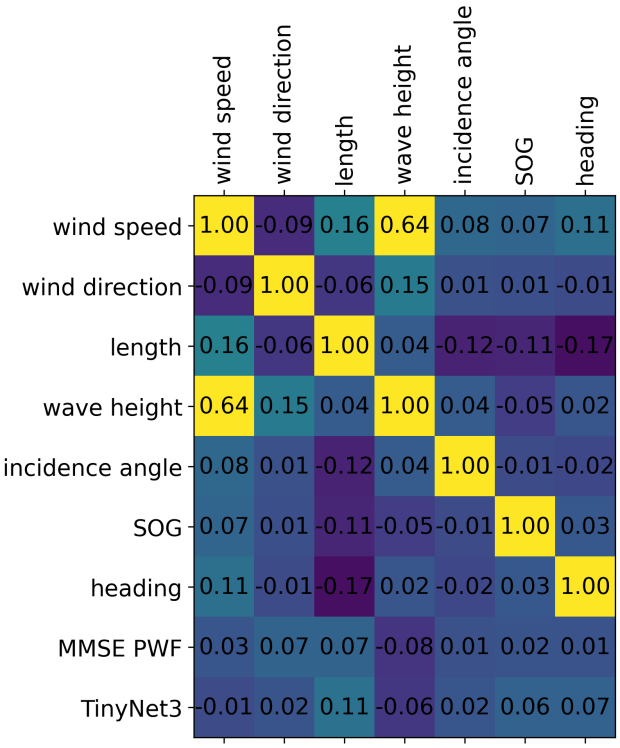
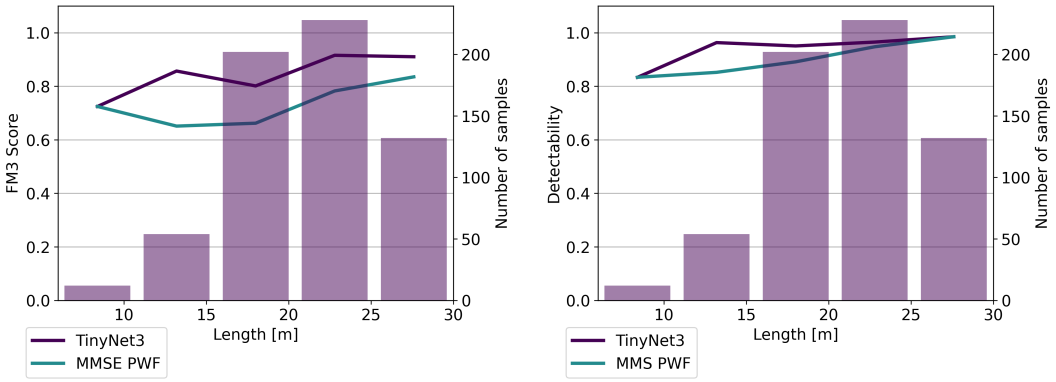


Figure 23: Partial correlation matrix between the different factors. The two additional lines at the bottom show the partial correlation between the factors and the respecting algorithm to predict the correct label.

Let's take a closer look at how the vessel length influences the performance. As expected, we do find a positive correlation for both algorithms, which is stronger for TinyNet3. When looking at Figure 24, we can verify that there is indeed a positive correlation, but for vessels of 10 to 15 meters, we see an increase in performance. There are far fewer samples available for these sizes, so the increase in performance might have been caused by another bias in the test dataset. When comparing the detectability over the ship length in Figure 24b, we see that it has mostly the same behavior as the FM3 score in Figure 24a, but the lower performance of the MMSE PWF for 15 and 20 meters bins is not picked up, which shows the importance of a suitable metric. Both algorithms show almost 10% higher detectability and noticeably better performance compared to Paolo et al., which is expected because these are more sophisticated algorithms compared to the CFAR algorithm [15]. However, this comparison

needs to be taken with a grain of salt because the detectability does not take the false alarm rate into account. Also, we did not use multilooked images, because they proved to have much lower performance at the beginning of the experiments. It has to be mentioned that Paolo et al. might have used a larger dataset and that the test dataset might be biased because of the use of the CFAR algorithm in its making.



(a) FM3 score (b) Detectability

Figure 24: FM3 score and detectability of MMSE PWF and TinyNet3 binned over the vessel length in meters. The second axis shows the number of samples for each length bin.

The speed of the vessel is suspected to have a negative correlation because the moving objects can not be properly focused in an SAR image, leading to smeared targets and a lower signal-to-noise ratio. When looking at the partial correlation in Figure 23 and the graph in Figure 25, we can see the opposite. While there is a decrease in FM3 score at first, the TinyNet3s performance increases for speeds above 4 knots, and for the MMSE PWF, the score increases again from 7 knots. Higher speeds lead to stronger wakes, which might be picked up by the TinyNet3. But again, it needs to be mentioned that the sample size for higher speeds is very low, which might cause biases.

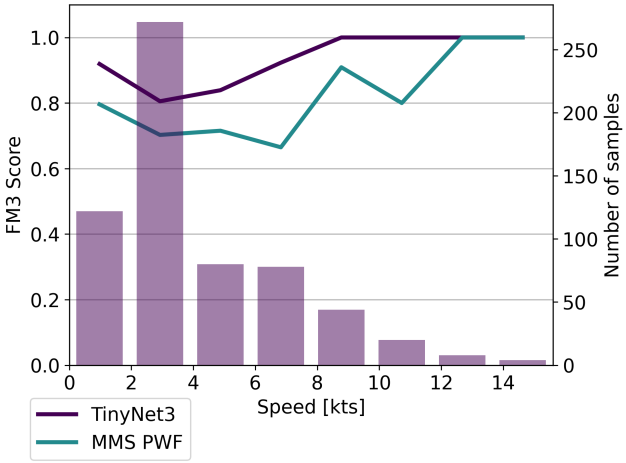


Figure 25: FM3 score of MMSE PWF and TinyNet3 binned over the vessel speed in knots. The second axis shows the number of samples for each speed bin.

Next, we take a closer look at the significant wave height, which is suspected to cause more backscatter and, therefore, make correct predictions harder. From Figure 23, we can see that there is indeed a negative correlation for both algorithms. However, it's weaker for TinyNet3. This indicates that the TinyNet3 is less distracted by the overall characteristics of the clutter. As can be seen in Figure 26, the sample size for higher waves is very small and, therefore, assumptions about these states are unreliable. However, there is indeed a negative correlation for wave heights below 1m, where the sample size is sufficient. The weak positive correlation between vessel length and sea state is not taken into account in Figure 26 and might explain the increase for bigger wave heights. Furthermore, Lanz. et al. reported a noticeable drop in performance only for a wave height of 2.5 meters [7], which is not covered in this dataset.

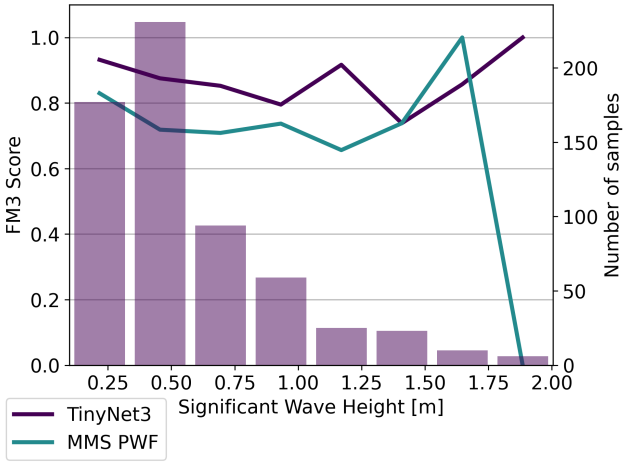
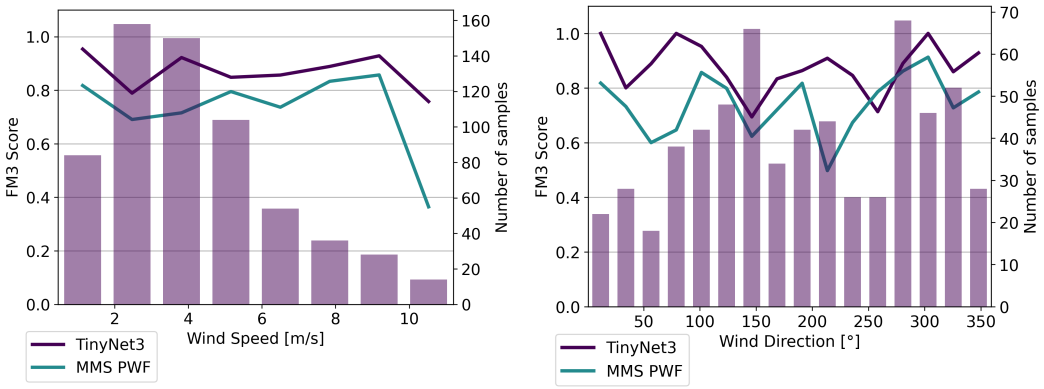


Figure 26: FM3 score of MMSE PWF and TinyNet3 binned over the significant wave height in meters. The second axis shows the number of samples for each wave height bin.

Many researchers suspect wind speed and direction to be significant factors [7], [16], [43], [68]. Surprisingly, the correlation of the wind speed is 0.03 for the MMSE PWF and only 0.01 for the TinyNet3. In Figure 27a, we can see the FM3 score over the wind speeds. Except for very low and very high wind speeds we see virtually no influence on the FM3 score; the reason for this is unknown but may be caused by a bias in the dataset. Figure 27b shows the FM3 score over the wind direction, which is much more volatile. However, the cause for the minimum FM3 scores could not yet be determined. The wind direction is not relative to the sensor's line of sight which might give further insight as suggested by [16].



(a) (b)

Figure 27: FM3 score of MMSE PWF and TinyNet3 for a) wind speed and b) wind direction. The second axis shows the number of samples for each bin.

In the literature, the incidence angle is also described as a strong driving factor [7], [14], [16], [43], [68], but in this dataset and with these algorithms, it shows only an insignificant correlation. Figure 28 shows that the distribution of incidence angles is uneven for this dataset, which might lead to biases. However, it can also be seen that for the low incidence angles, TinyNet3 shows noticeably less volatile behavior than the MMSE PWF. For high incidence angles $> 42°$, we see a decrease in performance, which might be caused by other parameters of the subset. The sample size for these bins is small and, therefore, assumptions about these states are unreliable. This might be caused by the higher backscatter that confuses the MMSE PWF, while the CNN can handle such effects better.

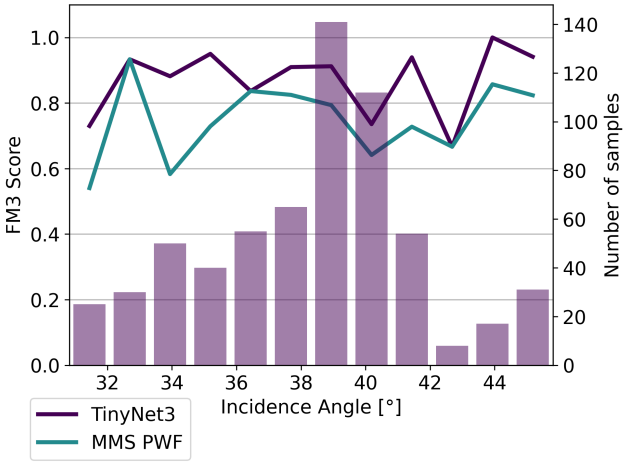


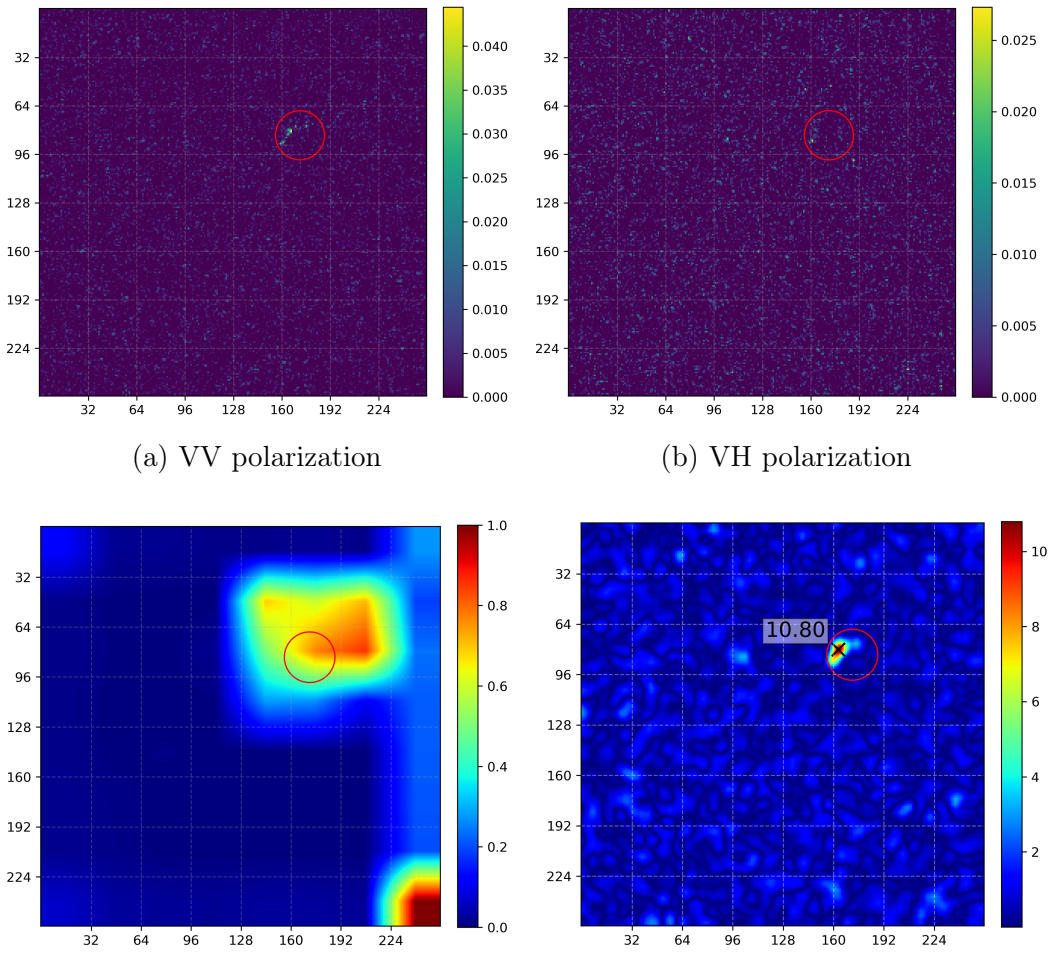
Figure 28: FM3 score of MMSE PWF and TinyNet3 binned over the incidence angle in degrees. The second axis shows the number of samples for each angle bin.

Overall, the TinyNet3 showed a more stable performance and seemed to deal a lot better with effects that are more spaced out, like smearing because of target movement or higher backscatter because of the weather or incidence angle. However, the sample size is often not big enough to be certain of the results. Unfortunately, the dataset size can only be addressed in future work.

4.4 Grad-CAM ++ and Filtered Image

Since we can not look at all images, only two scenes are shown here where TinyNet3 correctly predicted the vessel presence and the position estimation by AIS is relatively good. However, these results were checked against 43 random samples, and they showed more or less similar behavior. The first selected scene in Figure 29 shows a clearly visible ship in the top right corner; the red circle marks the ship's AIS position. In Figure 29c, we can see that TinyNet3 uses the ship's location and direct surroundings to predict the ship with a pseudo probability of 0.99. We can also see that there is a large artifact in the bottom right corner and some more artifacts on the other edges. This is seen often among the 43 samples in different intensities and locations along the edges. Also, for an untrained model, the edges show similar artifacts.

It may be caused by the zero padding applied by many convolutional layers. When we look at the MMSE PWF filtered image after the CFAR algorithm was applied without thresholding, we can see the ship's position. The highest value is 10.8, which is high above the tuned threshold for MMSE PWF of 4.59. The position of the ship is also picked up precisely, which can not be expected for the TinyNet3 because of the strong downsampling, as mentioned in Section 3.6.3.



(a) VV polarization

(b) VH polarization

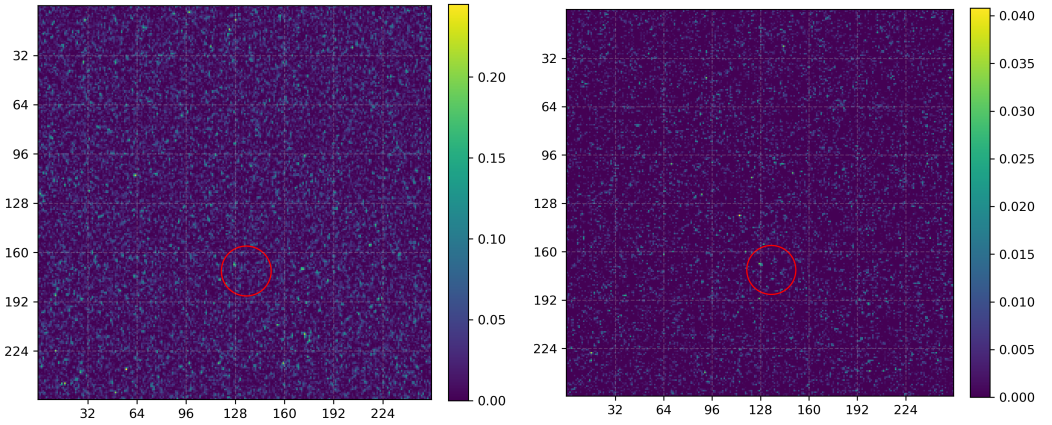
(c) Grad-CAM ++ for predicting ship

(d) MMSE PWF

Figure 29: Visual analysis of ship number 2714 called "Mare Chiaro", a 22 meter fishing vessel. a) and b) show the VV and VH polarization of the GRD TNR processed image. c) shows the Grad-CAM ++ result for the prediction ship and d) the image after the MMSE PWF and CFAR filtering. The red circle marks the reported AIS position while the black x marks the highest value in the filtered image.

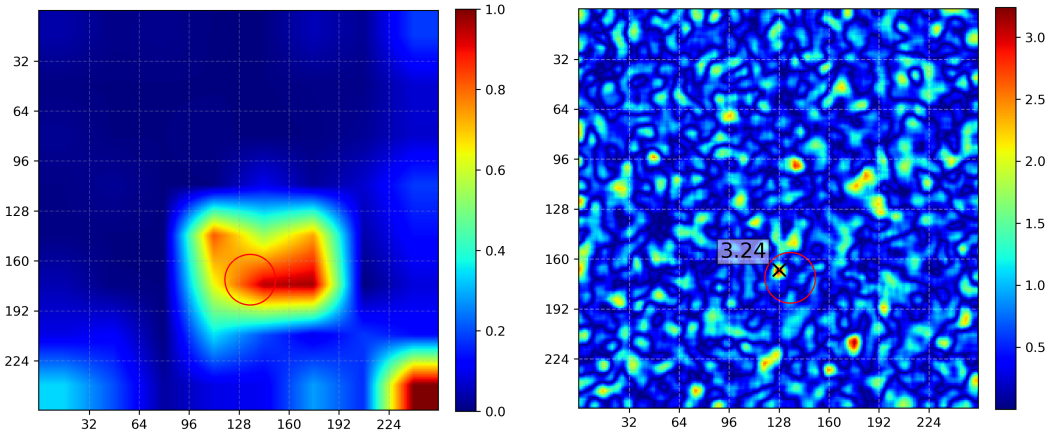
The second scene in Figure 30 shows a very weak signal of the ship in the image, and even experienced SAR experts might struggle to locate the vessel in the GRD TNR image. It's located in the center of the image, a little bit to the bottom. The signal is a bit more pronounced in the cross-polarization channel which shows the importance of using both channels. The TinyNet3 locates the

ship correctly with a pseudo probability of 0.99, again taking the surroundings into account. The adaptive-threshold-based MMSE PWF also locates the ship correctly, but with a value of 3.24, it would be below the tuned threshold and not detect the ship's presence. We can also see in this image how the surroundings produce similar patterns, which leads to confusion.



(a) VV polarization

(b) VH polarization

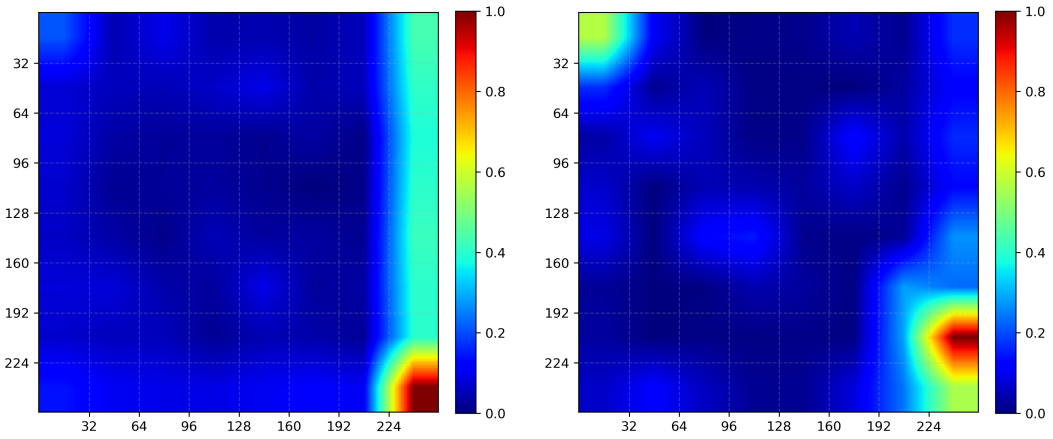


(c) Grad-CAM ++ for predicting ship

(d) CFAR

Figure 30: Visual analysis of ship number 960, called "Palermo Nostra", a 20 meter fishing vessel. a) and b) show the VV and VH polarization of the GRD TNR processed image. c) shows the Grad-CAM ++ result for the prediction ship and d) the image after the MMSE PWF and CFAR filtering. The red circle marks the reported AIS position while the black x marks the highest value in the filtered image.

We can conclude that the CNN does indeed capture the ship's location and takes the surrounding area into account, proving the hypothesis to be true. However, we can also see some artifacts on the edges that should be addressed in future work. While these artifacts are quite strong in these two examples, they are not always as strong and are not unique to images with ships. Figure 31 shows the respecting water images of ship numbers 961 and 2714, and the same artifacts are visible on the edges, especially in the bottom right corner. However, this did not affect the prediction, as both images are correctly predicted to be water with a pseudo probability of 1. Therefore, it's safe to assume that the artifacts do not show a bias of the TinyNet3, and they might explain why the global pooling operations lead to weaker performance.



(a) Water number 2714 (b) Water number 960

Figure 31: Visual analysis of the Grad-CAM ++ result for the water images corresponding to ship number a) 2714 and b) 960.

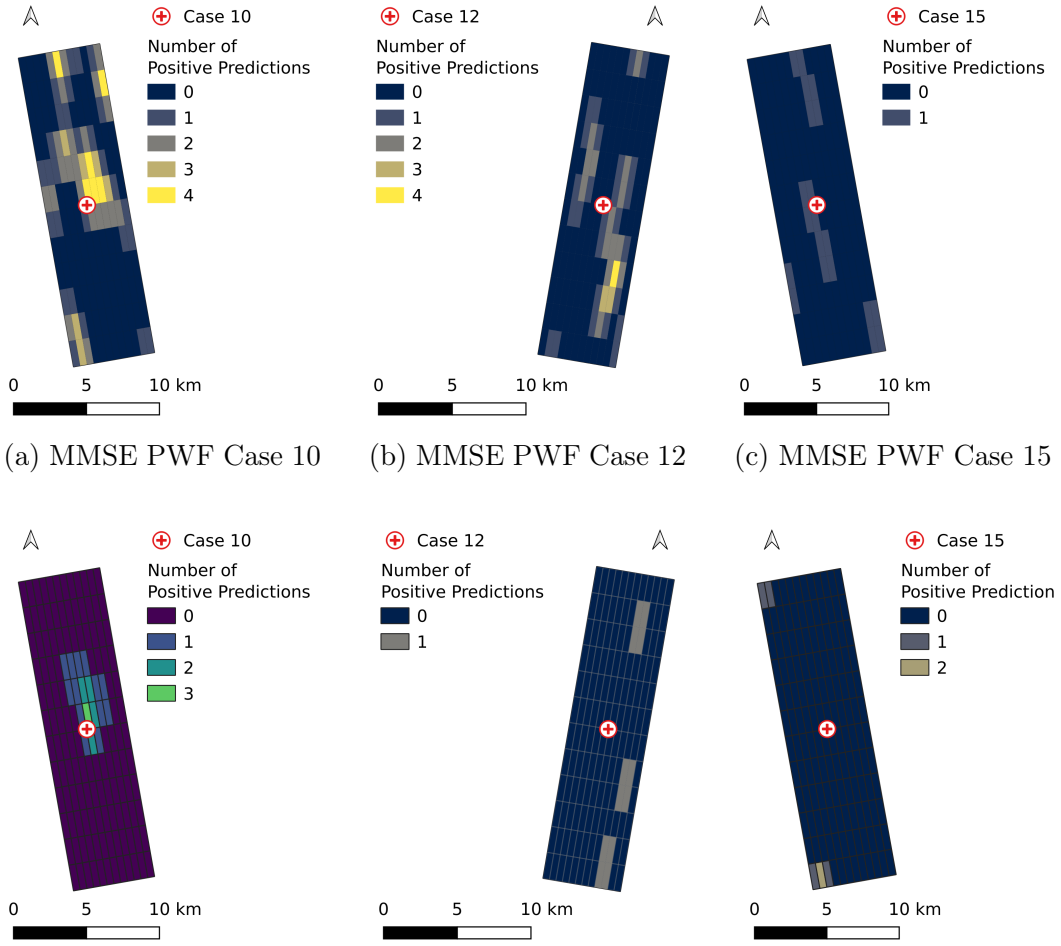
4.5 Analysis of Past Cases

As the performance regarding the training and the test set is evaluated in depth, it is important to test these models in real-world scenarios. Figure 32 compares the algorithms on three different cases by applying the threshold of 0.16 for TinyNet3 and 4.59 for MMSE PWF and then counting how many images were predicted as being positive within each overlapping area. When looking at case number 10 in Figures 32a and 32d, we can see that the MMSE

Table 6: Conditions for the three past cases.

case	mean wave height [m]	mean wind speed [m/s]	mean wind direction [°]	incidence angle [°]
10	0.63	5.39	158.24	39.24-39.58
12	0.53	4.26	73.48	41.87-42.17
15	0.27	4.21	121.16	37.50-37.86

PWF produces way too many false positives to be conclusive. When we change the threshold, the picture becomes clearer and similar to the result of TinyNet3. TinyNet3 is also not completely conclusive as not all four images of the overlap agree, but only three of them agree on the presence of the case. In the image of case number 12 in Figures 32e and 32b, the situation is similar, with a lot of noise and inconclusive information, especially for the TinyNet3. Here, the MMSE PWF agrees in one position with all four overlapping images, which also happen to be positively predicted by TinyNet3 but with only one single image. For the last case, no method was able to detect the presence of the case. The conditions of the image are favorable, as the wave height is quite low and the incidence angle of $>37°$ not too low. Therefore, the boat might have been too small to be detected or its position is not reported correctly and it is not present in this scene. In all three cases, the environmental conditions are favorable, as the wave height and wind speed are comparable and the incidence angle is high. When we compare the number of false positives for these three cases they might be explainable by the difference in wave height and wind speed, which is higher for case 10 and lower for case 12 and 15. These experiments show that it is not impossible but still very difficult for both models to detect known cases in Sentinel-1 images. It also shows that it is beneficial to combine both models to get a more clear image. Further research needs to be done with more cases and more detailed information about them, like the size and material of the boat or the number of people on board.



(a) MMSE PWF Case 10 (b) MMSE PWF Case 12 (c) MMSE PWF Case 15

(d) TinyNet3 Case 10 (e) TinyNet3 Case 12 (f) TinyNet3 Case 15

Figure 32: Three different cases analyzed with TinyNet3 and MMSE PWF.

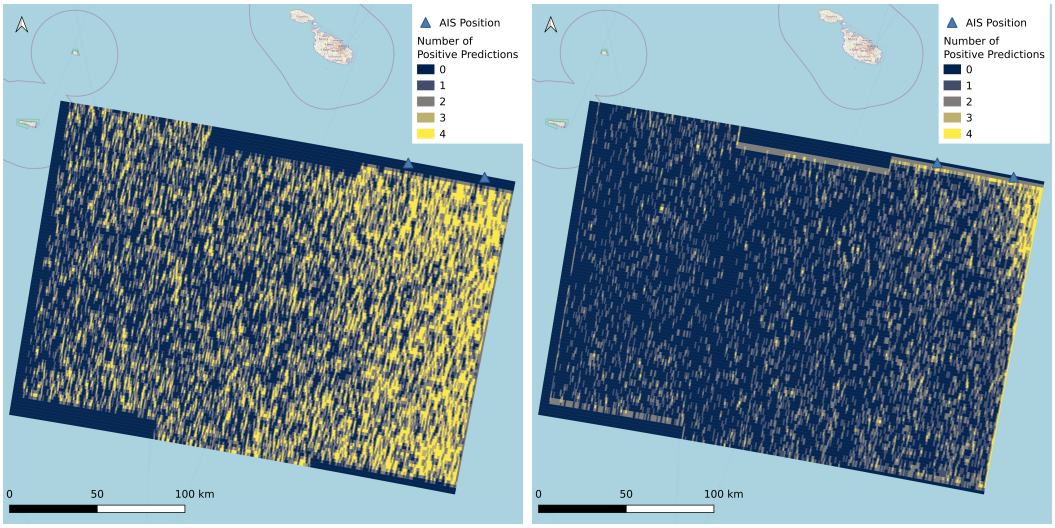
4.6 Analysis of Complete Image

To investigate if these models are suitable for monitoring the Search and Rescue area, we processed a complete scene, which was split into 55284 images as described in Section 3.6.5. The main focus lies in the time it takes to get the complete analysis. Table 7 shows the execution time and speed for each processing step. The download took 7 minutes with 20Mb/s, and the preprocessing took 11 minutes to finish. These processes are not fully optimized but are reasonably fast. The algorithms should be at least as fast as these steps so

Table 7: Execution time of the different processing steps.

Operation	Time $[s]$	Speed
Download	420	20 $[Mb/s]$
Preprocess	660	84 $[FPS]$
TinyNet3	21.4	2586 $[FPS]$
MMSE PWF	573	96 $[FPS]$

that they do not become the bottleneck. The TinyNet3 was able to analyze all images in only 21.4 seconds on a Nvidia RTX 4090 graphics card and a batch size of 256, which means it processed 2586 frames per second (FPS). The MMSE PWF, on the other hand, took 573 seconds, which is only 96 FPS, with 16 parallel Threads on an Intel i9-13900K CPU and 128 GB RAM. In this scenario, the CNN-based model is far superior, but the traditional algorithm could be improved by using bigger cutouts. Which model performs better will depend on the hardware available, but both models are reasonably fast for use in Search and Rescue. When we look at the quality of the result in Figure 33, we can conclude that both methods seem to produce a lot of false positives. The dependency on the incidence angle appears to be much stronger as the analysis of the test dataset suggested. The incidence angle increases to the west, where less false positives are seen. The main reason for the high number of false positives is probably the high waves and high wind speed as can be seen in Table 8. Both are higher than any image in the dataset. While that makes it harder to judge the quality of the results, it opens up a new insight: Similar to what Li et al. [19] suggested, the ML approach generalizes much better than the traditional algorithm because we can see that TinyNet3 struggles more on the edges but a lot less overall. The edges have no-data-values that are not present in the dataset and therefore the CNN could not learn that pattern properly. While the MMSE PWF cannot be further tuned without compromising on detectability, the ML model could be further trained to overcome this problem.



(a) MMSE PWF (b) TinyNet3

Figure 33: Analysis of a complete image with MMSE PWF and TinyNet3. Background map: OpenStreetMap®

Table 8: Environmental conditions for the complete image.

wave height [m]			wind speed [m/s]			wind direction[°]		
min	max	mean	min	max	mean	min	max	mean
2.06	5.80	4.25	12.44	16.24	14.04	82.78	99.94	93.62

5 Conclusion and further directions

In this thesis, we have achieved a milestone by successfully pushing the boundaries of tiny ship detectability using a novel Sentinel-1 dataset in combination with new CNN designs. Two novel design choices are introduced, which show noticeable improvements for CNNs in tiny object detection. We then explored its application in civil Search and Rescue by analyzing real cases and a theoretical monitoring application. This study, which encompasses various aspects from model design to limitations and possible applications, opens up numerous avenues for further exploration.

The novel approach to building a ship-water classification dataset with AIS to detect ship locations and CFAR to detect water locations successfully led to a medium-sized dataset with over 6000 images of over 1000 Sentinel-1 scenes. The size was big enough to successfully train lightweight CNNs from scratch, but a larger dataset would likely allow for better training and a better understanding of the limitations for detectability. A low-hanging fruit would be to repeat the experiments but switch the test dataset with the validation dataset. Rerunning the same evaluation, this time with twice as many images available for testing might lead to a more accurate evaluation of the limitations. Since the training data is still limited, it could be extended with augmentation methods beyond the flipping operation used in this study. Some researchers have proposed different methods especially designed to work with SAR images [69], or to simulate the training data of different targets in different environments based on statistical properties [70], machine learning techniques such as Generative Adversarial Networks [71], [72], or physical accurate models [73]. Some of these may decrease the amount of false positives by simulating realistic water patches of different sea states. Extending the dataset with publicly available datasets could mitigate the strong bias towards calm sea states and less intense background scattering. They often come with bounding boxes that could add more ship and water chips. Higher-order labels like bounding boxes could be derived from Grad-CAM ++ or MMSE PWF to improve the dataset's quality. This would require a lot of manual labor, but make the dataset compatible with detection tasks and other publicly available datasets. As explained in

Section 3.1, it is challenging to determine true water patches. The settings used in this study led to a clean dataset suitable for training with cross entropy loss. Lowering the CFAR threshold for the water patches would reduce the bias towards calm sea and more ship water pairs would be accepted at the cost of noisy labels. However, these noisy labels are a problem for the cross entropy loss [74] and need to be adressed with regularization and more suitable loss functions [74]–[76]. This could be used in future research with this dataset approach.

We also found that CNNs must be designed with special care to compete with adaptive-threshold-based methods. An architecture sweep was carried out, and a model was found that outcompetes the traditional algorithm and ML state-of-the-art algorithms. The results on the new dataset are noticeably better than what can be found in the literature [15]. The model uses squeeze excitation and residuals with max pooling instead of strided convolutions. Especially the max pooling operation is a novel design choice and noticeably contributed to the model performance. The experiments showed that average pooling and strided convolutions, common operations for many architectures, are unsuitable for tiny ship detection. Often, the valuable information is only a couple of pixels big and not carried along with these operations. The added computational complexity of this elemntwise operation is justified by the improvement. So far, that was only reported for global pooling operations by Pawlowski et al. in tiny object detection [20]. Using more 3x3 convolutions, as done in the TinyBlock, and changing the global average pooling of the squeeze excitation to max pooling may result in further improvements. Less wide model designs performed better on this binary classification task, meaning that D1 of Radosavovic et al. [22] does not hold in this case. That might be different for classification tasks with more than two classes.

Furthermore, we tested 3D convolutions and complex-valued convolutions. None of these showed better performance compared to the 2D convolutions. However, the 3D convolutions showed better median results, which hints that they are easier to train and worth further exploring. The complex-valued convolutions could have performed better, and the valuable phase information seemed not unlocked. This study only briefly tested simple approaches with

CReLU, but without normalization, so there might be more potential that could not be explored in this study. Investigating the complex-valued convolutions is still promising, and the performance might increase with suitable batch normalization or other activation functions as the research suggests [62], [77]–[80]. However, it is unclear if the model will capture meaningful phase information or simply introduce a new dimension to separate the data, as suggested by Trabelsi et al. [81]. With the current state of research, it is hard to draw a definite conclusion. All these techniques could be extended to the whole network architecture, coupled with group convolutions or depthwise convolutions. A network architecture search with more variations would help to determine further which design choices influence the performance.

We introduced a new combination of dilated convolution to increase the receptive field of a convolution layer, the so-called Reception block. It showed promising results in this study with a higher median score during training and the highest FM3 score on the validation set. Further research might be done incorporating this idea into other blocks and using it in more parts of the network. Since the receptive field plays a noticeable role, transformer-based models are especially interesting as they keep a global receptive field [82], [83]. They showed promising results in the literature regarding remote sensing and tiny object detection [72], [82], [83].

The network could also be improved by adding more outputs like vessel type and activity or by using a regression task to predict the vessel size. This information is already available in the dataset. During training, the images were only preprocessed with standard image normalization, with a mean and standard deviation of 0.5. This preprocessing is an elementary part of machine learning and might influence performance noticeably but could not be investigated closely in this study.

The experiments on real-world Search and Rescue cases show that Sentinel-1 images might be helpful in the investigation of past cases. None of the methods showed satisfactory results alone, but only in combination they gave conclusive results on the presence of a ship in the image. With an ensemble of detection algorithms tuned and trained on this dataset, a framework for investigating past cases could be set up. The results obtained in this study

could not yet be effectively used for regular monitoring with Sentinel-1 images. The traditional algorithm produces far too many false positives, and raising the threshold would lead to more false negatives. The TinyNet3, on the other hand, could be further trained to better distinguish between ship and water; however, a larger dataset is required to do so. To this goal, the dataset-building approach of this study can not be used because it introduces a bias towards high incident angles and less diverse water images. After more suitable training, the architecture itself could be used for the inference of complete images, as it computes extremely fast with less than 30 seconds on a consumer-grade GPU. The proposed method of using four overlapping patches already leads to satisfactory detection performance with roughly one nautical mile precision. Higher-quality image detection tasks like bounding boxes or polygons are, therefore, optional. Further research needs to be done before Sentinel-1 images will be usable in the context of Search and Rescue monitoring.

References

[1] *Mediterranean | Missing Migrants Project*. [Online]. Available: https://missingmigrants.iom.int/region/mediterranean#close (visited on 10/04/2023).

[2] P. d'Argent and M. Kuritzky, "Refoulement by Proxy? The Mediterranean Migrant Crisis and the Training of Libyan Coast Guards by EUNAVFOR MED Operation Sophia," in *Israel Yearbook on Human Rights, Volume 47 (2017)*, Y. Dinstein, Ed., Brill | Nijhoff, Jan. 2017, pp. 233–264, ISBN: 978-90-04-34195-1 978-90-04-34194-4. DOI: 10.1163/9789004341951_010. [Online]. Available: https://brill.com/view/book/edcoll/9789004341951/B9789004341951_010.xml (visited on 01/16/2024).

[3] O. Kulikowski, *Sea-Watch vs. Frontex • Sea-Watch e.V.* de-DE, Oct. 2023. [Online]. Available: https://sea-watch.org/sea-watch-vs-frontex/ (visited on 04/05/2024).

[4] *Final Report Summary - SAGRES (Services Activations for GRowing Eurosur's Success) | FP7*, en. [Online]. Available: https://cordis.europa.eu/project/id/313305/reporting (visited on 10/19/2023).

[5] P. Lanz, A. Marino, T. Brinkhoff, F. Köster, and M. Möller, "The InflateSAR Campaign: Evaluating SAR Identification Capabilities of Distressed Refugee Boats," *Remote Sensing*, vol. 12, no. 21, p. 3516, 2020. DOI: 10.3390/rs12213516.

[6] P. Lanz, A. Marino, T. Brinkhoff, F. Köster, and M. Möller, "The InflateSAR Campaign: Testing SAR Vessel Detection Systems for Refugee Rubber Inflatables," *Remote Sensing*, vol. 13, no. 8, p. 1487, 2021. DOI: 10.3390/rs13081487.

[7] P. Lanz, A. Marino, M. D. Simpson, T. Brinkhoff, F. Köster, and M. Möller, "The InflateSAR Campaign: Developing Refugee Vessel Detection Capabilities with Polarimetric SAR," en, *Remote Sensing*, vol. 15, no. 8, p. 2008, Apr. 2023, ISSN: 2072-4292. DOI: 10.3390/rs15082008.

[Online]. Available: https://www.mdpi.com/2072-4292/15/8/2008 (visited on 01/09/2024).

[8] F. Topputo, M. Massari, R. Lombardi, *et al.*, "Space shepherd: Search and rescue of illegal immigrants in the mediterranean sea through satellite imagery," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Milan, Italy: IEEE, Jul. 2015, pp. 4852–4855, ISBN: 978-1-4799-7929-5. DOI: 10.1109/IGARSS.2015.7326917. [Online]. Available: http://ieeexplore.ieee.org/document/7326917/ (visited on 04/05/2024).

[9] U. Kanjir, "Detecting migrant vessels in the Mediterranean Sea: Using Sentinel-2 images to aid humanitarian actions," en, *Acta Astronautica*, vol. 155, pp. 45–50, Feb. 2019, ISSN: 00945765. DOI: 10.1016/j.actaastro.2018.11.012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S009457651830393X (visited on 04/05/2024).

[10] webmaster, *Copernicus Maritime Surveillance. Use Case - Maritime Safety*, en-GB. [Online]. Available: https://www.emsa.europa.eu/copernicus/cms-cases/item/5139-copernicus-maritime-surveillance-use-case-maritime-safety.html (visited on 04/02/2024).

[11] *Dokumente des 2. Workshops über Forschung und Entwicklung für die Entwicklung des CopernicusEU-Sicherheitsdienstes vom 12.12.2023*, de. [Online]. Available: https://fragdenstaat.de/anfrage/dokumente-des-2-workshops-ueber-forschung-und-entwicklung-fuer-die-entwicklung-des-copernicuseu-sicherheitsdienstes-vom-12-12-2023-1/ (visited on 04/01/2024).

[12] G. Melillos, K. Themistocleous, C. Danezis, *et al.*, "Detecting migrant vessels in the Cyprus region using Sentinel-1 SAR data," in *Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies IV*, H. Bouma, R. J. Stokes, Y. Yitzhaky, and R. Prabhu, Eds., Online Only, United Kingdom: SPIE, Sep. 2020, p. 20, ISBN: 978-1-5106-3897-6 978-1-5106-3898-3. DOI: 10.1117/12.2573744. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11542/2573744/Detecting-migrant-vessels-in-the-Cyprus-

region-using-Sentinel-1/10.1117/12.2573744.full (visited on 04/05/2024).

[13] R. Pelich, N. Longepe, G. Mercier, G. Hajduch, and R. Garello, "Performance evaluation of Sentinel-1 data in SAR ship detection," en, in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Milan, Italy: IEEE, Jul. 2015, pp. 2103–2106, ISBN: 978-1-4799-7929-5. DOI: 10.1109/IGARSS.2015.7326217. [Online]. Available: http://ieeexplore.ieee.org/document/7326217/ (visited on 09/27/2023).

[14] R. Torres, P. Snoeij, D. Geudtner, *et al.*, "GMES Sentinel-1 mission," en, *Remote Sensing of Environment*, vol. 120, pp. 9–24, May 2012, ISSN: 00344257. DOI: 10.1016/j.rse.2011.05.028. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0034425712000600 (visited on 04/07/2024).

[15] F. Paolo, D. Kroodsma, J. Raynor, *et al.*, "Satellite mapping reveals extensive industrial activity at sea," en, *Nature*, vol. 625, no. 7993, pp. 85–91, Jan. 2024, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06825-8. [Online]. Available: https://www.nature.com/articles/s41586-023-06825-8 (visited on 01/08/2024).

[16] C. Bentes, D. Velotto, and S. Lehner, "Analysis of ship size detectability over different TerraSAR-X modes," in *2014 IEEE Geoscience and Remote Sensing Symposium*, Quebec City, QC: IEEE, Jul. 2014, pp. 5137–5140, ISBN: 978-1-4799-5775-0. DOI: 10.1109/IGARSS.2014.6947654. [Online]. Available: http://ieeexplore.ieee.org/document/6947654/ (visited on 10/24/2023).

[17] C. Liu, "A dual-polarization ship detection algorithm," *Defence Research and Development Canada*, vol. R109, 2015. [Online]. Available: https://cradpdf.drdc-rddc.gc.ca/PDFS/unc212/p803086_A1b.pdf (visited on 03/17/2024).

[18] A. Marino, "A Notch Filter for Ship Detection With Polarimetric SAR Data," *IEEE Journal of Selected Topics in Applied Earth Observations*

and Remote Sensing, vol. 6, no. 3, pp. 1219–1232, Jun. 2013, ISSN: 1939-1404, 2151-1535. DOI: 10.1109/JSTARS.2013.2247741. [Online]. Available: https://ieeexplore.ieee.org/document/6475202/ (visited on 03/16/2024).

[19] J. Li, C. Xu, H. Su, L. Gao, and T. Wang, "Deep Learning for SAR Ship Detection: Past, Present and Future," en, *Remote Sensing*, vol. 14, no. 11, p. 2712, Jan. 2022, Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2072-4292. DOI: 10.3390/rs14112712. [Online]. Available: https://www.mdpi.com/2072-4292/14/11/2712 (visited on 09/27/2023).

[20] N. Pawlowski, S. Bhooshan, N. Ballas, F. Ciompi, B. Glocker, and M. Drozdzal, "Needles in Haystacks: On Classifying Tiny Objects in Large Images," 2019, Publisher: [object Object] Version Number: 2. DOI: 10.48550/ARXIV.1908.06037. [Online]. Available: https://arxiv.org/abs/1908.06037 (visited on 03/05/2024).

[21] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "R^2-CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5512–5524, Aug. 2019, ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2019.2899955. [Online]. Available: https://ieeexplore.ieee.org/document/8672899/ (visited on 03/05/2024).

[22] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, *Designing Network Design Spaces*, arXiv:2003.13678 [cs], Mar. 2020. [Online]. Available: http://arxiv.org/abs/2003.13678 (visited on 03/07/2024).

[23] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," 2018, Publisher: [object Object] Version Number: 1. DOI: 10.48550/ARXIV.1807.11164. [Online]. Available: https://arxiv.org/abs/1807.11164 (visited on 03/04/2024).

[24] T. Zhang, X. Zhang, X. Ke, *et al.*, "LS-SSDD-v1.0: A Deep Learning Dataset Dedicated to Small Ship Detection from Large-Scale Sentinel-1 SAR Images," *Remote Sensing*, vol. 12, no. 18, p. 2997, 2020. DOI: 10.3390/rs12182997.

[25] B. Li, B. Liu, L. Huang, W. Guo, Z. Zhang, and W. Yu, "OpenSARShip 2.0: A large-volume dataset for deeper interpretation of ship targets in Sentinel-1 imagery," in *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, Beijing: IEEE, Nov. 2017, pp. 1–5, ISBN: 978-1-5386-4519-2. DOI: 10.1109/BIGSARDATA.2017.8124929. [Online]. Available: http://ieeexplore.ieee.org/document/8124929/ (visited on 01/23/2024).

[26] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds," en, *Remote Sensing*, vol. 11, no. 7, p. 765, Mar. 2019, ISSN: 2072-4292. DOI: 10.3390/rs11070765. [Online]. Available: https://www.mdpi.com/2072-4292/11/7/765 (visited on 01/15/2024).

[27] *AIS transponders*. [Online]. Available: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx (visited on 03/20/2024).

[28] Y. Zhang and Y. Hao, "A Survey of SAR Image Target Detection Based on Convolutional Neural Networks," *Remote Sensing*, vol. 14, no. 24, p. 6240, 2022. DOI: 10.3390/rs14246240.

[29] J. Van Zyl and Y. Kim, *Synthetic Aperture Radar Polarimetry*, en, 1st ed. Wiley, Oct. 2011, ISBN: 978-1-118-11511-4 978-1-118-11610-4. DOI: 10.1002/9781118116104. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118116104 (visited on 04/11/2024).

[30] *Microwaves and Radar Institute - DLR SAR Calibration Center*. [Online]. Available: https://www.dlr.de/hr/en/desktopdefault.aspx/tabid-2459/3715_read-53570/ (visited on 04/19/2024).

[31] A. Flores, K. Herndon, R. Thapa, and E. Cherrington, "Synthetic Aperture Radar (SAR) Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation," en, 2019, Publisher: [object Ob-

ject]. DOI: 10.25966/NR2C-S697. [Online]. Available: https://gis1.servirglobal.net/TrainingMaterials/SAR/SARHB_FullRes.pdf (visited on 01/11/2024).

[32] D. Small and A. Schubert, *Multilook*. [Online]. Available: https://step.esa.int/main/wp-content/help/versions/10.0.0/snap-toolboxes/eu.esa.microwavetbx.sar.op.sar.processing.ui/operators/MultilookOp.html (visited on 01/10/2024).

[33] I. Hajnsek and Y.-L. Desnos, Eds., *Polarimetric Synthetic Aperture Radar: Principles and Application* (Remote sensing and digital image processing Volume 25), eng. Cham: Springer, 2021, ISBN: 978-3-030-56504-6 978-3-030-56502-2.

[34] J.-S. Lee and E. Pottier, *Polarimetric radar imaging: from basics to applications* (Optical science and engineering 142). Boca Raton: CRC Press, 2009, OCLC: ocn277040880, ISBN: 978-1-4200-5497-2.

[35] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," 2014, Publisher: [object Object] Version Number: 3. DOI: 10.48550/ARXIV.1409.0575. [Online]. Available: https://arxiv.org/abs/1409.0575 (visited on 03/20/2024).

[36] T. Zhang, X. Zhang, J. Li, *et al.*, "SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis," en, *Remote Sensing*, vol. 13, no. 18, p. 3690, Sep. 2021, ISSN: 2072-4292. DOI: 10.3390/rs13183690. [Online]. Available: https://www.mdpi.com/2072-4292/13/18/3690 (visited on 01/15/2024).

[37] X. Sun, Z. Wang, Y. Sun, W. Diao, Y. Zhang, and K. Fu, "AIR-SARShip-1.0: High-resolution SAR Ship Detection Dataset (in English)," *Journal of Radars*, vol. 8, p. 852, 2019, ISSN: 2095-283X. DOI: 10.12000/JR19097. [Online]. Available: https://radars.ac.cn/en/article/doi/10.12000/JR19097.

[38] *Task Team on AIS Data — UN-CEBD*. [Online]. Available: https://unstats.un.org/bigdata/task-teams/ais/index.cshtml (visited on 01/16/2024).

[39] Genyuan Wang, X.-g. Xia, and V. Chan, "Dual-speed SAR imaging of moving targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 368–379, Jan. 2006, ISSN: 0018-9251, 1557-9603, 2371-9877. DOI: 10.1109/TAES.2006.1603430. [Online]. Available: https://ieeexplore.ieee.org/document/1603430/ (visited on 04/04/2024).

[40] G. Korres, M. Ravdas, A. Zacharioudaki, D. Denaxa, and M. Sotiropoulou, *Mediterranean Sea Waves Reanalysis (CMEMS Med-Waves, MedWAM3 system): MEDSEA_multiyear_wav_006_012*, en, 2021. DOI: 10.25423/CMCC/MEDSEA_MULTIYEAR_WAV_006_012. [Online]. Available: https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=MEDSEA_MULTIYEAR_WAV_006_012 (visited on 01/11/2024).

[41] European Union-Copernicus Marine Service, *Global Ocean Hourly Sea Surface Wind and Stress from Scatterometer and Model*, en, 2022. DOI: 10.48670/MOI-00305. [Online]. Available: https://resources.marine.copernicus.eu/product-detail/WIND_GLO_PHY_L4_NRT_012_004/INFORMATION (visited on 01/11/2024).

[42] C3S, *ERA5 hourly data on single levels from 1940 to present*, 2018. DOI: 10.24381/CDS.ADBB2D47. [Online]. Available: https://cds.climate.copernicus.eu/doi/10.24381/cds.adbb2d47 (visited on 01/11/2024).

[43] B. Tings, C. Bentes, D. Velotto, and S. Voinov, "Modelling ship detectability depending on TerraSAR-X-derived metocean parameters," en, *CEAS Space Journal*, vol. 11, no. 1, pp. 81–94, Mar. 2019, ISSN: 1868-2502, 1868-2510. DOI: 10.1007/s12567-018-0222-8. [Online]. Available: http://link.springer.com/10.1007/s12567-018-0222-8 (visited on 04/24/2024).

[44] H. Finn, "Adaptive detection in clutter," in *Fifth Symposium on Adaptive Processes*, USA: IEEE, Oct. 1966, pp. 562–567. DOI: 10.1109/SAP.1966.271149. [Online]. Available: http://ieeexplore.ieee.org/document/4043676/ (visited on 03/13/2024).

[45] L. Novak, M. Burl, R. Chaney, and G. Owirka, "Optimal Processing of Polarimetric Synthetic-Aperture Radar Imagery," *The Lincoln Laboratory Journal*, vol. 3, no. 2, pp. 273–290, 1990. [Online]. Available: https://archive.ll.mit.edu/publications/journal/pdf/vol03_no2/3.2.5.opticalprocessingSAR.pdf (visited on 03/12/2024).

[46] W. An, M. Lin, C. Xie, G. Zhou, and X. Yuan, "Modified polarimetric whitening filter for polarimetric SAR data," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, Melbourne, Australia: IEEE, Jul. 2013, pp. 2481–2484, ISBN: 978-1-4799-1114-1. DOI: 10.1109/IGARSS.2013.6723324. [Online]. Available: http://ieeexplore.ieee.org/document/6723324/ (visited on 03/16/2024).

[47] L. Novak, M. Sechtin, and M. Cardullo, "Studies of target detection algorithms that use polarimetric radar data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 150–165, Mar. 1989, ISSN: 0018-9251. DOI: 10.1109/7.18677. [Online]. Available: http://ieeexplore.ieee.org/document/18677/ (visited on 03/16/2024).

[48] A. Marino, W. Dierking, and C. Wesche, "A Depolarization Ratio Anomaly Detector to Identify Icebergs in Sea Ice Using Dual-Polarization SAR Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5602–5615, Sep. 2016, ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2016.2569450. [Online]. Available: http://ieeexplore.ieee.org/document/7485879/ (visited on 03/17/2024).

[49] A. Passah, S. N. Sur, A. Abraham, and D. Kandar, "Synthetic Aperture Radar image analysis based on deep learning: A review of a decade of research," en, *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106 305, Aug. 2023, ISSN: 09521976. DOI: 10.1016/j.engappai.2023.106305. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095219762300489X (visited on 03/20/2024).

[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," 2017, Publisher: [object Object] Version Number: 2. DOI: 10.48550/ARXIV.1708.02002. [Online]. Available: https://arxiv.org/abs/1708.02002 (visited on 03/20/2024).

[51] Y. Ren, W. Jiang, and Y. Liu, "A New Architecture of a Complex-Valued Convolutional Neural Network for PolSAR Image Classification," en, *Remote Sensing*, vol. 15, no. 19, p. 4801, Oct. 2023, ISSN: 2072-4292. DOI: 10.3390/rs15194801. [Online]. Available: https://www.mdpi.com/2072-4292/15/19/4801 (visited on 01/24/2024).

[52] L. Li, K. Jamieson, A. Rostamizadeh, *et al.*, *A System for Massively Parallel Hyperparameter Tuning*, arXiv:1810.05934 [cs, stat], Mar. 2020. [Online]. Available: http://arxiv.org/abs/1810.05934 (visited on 03/05/2024).

[53] I. Rodriguez-Conde, C. Campos, and F. Fdez-Riverola, "Optimized convolutional neural network architectures for efficient on-device vision-based object detection," en, *Neural Computing and Applications*, vol. 34, no. 13, pp. 10 469–10 501, Jul. 2022, ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-021-06830-w. [Online]. Available: https://link.springer.com/10.1007/s00521-021-06830-w (visited on 03/04/2024).

[54] J. Wang, W. Yang, H. Guo, R. Zhang, and G.-S. Xia, "Tiny Object Detection in Aerial Images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy: IEEE, Jan. 2021, pp. 3791–3798, ISBN: 978-1-72818-808-9. DOI: 10.1109/ICPR48806.2021.9413340. [Online]. Available: https://ieeexplore.ieee.org/document/9413340/ (visited on 03/05/2024).

[55] Z. Chen, Y. Liang, Z. Yu, *et al.*, "TO–YOLOX: A pure CNN tiny object detection model for remote sensing images," en, *International Journal of Digital Earth*, vol. 16, no. 1, pp. 3882–3904, Oct. 2023, ISSN: 1753-8947, 1753-8955. DOI: 10.1080/17538947.2023.2261901. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/17538947.2023.2261901 (visited on 03/05/2024).

[56] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385 [cs], Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03385 (visited on 03/07/2024).

[57] X. Tan, M. Li, P. Zhang, Y. Wu, and W. Song, "Complex-Valued 3-D Convolutional Neural Network for PolSAR Image Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 6, pp. 1022–1026, Jun. 2020, ISSN: 1545-598X, 1558-0571. DOI: 10.1109/LGRS.2019.2940387. [Online]. Available: https://ieeexplore.ieee.org/document/8864110/ (visited on 04/11/2024).

[58] H. Dong, L. Zhang, and B. Zou, "PolSAR Image Classification with Lightweight 3D Convolutional Networks," en, *Remote Sensing*, vol. 12, no. 3, p. 396, Jan. 2020, ISSN: 2072-4292. DOI: 10.3390/rs12030396. [Online]. Available: https://www.mdpi.com/2072-4292/12/3/396 (visited on 04/11/2024).

[59] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, "Hyperspectral Image Spatial Super-Resolution via 3D Full Convolutional Neural Network," en, *Remote Sensing*, vol. 9, no. 11, p. 1139, Nov. 2017, ISSN: 2072-4292. DOI: 10.3390/rs9111139. [Online]. Available: http://www.mdpi.com/2072-4292/9/11/1139 (visited on 04/11/2024).

[60] Y. Li, H. Zhang, and Q. Shen, "Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network," en, *Remote Sensing*, vol. 9, no. 1, p. 67, Jan. 2017, ISSN: 2072-4292. DOI: 10.3390/rs9010067. [Online]. Available: http://www.mdpi.com/2072-4292/9/1/67 (visited on 04/11/2024).

[61] Q. Li, Q. Wang, and X. Li, "Exploring the Relationship Between 2D/3D Convolution for Hyperspectral Image Super-Resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 10, pp. 8693–8703, Oct. 2021, ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2020.3047363. [Online]. Available: https://ieeexplore.ieee.org/document/9334383/ (visited on 04/11/2024).

[62] H. Parikh, S. Patel, and V. Patel, "Classification of SAR and PolSAR images using deep learning: A review," en, *International Journal of Image and Data Fusion*, vol. 11, no. 1, pp. 1–32, Jan. 2020, ISSN: 1947-9832, 1947-9824. DOI: 10.1080/19479832.2019.1655489. [Online]. Available:

https://www.tandfonline.com/doi/full/10.1080/19479832.2019.1655489 (visited on 04/12/2024).

[63] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, *Squeeze-and-Excitation Networks*, arXiv:1709.01507 [cs], May 2019. [Online]. Available: http://arxiv.org/abs/1709.01507 (visited on 03/27/2024).

[64] T.-D. Truong, V.-T. Nguyen, and M.-T. Tran, "Lightweight Deep Convolutional Network for Tiny Object Recognition," in *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 675–682, ISBN: 978-989-758-276-9. DOI: 10.5220/0006752006750682. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006752006750682 (visited on 03/05/2024).

[65] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV: IEEE, Mar. 2018, pp. 839–847, ISBN: 978-1-5386-4886-5. DOI: 10.1109/WACV.2018.00097. [Online]. Available: https://ieeexplore.ieee.org/document/8354201/ (visited on 04/21/2024).

[66] D. Velotto, F. Nunziata, M. Migliaccio, and S. Lehner, "Dual-Polarimetric TerraSAR-X SAR Data for Target at Sea Observation," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1114–1118, Sep. 2013, ISSN: 1545-598X, 1558-0571. DOI: 10.1109/LGRS.2012.2231048. [Online]. Available: http://ieeexplore.ieee.org/document/6472773/ (visited on 03/13/2024).

[67] P. Lanz, A. Marino, M. D. Simpson, T. Brinkhoff, F. Köster, and M. Möller, "Correction: Lanz et al. The InflateSAR Campaign: Developing Refugee Vessel Detection Capabilities with Polarimetric SAR. Remote Sens. 2023, 15, 2008," en, *Remote Sensing*, vol. 15, no. 22, p. 5344, Nov. 2023, ISSN: 2072-4292. DOI: 10.3390/rs15225344. [Online]. Avail-

able: https://www.mdpi.com/2072-4292/15/22/5344 (visited on 01/09/2024).

[68] L. Færch, W. Dierking, N. Hughes, and A. P. Doulgeris, "A comparison of constant false alarm rate object detection algorithms for iceberg identification in L- and C-band SAR imagery of the Labrador Sea," en, *The Cryosphere*, vol. 17, no. 12, pp. 5335–5355, Dec. 2023, ISSN: 1994-0424. DOI: 10.5194/tc-17-5335-2023. [Online]. Available: https://tc.copernicus.org/articles/17/5335/2023/ (visited on 03/17/2024).

[69] R. Yang, R. Wang, Y. Deng, X. Jia, and H. Zhang, "Rethinking the Random Cropping Data Augmentation Method Used in the Training of CNN-Based SAR Image Ship Detector," en, *Remote Sensing*, vol. 13, no. 1, p. 34, Dec. 2020, ISSN: 2072-4292. DOI: 10.3390/rs13010034. [Online]. Available: https://www.mdpi.com/2072-4292/13/1/34 (visited on 04/23/2024).

[70] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional Neural Network With Data Augmentation for SAR Target Recognition," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2016, ISSN: 1545-598X, 1558-0571. DOI: 10.1109/LGRS.2015.2513754. [Online]. Available: http://ieeexplore.ieee.org/document/7393462/ (visited on 04/23/2024).

[71] M. Zhang, Z. Cui, X. Wang, and Z. Cao, "Data Augmentation Method of SAR Image Dataset," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia: IEEE, Jul. 2018, pp. 5292–5295, ISBN: 978-1-5386-7150-4. DOI: 10.1109/IGARSS.2018.8518825. [Online]. Available: https://ieeexplore.ieee.org/document/8518825/ (visited on 04/23/2024).

[72] M. Muzammul and X. Li, "A Survey on Deep Domain Adaptation and Tiny Object Detection Challenges, Techniques and Datasets," 2021, Publisher: [object Object] Version Number: 1. DOI: 10.48550/ARXIV.2107.07927. [Online]. Available: https://arxiv.org/abs/2107.07927 (visited on 03/05/2024).

[73] N. Ødegaard, "Realistic augmentations for SAR images of ships for machine learning," in *EUSAR 2024*, VDE VERLAG GMBH, 2024, pp. 1075–1080, ISBN: 978-3-8007-6286-6.

[74] D.-B. Wang, Y. Wen, L. Pan, and M.-L. Zhang, "Learning from Noisy Labels with Complementary Loss Functions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, pp. 10 111–10 119, May 2021, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v35i11.17213. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17213 (visited on 01/16/2024).

[75] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning From Noisy Labels With Deep Neural Networks: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8135–8153, Nov. 2023, ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2022.3152527. [Online]. Available: https://ieeexplore.ieee.org/document/9729424/ (visited on 01/16/2024).

[76] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized Loss Functions for Deep Learning with Noisy Labels," 2020, Publisher: [object Object] Version Number: 1. DOI: 10.48550/ARXIV.2006.13554. [Online]. Available: https://arxiv.org/abs/2006.13554 (visited on 01/16/2024).

[77] Z. Huang, M. Datcu, Z. Pan, and B. Lei, "Deep SAR-Net: Learning objects from signals," en, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 161, pp. 179–193, Mar. 2020, ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2020.01.016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0924271620300162 (visited on 03/25/2024).

[78] A. H. Oveis, E. Giusti, S. Ghio, and M. Martorella, "A Survey on the Applications of Convolutional Neural Networks for Synthetic Aperture Radar: Recent Advances," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 5, pp. 18–42, May 2022, ISSN: 0885-8985, 1557-959X. DOI: 10.1109/MAES.2021.3117369. [Online]. Available: https://ieeexplore.ieee.org/document/9656541/ (visited on 03/20/2024).

[79] A. Jamali, M. Mahdianpari, F. Mohammadimanesh, A. Bhattacharya, and S. Homayouni, "PolSAR Image Classification Based on Deep Convolutional Neural Networks Using Wavelet Transformation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, ISSN: 1545-598X, 1558-0571. DOI: 10.1109/LGRS.2022.3185118. [Online]. Available: https://ieeexplore.ieee.org/document/9802110/ (visited on 01/24/2024).

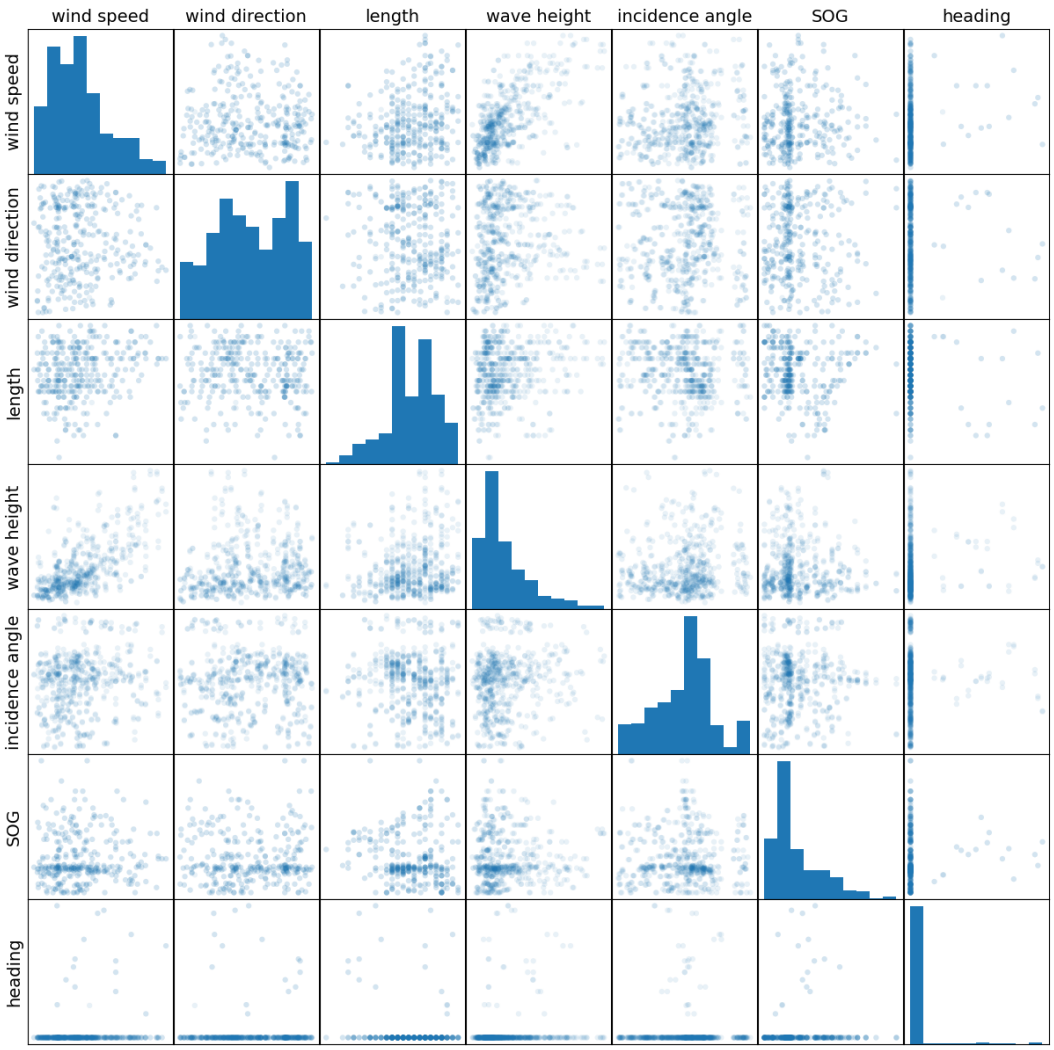
[80] H. Wang, F. Xu, and Y.-Q. Jin, "A Review of Polsar Image Classification: From Polarimetry to Deep Learning," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan: IEEE, Jul. 2019, pp. 3189–3192, ISBN: 978-1-5386-9154-0. DOI: 10.1109/IGARSS.2019.8899902. [Online]. Available: https://ieeexplore.ieee.org/document/8899902/ (visited on 01/24/2024).

[81] C. Trabelsi, O. Bilaniuk, Y. Zhang, *et al.*, "Deep Complex Networks," 2017, Publisher: arXiv Version Number: 4. DOI: 10.48550/ARXIV.1705.09792. [Online]. Available: https://arxiv.org/abs/1705.09792 (visited on 01/24/2024).

[82] Z. Tu, H. Talebi, H. Zhang, *et al.*, *MaxViT: Multi-Axis Vision Transformer*, arXiv:2204.01697 [cs], Sep. 2022. [Online]. Available: http://arxiv.org/abs/2204.01697 (visited on 03/07/2024).

[83] F. Kong, X.-y. Liu, and R. Henao, "Quantum Tensor Network in Machine Learning: An Application to Tiny Object Classification," 2021, Publisher: [object Object] Version Number: 1. DOI: 10.48550/ARXIV.2101.03154. [Online]. Available: https://arxiv.org/abs/2101.03154 (visited on 03/05/2024).

A Scattering matrix of the dataset



B Archicteture design test resuls

Name	Params	TP	TN	FP	FN	ACC
TinyNet	485570	291	295	19	23	0.93
TinyNet2	504482	287	302	12	27	0.94
TinyNet3	513988	296	302	12	18	0.95
TinyNet4	606470	297	298	16	17	0.95
TinyNet5	413298	293	292	22	21	0.93
TinyNet6	454258	292	302	12	22	0.95
TinyNet7	474840	288	303	11	26	0.94
TinyNet8	680422	296	297	17	18	0.94
TinyNet9	511972	289	298	16	25	0.93
TinyNet10	226084	296	296	18	18	0.94
TinyNet11	217012	299	289	25	15	0.94
TinyNet12	217012	283	297	17	31	0.92
TinyNet13	225892	284	303	11	30	0.93
TinyNet14	227380	283	296	18	31	0.92
TinyNet15	476476	288	298	16	26	0.93
TinyNet16	476572	298	282	32	16	0.92
TinyNet17	479788	273	302	12	41	0.92
TinyNet18	463372	288	291	23	26	0.92
TinyNet19	463468	297	287	27	17	0.93
TinyNet20	466684	277	291	23	37	0.90
TinyNet21	466420	299	292	22	15	0.94
TinyNet22	515128	275	291	23	39	0.90
TinyNet23	515248	287	288	26	27	0.92
TinyNet24	463366	284	279	35	30	0.90
TinyRegNetX	1792482	283	301	13	31	0.93
TinyRegNetY	2086838	288	300	14	26	0.94
TinyRegNetX2	2609746	297	295	19	17	0.94
TinyRegNetY2	3040602	289	306	8	25	0.95

TP: True positive, TN: True negative, FP: False positive, FN: False negative, ACC: Accuracy

C Baseline test results machine learning

Name	Params	Input	Channels	TP	TN	FP	FN	ACC
ResNet18	11174402	C2	C12	255	309	5	59	0.90
ResNet18	11174402	C2	C2	237	312	2	77	0.87
ResNet18	11174402	GRD ML	VH	206	311	3	108	0.82
ResNet18	11174402	GRD	VH,VV	243	311	3	71	0.88
EfficientNet-B2	7703524	C2	C12	166	312	2	148	0.76
EfficientNet-B2	7703524	C2	C2	192	311	3	122	0.80
EfficientNet-B2	7703524	GRD ML	VH	171	310	4	143	0.77
EfficientNet-B2	7703524	GRD	VH	176	312	2	138	0.78
DenseNet121	6947650	GRD	VH,VV	233	310	4	81	0.86
DenseNet121	6947650	C2	C2	217	309	5	97	0.84
DenseNet121	6947650	C2	C12	245	311	3	69	0.89
DenseNet121	6947650	GRD ML	VH,VV	156	307	7	158	0.74
EfficientNet-B1	6515458	GRD	VH	188	311	3	126	0.79
EfficientNet-B1	6515458	C2	C2	164	310	4	150	0.75
EfficientNet-B1	6515458	C2	C12	210	309	5	104	0.83
EfficientNet-B1	6515458	GRD ML	VH	173	309	5	141	0.77
RegNetY 800MF	5648794	GRD ML	VH	212	301	13	102	0.82
RegNetY 800MF	5648794	C2	C12	252	310	4	62	0.89
RegNetY 800MF	5648794	C2	C2	233	310	4	81	0.86
RegNetY 800MF	5648794	GRD	VH	238	306	8	76	0.87
ShuffleNetV2 X2.0	5348878	C2	C2	211	307	7	103	0.82
ShuffleNetV2 X2.0	5348878	GRD	VH,VV	224	311	3	90	0.85
ShuffleNetV2 X2.0	5348878	C2	C12	207	311	3	107	0.82
ShuffleNetV2 X2.0	5348878	GRD ML	VH,VV	213	311	3	101	0.83
MobileNetV3-Large	4204418	GRD ML	VH	188	306	8	126	0.79
MobileNetV3-Large	4204418	C2	C12	235	310	4	79	0.87
MobileNetV3-Large	4204418	C2	C2	214	311	3	100	0.84
MobileNetV3-Large	4204418	GRD	VH	201	312	2	113	0.82
EfficientNet-B0	4009822	GRD	VH	194	312	2	120	0.81
EfficientNet-B0	4009822	C2	C2	176	310	4	138	0.77
EfficientNet-B0	4009822	C2	C12	210	311	3	104	0.83
EfficientNet-B0	4009822	GRD ML	VH	164	312	2	150	0.76

Name	Params	Input	Channels	TP	TN	FP	FN	ACC
RegNetY 400MF	3903738	C2	C2	221	311	3	93	0.85
RegNetY 400MF	3903738	C2	C12	242	309	5	72	0.88
RegNetY 400MF	3903738	GRD	VH	237	309	5	77	0.87
RegNetY 400MF	3903738	GRD ML	VH	207	306	8	107	0.82
ShuffleNetV2 X1.5	2480458	C2	C2	210	307	7	104	0.82
ShuffleNetV2 X1.5	2480458	GRD ML	VH,VV	190	311	3	124	0.80
ShuffleNetV2 X1.5	2480458	C2	C12	210	307	7	104	0.82
ShuffleNetV2 X1.5	2480458	GRD	VH,VV	247	308	6	67	0.88
MobileNetV3-Small	1519730	GRD	VH	175	313	1	139	0.78
MobileNetV3-Small	1519730	GRD ML	VH	190	310	4	124	0.80
MobileNetV3-Small	1519730	C2	C12	200	309	5	114	0.81
MobileNetV3-Small	1519730	C2	C2	177	310	4	137	0.78
ShuffleNetV2 X1.0	1255438	C2	C12	224	308	6	90	0.85
ShuffleNetV2 X1.0	1255438	GRD	VH,VV	233	308	6	81	0.86
ShuffleNetV2 X1.0	1255438	C2	C2	217	309	5	97	0.84
ShuffleNetV2 X1.0	1255438	GRD ML	VH	173	312	2	141	0.77
ShuffleNetV2 X0.5	343626	GRD	VH	166	313	1	148	0.76
ShuffleNetV2 X0.5	343626	C2	C12	214	311	3	100	0.84
ShuffleNetV2 X0.5	343626	GRD ML	VH	138	312	2	176	0.72
ShuffleNetV2 X0.5	343626	C2	C2	213	310	4	101	0.83

TP: True positive, TN: True negative, FP: False positive, FN: False negative, ACC: Accuracy, GRD: GRD TNR, C2: C11, C22 and C12 from the covariance matrix

D Baseline test results traditional algorithms

Name	Size	Input	Channel	TP	TN	FP	FN	ACC
CFAR	7	C2 large	C11	253	294	20	61	0.87
CoDePol	11	GRD large	VH,VV	236	251	63	78	0.78
CrossDePol	11	GRD	VH,VV	238	286	28	76	0.83
MMSE PWF	7	C2	C2	279	284	30	35	0.90
NIS	5	GRD ML	VH,VV	117	212	102	197	0.52
PMF	11	GRD ML	VH,VV	199	277	37	115	0.76
PNF	3	C2	C2	255	294	20	59	0.87
PWF	11	C2 large	C2	175	232	82	139	0.65

For each algorithm only the the top-performing configuration regarding FM3 score can be shown here. TP: True positive, TN: True negative, FP: False positive, FN: False negative, ACC: Accuracy, GRD: GRD TNR, C2: C11, C22 and C12 from the covariance matrix, Size: Size of the target window