Universität Rostock

Traditio et Innovatio

Aus der Professur für Geodäsie und Geoinformatik

der Agrar- und Umweltwissenschaftlichen Fakultät

# Deep Learning-based Vessel Detection from Very High and Medium Resolution Optical Satellite Images as Component of Maritime Surveillance Systems

**Dissertation**

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

an der Agrar- und Umweltwissenschaftlichen Fakultät

der Universität Rostock

**vorgelegt von**

M.Sc. Sergey Voinov

aus Neustrelitz

Rostock, 2020

https://doi.org/10.18453/rosdok_id00002876

**Gutachter:**

**Prof. Dr. Ralf Bill,** Universität Rostock, Agrar- und Umweltwissenschaftliche Fakultät, Geodäsie und Geoinformatik

**Dr. Frank Heymann,** Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Solar-Terrestrische Physik

**Prof. Dr. Peter Reinartz,** Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Methodik der Fernerkundung, Photogrammetrie und Bildanalyse

**Jahr der Einreichung:** 2020

**Jahr der Verteidigung:** 2020

# Abstract

Today vessel detection from remote sensing images is increasingly becoming a crucial component in maritime surveillance applications. The increasing number of very high and medium resolution (VHR and MR) optical satellites shortens the revisit time as it was never before. This makes the technology especially attractive for a variety of maritime monitoring tasks. Nevertheless, it is quite a challenge to perform object detection on enormous large satellite images that cover several hundreds of square kilometers and derive results under near real time constraints.

This thesis presents an end-to-end multiclass vessel detection method from optical satellite images. The proposed workflow covers the complete processing chain and involves rapid image enhancement techniques, the fusion with automatic identification system (AIS) data, and the detection algorithm based on convolutional neural networks (CNN). To train the CNNs, two versions of training datasets were generated. The VHR training dataset was produced from the set of WorldView-[1-3] and GeoEye-1 images and contains about 40 000 of uniquely annotated vessels divided into 14 different classes. The MR training dataset was generated from the set of Landsat-8 images and contains about 14 000 of uniquely annotated vessels of 7 different classes.

The algorithms presented are implemented in the form of independent software processors and integrated in an automated processing chain as part of the Earth Observation Maritime Surveillance System (EO-MARISS). The solution developed from the methods presented has proven its usability within different projects and is used and further developed at the ground station of the German Aerospace Center (DLR) in Neustrelitz.

**Keywords:** optical remote sensing, vessel detection, ship detection, object detection, CNN, deep learning, AIS, data fusion

## Zusammenfassung

Schiffserkennung unter Nutzung von Satellitenbildern gewinnt heutzutage zunehmend an Bedeutung und leistet inzwischen in Anwendungen der Meeresüberwachung einen wichtigen Beitrag. Die zunehmende Anzahl optischer Satelliten mit sehr hoher (VHR) und mittlerer (MR) Auflösung ermöglicht eine deutliche Verkürzung der Wiederaufnahme gleicher Gebiete. Dies macht diese Technologie für eine Vielzahl von maritimen Überwachungsanwendungen immer attraktiver. Dabei ist die Objektdetektion auf sehr großen Satellitenbildern, welche mehrere hundert Quadratkilometer abdecken, eine enorme Herausforderung, insbesondere wenn dies in naher Echtzeit (NRT) geschehen soll.

In der vorliegenden Arbeit wird eine Methode zur Detektion von Schiffen unterschiedlicher Klassen in optischen Satellitenbildern vorgestellt. Diese gliedert sich in drei aufeinanderfolgende Funktionen: i) die Bildbearbeitung zur Verbesserung der Bildeigenschaften, ii) die Datenfusion mit den Daten des Automatischen Identifikation Systems (AIS) und iii) dem auf „Convolutional Neural Network“ (CNN) basierenden Detektionsalgorithmus. Um die CNNs zu trainieren, wurden zwei Versionen von Datensätzen erzeugt. Der VHR-Datensatz wurde aus WorldView-[1-3] sowie GeoEye-1 Bildern erstellt und enthält, eingeteilt in 14 verschiedene Klassen, mehr als 40 000 eindeutig annotierte Schiffe. Der MR-Datensatz wurde aus Aufnahmen vom Satelliten Landsat-8 generiert und enthält mehr als 14 000 eindeutig annotierte Schiffe in 7 verschiedenen Klassen.

Die vorgestellten Algorithmen wurden in Form eigenständiger Softwareprozessoren implementiert und als Teil des maritimen Erdbeobachtungssystems EO-MARISS (Earth Observation Maritime Surveillance System) in eine automatisierte Verarbeitungskette integriert. Diese Lösung hat ihre Anwendbarkeit innerhalb von verschiedenen Projekten unter Beweis gestellt und wird an der Bodenstation des Deutschen Zentrums für Luft- und Raumfahrt (DLR) in Neustrelitz eingesetzt und weiterentwickelt.

**Schlüsselwörter**: optische Fernerkundung, Schiffsdetektion, Objektdetektion, CNN, AIS, Datenfusion.

## Acknowledgements

I can hardly imagine that I would have ever finished this thesis without the support and help of many people. First of all, I would like to express my personal gratitude to:

**Prof. Dr. Ralf Bill,**

*Thank you for raising your interest in my work and accepting me as a PhD student. Thank you for the freedom you gave me in my research, your continuous support and your wise guidance during the entire research path.*

**Dr. Frank Heymann,**

*Thank you for your dedicated support and guidance. Your expertise, your thoughtful comments and recommendations on this dissertation were vital.*

**Prof. Dr. Peter Reinartz,**

*Thank you for your interest in my research, your support and your advices throughout the study.*

**Holger Maass,**

*Thank you for your great support in my initiative to accomplish this study. Your personal involvement in the process made it possible.*

**Egbert Schwarz,**

*Thank you for your patience, your trust and wise support. You did everything to make this dissertation happen. When it was necessary to concentrate on my thesis, you let me do this without interrupting me on my other duties. It is a great honor to be a member of your team!*

**Detmar Krause,**

*Thank you for your encouragement and enthusiasm in assisting me in any way, whether it is fruitful conversation or programming the PSM or solving some technical issues.*

I would like to acknowledge many colleagues from the Department National Ground Segment, especially: Matthias Berg, Holger Daedelow, Steffi Domris, Sebastian Hartung, Tobias Kaminski, Jens Pollex, Olga Schmidt and Hans-Hermann Vajen for using my software and giving me essential feedbacks as well as for their valuable contribution to training data production.

Finally, I would like to express my sincere thanks to my wife Galina for her unconditional love and constant support and encouragement during the research.

# Table of Contents

# List of abbreviations

| | |
|---|---|
| AIS | Automatic Identification System |
| CCTV | Closed-circuit television |
| AC | Atmospheric correction |
| AOI | Area of Interest |
| API | Application programming interface |
| CFAR | Constant false alarm rate |
| CFAR | Constant false alarm rate |
| CNN | Convolutional neural network |
| COG | Course-over-ground |
| CV | Computer vision |
| DB | Database |
| DEM | Digital Elevation Model |
| DFD | Deutsches Fernerkundungsdatenzentrum (German Remote Sensing Data Center) |
| DL | Deep learning |
| DLR | Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center) |
| DN | Digital numbers |
| DOS | Dark object subtraction |
| EMSA | European Maritime Safety Agency |
| EO | Earth observation |
| EU | European Union |
| GIS | Geographic information system |
| GNSS | Global Navigation Satellite System |
| GPU | Graphics processing unit |
| GSD | Ground sampling distance |
| HBB | Horizontal bounding box |
| HTTP | HyperText Transfer Protocol |
| IMO | International Maritime Organization |
| L1 | Level 1 |

| | |
|---|---|
| LBR | Length-beam ratio |
| LEO | Low-Earth orbit |
| LRIT | Long Range Identification and Tracking |
| MMSI | Maritime Mobile Service Identity |
| MR | Medium resolution |
| NMEA | National Marine Electronics Association |
| NMS | Non-Maximum Suppression |
| NRT | Near-real Time |
| OGC | Open Geospatial Consortium |
| OSM | OpenStreetMap |
| PCA | Principle component analysis |
| PSM | Processing System Management |
| RAM | Random Access Memory |
| RBB | Rotated bounding boxes |
| RPC | Rational polynomial coefficients |
| RPN | Region proposal network |
| RS | Remote Sensing |
| SAR | Synthetic aperture radar |
| SGD | Stochastic gradient descent |
| SOG | Speed-over-ground |
| SQL | Structured Query Language |
| SR | Surface reflectance |
| SSD | Solid-State Drive |
| TOA | Top-of-atmosphere |
| USGS | United States Geological Survey |
| VGG | Visual Geometry Group |
| VHF | Very high frequency |
| VHR | Very high resolution |

# 1 Introduction

From ancient times to today’s world, humanity has been highly dependent on the sea. The sea is a great supplier of food, energy and mineral resources. The maritime transport is responsible for carrying over 90% of all the goods in the world [1] which makes the global economy highly dependent on it. For this reason, it is of paramount importance to ensure that our seas are safe and secure, thus contributing to the global sustainable development.

Today’s maritime safety and security faces a number of various threats, natural and man-made [2]. Natural threats can be grouped into climatogenic, such as hurricanes or storms; and seismogenic, such as earthquakes followed by tsunamis. The man-made threats include anthropogenic activities (for example oil pollution) and different unlawful actions such as piracy, armed robbery, drug trafficking, warlike activities, illegal fishing, illegal border crossing and many others.

These threats are the forcing power for the authorities at different levels to develop and operate maritime surveillance systems. The main objective of such type of systems is the collection of wide range of data and transforming them into the knowledge about the current situation at sea [3].

Currently, satellite remote sensing technologies are being actively used within maritime surveillance systems [4] [5]. Satellite images are serving as valuable source of information for environmental and sea traffic monitoring.

This manuscript describes the methodology for vessel detection from optical satellite images. Presented algorithms are implemented as a set of independent software processors which are developed for use in near-real time (NRT) applications as part of maritime surveillance system.

## 1.1 Motivation and research objectives

This thesis addresses the problem of automated vessel detection from optical satellite imagery. This problem can be treated as a typical object detection task in the computer vision field. A number of researches and operational services over several years exist in the domain of satellite SAR (synthetic aperture radar) ship detection problem [6] [7]. The most popular algorithms which are based on CFAR (constant false alarm rate) method are very effective in detection of ships which are represented as very bright features on SAR images. However, ship detection from optical images requires a completely different approach. The main obstacle is heterogeneous vessel appearance on the image and thus making it very hard to detect using classical computer vision algorithms. Another topic of interest is not only detecting vessels, but also their classification. For many surveillance applications such as search and rescue operations, customs control, law enforcement and many others, the information about ship types detected on the image might be of high interest.

The recent advances in deep learning methods, especially (Deep) Convolutional Neural Networks (CNN) for image classification and object detection have achieved impressive results [8] [9] [10] [11] and surpassed all classical computer vision algorithms. These methods have been proven by many studies to be effective with satellite images as well [12] [13] [14] [15]. Deep learning opens new opportunities for vessel detection and classification as well. An overview of some related researches is given in the next subchapter.

The majority of existing works are rather experimental studies which are conducted on small and fixed image sizes and are limited in terms of vessel classification. The image size problem is related to the CNN architectures and hardware capacities. Supported image size by the most popular CNNs is typically between 300 and 1500 pixels rendering the longest side. In the computer vision world, scaling photographs from the camera of a smartphone would not lead to dramatic information loss, as the most interesting objects would still be clearly visible. In the RS context this situation becomes a real problem as the objects of interest are very small compared to the total image size. Regarding the classification problem, it is mainly caused by the lack of publicly available annotated datasets for the specific subject context, namely vessels or other maritime related objects. Furthermore, varieties of different satellite sensors, from a technical point of view, like spectral characteristics of imaging bands and their spatial resolution may limit the use of such datasets from project to project.

The developments presented in this thesis are an attempt to combine different technologies in order to provide an end-to-end automated solution for vessel detection from MR (medium resolution) and VHR (very high resolution) optical sensors for near-real time (NRT) applications. The developed solution is integrated at DLR's Ground Station Neustrelitz and already in operational use for CleanSeaNet [4] and Copernicus Maritime Surveillance Service [5].

Covered topics are image pre-processing techniques and vessel detection pipeline including data fusion with AIS and generated training datasets.

## 1.2 Related researches

Vessel Detection from optical satellite images is becoming frequently studied by a number of researches. First attempts were done years before the deep learning revolution in 2012 [8]. During that period, the most popular methods were classical computer vision object detection techniques. However, the CNN-based solutions outperformed them with a large margin in terms of speed as well as accuracy. Therefore, this review will be focused on the most relevant CNN-based methods applied on similar datasets as in this research.

Rainey, et al. [16] experimented with relatively small CNN architecture for ship type recognition. Their main intention was to find a suitable setting for such possible scenario like search for image locations containing particular vessel types. They have selected four classes (barge, cargo, container and tanker) and trained CNN-based binary classifiers one-vs-all. This approach allowed them to overcome the problem with unbalanced amount of training samples between the classes. The training and test datasets were generated out of WorldView-1 and WorldView-2 satellite images.  The results showed the potential of CNN for discriminating ship types earlier mentioned from other features on the image. However, authors pointed out that the experiment suffered from a relatively small amount of training samples. Partly, it could be solved by applying data augmentation as well as by using pre-trained CNN models on large dataset such as ImageNet [17].

Yamamoto and Kazama [18] presented CNN-based approach for extraction of ship-containing regions of WorldView-2 satellite images. They proposed to apply sliding window on entire satellite image and classifying image patches whether they contain vessel or not. Their focus was to find locations containing limited types of ships. They use simplified VGG [19] image classification model as a basis for that. Individual ship detection and type recognition topics are defined as a future research direction.

Yao, et al. [20] developed ship detection framework which used deep CNN to extract features and then region proposal network (RPN) to extract object bounding boxes. This concept is similar to Faster R-CNN [21], with the exception that CNN is not used as a multicategory classifier. Since the interest was in one generic object class “ship”, there is no need to apply an additional classifier. Authors report promising results in accuracy as well as performance. Nevertheless, with the proposed concept ship type recognition is not possible unless some post-classification step is included. The research was performed with the use of Google Earth imagery.

Nie, et al. [22] applied modified version of instance segmentation model Mask R-CNN [23] to detect vessels in harbor areas. They proposed to use Soft-Non-Maximum Suppression (Soft-NMS) [24] in order to improve the detection performance in harbor areas. This approach helped to reduce information loss in situations where the multiple objects (ships) may be located close to each other which results in high intersection of their bounding boxes. They could successfully extract vessel instances of two classes (merchant ships and battleships) in the dense harbor areas. The research dataset was based on a fixed sized image clips from Google Earth.

Štepec, et al. [25] presented a ship detection pipeline designed to work with MR satellite images from Copernicus Sentinel-2 and Planet Labs Dove satellites. For this task an adapted version Mask R-CNN [23] model was used. Authors utilized

AIS data to produce automatically annotated datasets from the above-mentioned satellite images, and then manually validated annotations. In addition, they resampled to lower resolution the Kaggle Airbus VHR ship detection dataset [26] and used it for training purposes. This showed an evidence of successful domain adaptation of training data from higher resolution to lower. Reported results showed promising performance in terms of detectability, however computational cost of the proposed approach is not mentioned. Furthermore, ship type recognition problem as well as parameter estimation are not addressed in this research.

## 1.3 Thesis structure

This thesis comprises six chapters. Chapter 1 gives an introduction and the research objectives of the thesis. It provides an overview of related researches and the thesis structure.

Chapter 2 provides an overview of the main components used for marine traffic monitoring tasks as part of complex maritime surveillance systems. Some background information as well as functional purposes of the most common systems are discussed.

Chapter 3 presents the developed vessel detection method. The proposed workflow includes preprocessing techniques for optical satellite images and data from Automatic Identification System (AIS) as well as actual deep learning-based vessel detection algorithm.

Chapter 4 presents the implementation of the proposed method. It describes the overall system architecture, processing chain and the hardware environment. Further, it covers the core software components developed by the author and generated training datasets.

Chapter 5 discusses the main outcome of this study. A short summary of all developments as well as performance evaluation are provided.

Finally, Chapter 6 concludes the thesis and discusses further research directions.

# 2 Components of Maritime Surveillance Systems

Maritime surveillance systems can be described as system of systems that combine many different sensor and information types in order to build the overall situational awareness at sea [3]. Depending on the domain they may include cooperative ship reporting systems as well as non-cooperative sensor systems. Fusion of information derived from both types of systems helps to reduce limitations and performance gaps of any particular system [3]. Figure 2.1 shows different sensor systems as components of an integrated maritime surveillance system. It combines remote sensing satellites, cooperative ship reporting systems as well as other maritime traffic monitoring systems. In particular, this thesis presents the technology for vessel detection from optical satellite sensors and data fusion with one of the cooperative ship reporting systems. Therefore, this chapter provides a short overview of the relevant system types.

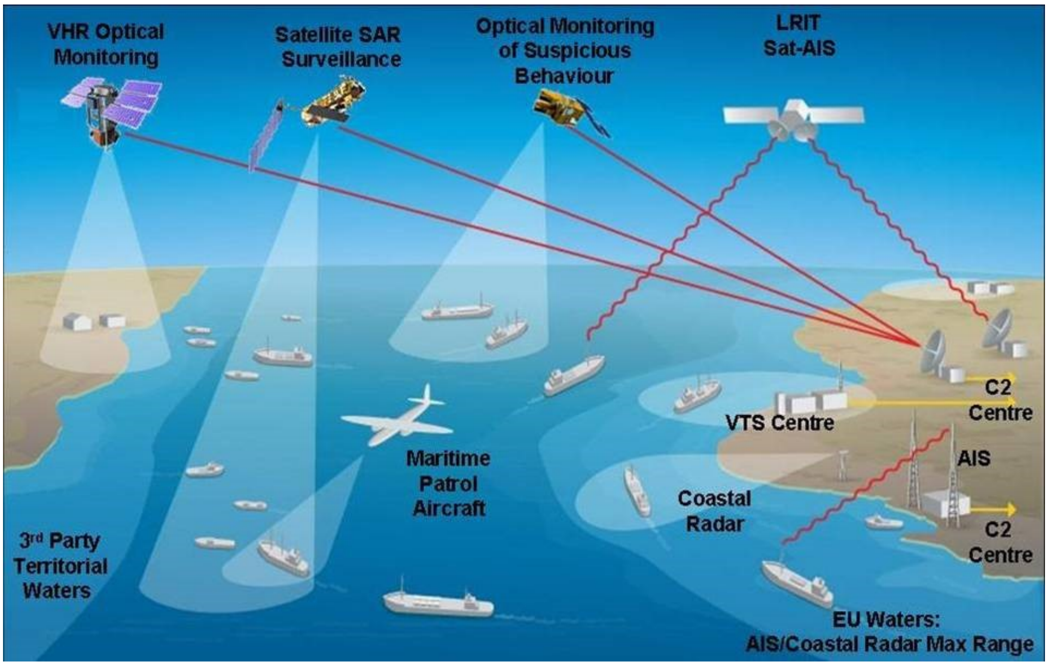

VHR Optical Monitoring
Satellite SAR Surveillance
Optical Monitoring of Suspicious Behaviour
LRIT Sat-AIS
C2 Centre
VTS Centre
AIS
Maritime Patrol Aircraft
Coastal Radar
C2 Centre
3rd Party Territorial Waters
EU Waters: AIS/Coastal Radar Max Range

**Figure 2.1: Components of Integrated Maritime Surveillance System.**
Image credit: ©ESA
https://www.esa.int/ESA_Multimedia/Images/2007/10/Integrated_maritime_surveillance

## 2.1 Automatic Identification System (AIS)

The Automatic Identification System (AIS) is an on-board position reporting system which was initially designed for collision avoidance. According to the International Maritime Organization (IMO) regulation [27], the carriage of AIS is mandatory for cargo vessels of 300 gross tonnages and upwards in international voyages and 500 gross tonnages and upwards in non-international voyages as well as all passenger vessels and tankers of any sizes in international voyages.

The AIS equipment includes very high frequency (VHF) transceiver, Global Navigation Satellite System (GNSS) receiver and display or terminal. In addition, it may also be connected with other on-board instruments like gyrocompass or rate of turn indicator. The device transmits and receives from other cooperative vessels information which contains geographical position, speed and course over ground, true heading and additional attributes about the vessel such as identification, vessel type and size, ports of call and destination. The geographical position and other movement attributes are retrieved from appropriate devices (GNSS receiver, gyrocompass, etc.); the Maritime Mobile Service Identity (MMSI) number of the vessel (unique identification) and vessel static parameters are hardcoded in the device; other attributes are entered manually, which is always a subject of their reliability. The transmission rate is related to the vessel's speed. For moving vessels, it ranges from 2 to 10 seconds and for anchored it is 3 minutes. The AIS broadcast coverage is limited to VHF range, namely line of sight.

Collection of AIS messages can also be carried by the coastal receivers. This opportunity makes the AIS an effective tool for real-time overview of the ship traffic in port areas. Coastal AIS reception is frequently integrated as part of Vessel Traffic Services (VTS) [3]. The coverage of the coastal receivers is about 40 nm but can be extended by installing the antennas on higher elevated positions.

There has been an active development in the satellite AIS technology over the past decade. The concept of this technology is based on the use of VHF receivers onboard of low-Earth orbit (LEO) satellite constellation. Collected information is then downlinked to the ground stations and distributed to the end users. This approach allows collecting AIS messages on wider and remote areas for use in vessel tracking tasks. The main limitations of these systems are related to the satellite revisit time which leads to observation time gaps.

Currently a large number of AIS data providers worldwide are offering access to the data collected from the network of terrestrial and satellite AIS receivers. Besides the real-time data many of them are offering historical datasets as well. The access to the AIS from providers is frequently organized via dedicated web-based services on a subscription basis.

## 2.2 Vessel Monitoring System (VMS)

The Vessel Monitoring System (VMS) is the type of position reporting systems for fishing vessels. Components and functionality of the VMS vary according to the nation of the vessel’s registry and the area where it is operating. Using VMS device unit, which is sometimes called “blue box” [3], fishing vessels are requested to send their identification, position, time, course and speed. These reports are transmitted with frequency defined by the authorities (from minutes to hours) or could be fetched in the polling mode. Usually the VMS unit is connected to the GNSS receiver and operates fully automatically. The satellite communication channel is used for data transmission.

## 2.3 Long Range Identification and Tracking (LRIT)

The Long-Range Identification and Tracking (LRIT) is another position reporting system designed for security and search and rescue purposes. The LRIT regulations are included in SOLAS Chapter V [28] which states its compulsoriness for vessels on international voyages of the following categories: passenger ships, mobile offshore drilling units and cargoes of 300 gross tonnage and upwards. Compared to AIS, the LRIT message contains much less information; it is limited to vessel’s identification, position and timestamp. Another difference to AIS is that it is not a broadcast system; LRIT messages are confidential and sent to dedicated recipients via satellite communication channel.

## 2.4 Vessel Traffic Services (VTS)

The Vessel Traffic Services (VTS) are shore-side integrated marine traffic monitoring systems. Their main tasks are surveillance and provision of safe marine trafficking in harbor and coastal areas with an increased risk. In terms of surveillance, the VTS centers are typically equipped with S-, X-, and K-band radar as well as closed-circuit television (CCTV) cameras. These non-cooperative sensors are used to provide real-time information for a very limited coverage. In addition, most of the modern VTS centers are equipped with AIS receivers or connected to AIS providers. The new technological trends are aimed at automated fusion of radar and AIS signals [3] into one integrated map-like visualization system.

## 2.5 Satellite Remote Sensing Data

Currently the satellite remote sensing data are getting more and more involved in maritime surveillance applications. Satellite imagery is an especially attractive source of information when there is a need to observe larger and/or remote areas. The most frequent use cases of satellite images in this domain are environmental and sea traffic monitoring.

From the technological perspective remote sensing systems can be divided into active and passive sensors [29]. The active sensors are emitting energy towards the objects and then receiving reflected signals. Measured time delay between the emission and return is used to characterize observing objects. The most popular representative of active sensors in satellite remote sensing field is the synthetic aperture radar (SAR). The passive sensors on the other hand are receiving energy which is different from sensor origin and reflected by the objects. Typically, this is sunlight energy or energy emitted by the object itself, such as thermal radiation. Optical satellite sensors are the examples of passive sensors.

Both types of sensors, SAR and optical, are providing unique capabilities for solving maritime related observation tasks. Over several decades Synthetic Aperture Radar (SAR) satellite data has been proven to be effective for object detection [6] [30] and environmental monitoring, such as oil spill detection, as well as sea state parameter estimation [31]. The main advantages of SAR sensors are their weather independence and capacity to cover very large areas. However, they are limited in terms of ship classification and identification without help of auxiliary datasets.

Optical satellite images, being closer to human-like perception in visible spectrum, as opposed to SAR, can provide more contextual information as well as details about the object (vessel) itself, such as texture, shape and color. Medium resolution (MR) multispectral optical sensors like the United States Geological Survey (USGS) Landsat-8 or Copernicus Sentinel-2 are beneficial for environmental monitoring tasks due to their spectral resolution. In addition, MR sensors are suitable for sea traffic monitoring as long as detecting targets have large enough sizes to appear on the image. This limitation is set by their spatial resolution, which is about 10-15 meters per pixel. For example, detectable targets can be large vessel types like container carriers or tankers. Another advantage of MR sensors is their spatial coverage, sometimes comparable to SAR missions. Very high resolution (VHR) optical satellite images are beneficial for sea traffic monitoring. The sub-meter spatial resolution of such images enables to detect and classify vessels of different types and sizes. Currently, rapid increase in the constellation of satellites with VHR optical sensors offers short revisit times for targeting areas. For example, the upcoming WorldView Legion (in 2021) constellation will offer more than 15 revisits per day. Figure 2.2 shows sample images from all mentioned satellite systems with overlapping coverages.

Authorities at national and supra national levels nowadays are more frequently involving remote sensing technologies in maritime surveillance systems. For example, in the EU the European Maritime Safety Agency (EMSA) is providing operational surveillance services CleanSeaNet [4] and Copernicus Maritime Surveillance Service [5] which are based on remote sensing data. Both types of sensors, SAR and optical, are used for solving different types of surveillance tasks as described above. Furthermore, developments presented in this thesis are already contributing to these services.

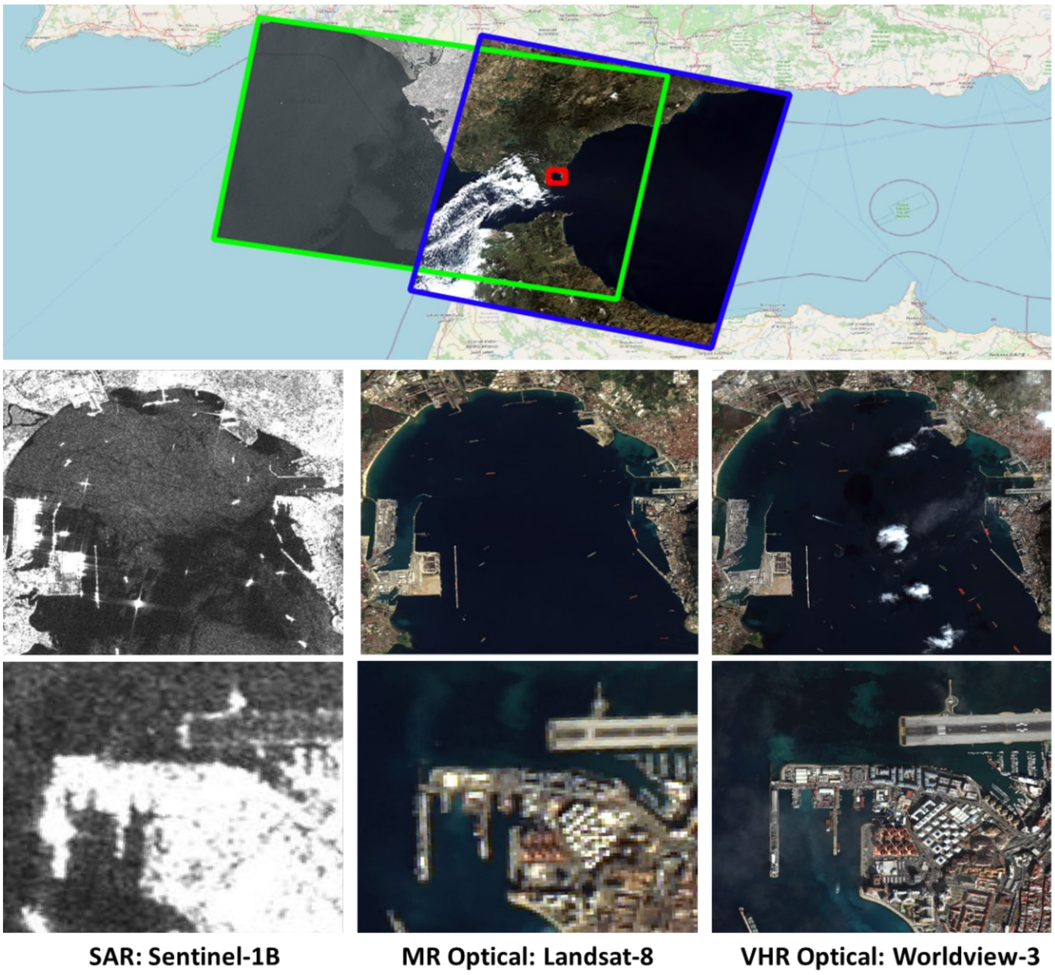


**Figure 2.2: Comparison of SAR and MR and VHR optical sensors.**
Colored polygons showing spatial coverages of different sensors: green – SAR (Sentinel 1); blue – optical MR (Landsat-8) and red – optical VHR (WorldView-3). VHR sensors provide more detailed picture, but they cover much smaller areas compared to MR sensors.
Image credit: ©OpenStreetMap; Copernicus Sentinel 1B © 2019 ESA; Landsat-8 © 2019 USGS; WorldView-3 © 2020 European Space Imaging / Maxar

# 3 EO-Based Vessel Detection Method

This chapter describes the developed methods for core functions of the vessel detection framework. It covers theoretical aspects as well as developed approach, whereas the software implementation is covered in the chapter 4. Algorithms and methods are described in the sequence of their occurrence within the processing workflow which is visualized in Figure 3.1. The subchapter 3.1 covers image preprocessing techniques which include fast atmospheric correction algorithm applied on level 1 (L1) satellite images and orthorectification. Further, the interpolation of position reporting system AIS is covered in the subchapter 3.2. The subchapter 3.3 covers algorithm for vessel detection from VHR and MR optical satellite sensors.

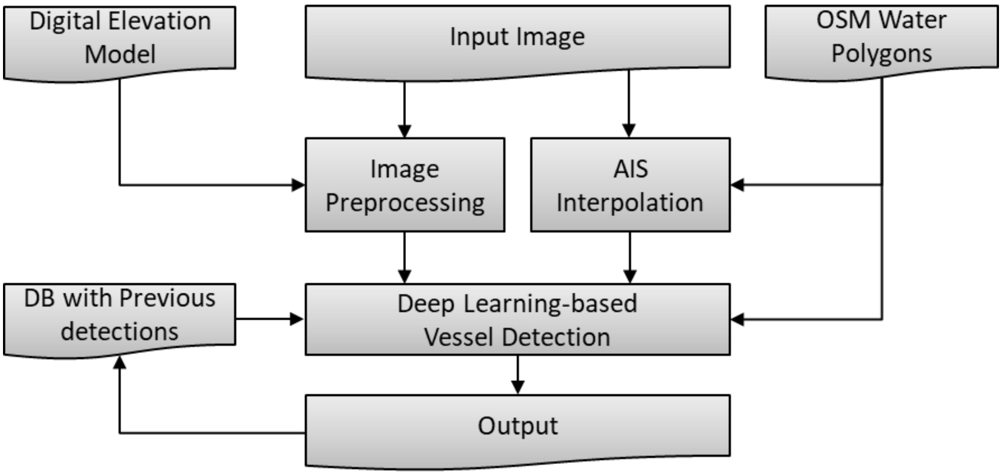


**Figure 3.1: Vessel detection framework core functions workflow.**

## 3.1 Image preprocessing

This subchapter presents two image preprocessing procedures applied before it is used in the detection process. The purpose of the first procedure is to reduce the atmospheric influence on object appearance in the image and it is called atmospheric correction. The second procedure is orthorectification which is needed to account heterogeneous terrain as well as the satellite movement and sensor-specific characteristics in order to increase position accuracy of detected vessels.

### 3.1.1 Image-based atmospheric correction

The amount of electromagnetic energy collected by the optical sensors, especially in the visible range, is greatly influenced by atmospheric conditions [29]. Even under cloud-free weather conditions, the atmosphere may produce different distorting effects such as scattering, absorbing or reflecting the light. The degree of these effects depends on specific factors such as the atmospheric composition, sun and sensor positions, as well as the angle between them at imaging time and location. The most dominant effect is the scattering which is caused by the atmospheric gases, aerosols and clouds.

Scattering means redirection of electromagnetic radiation by particles in the atmosphere. Particles and gas molecules with a smaller diameter size than the wavelength of radiation passing through the atmosphere exhibit Raleigh scattering. Shorter wavelengths (such as blue light) produce more apparent scattering. Higher solar zenith and satellite off-nadir (viewing zenith) angles cause a longer sun-target-sensor path of the radiation. This is another factor that leads to more intensive scattering. The Raleigh scattering can produce haze and distorted color appearance of the sensing targets. Mie scattering occurs when the wavelength is of similar size as the diameter of the atmospheric particles, visually appearing as haze in images. Water vapor, dust and smog are usually the main sources of Mie scattering. The non-selective scattering occurs when particles in the atmosphere are much larger than the radiation wavelength. For example, clouds and fog are the main reasons for non-selective scattering. While non-selective scattering is almost impossible to correct, the effects produced by the absorption as well as Mie and Rayleigh scattering can be minimized. This process is called atmospheric correction (AC).

All AC techniques can be grouped in two types: physics based and image based. Physics based methods are recommended when the physical reflectance of the targeting objects is of high importance, for example precise land cover classification or estimation of vegetation indices, time series analysis. Physics based methods are more accurate because they employ additional meteorological data at imaging time and location and may also consider surface geometry (elevation). The most popular methods are 6S [32] and ATCOR [33]. However, due to the fact that meteorological data usually becomes available after significant time delays they are not suitable for NRT applications. Image based AC methods, in contrast, do not require any additional information and are of low computational cost. The most popular image-based AC algorithm is dark object subtraction (DOS) [34].

The DOS algorithm assumes that the image contains regions which have almost zero reflectance, such as water bodies. This assumption perfectly agrees with the context of this project, where every image has significant areas covered by water. The non-dark appearance of the corresponding pixels is considered to be proportional to the atmospheric path radiance, and therefore can be used to compensate the additive effects of atmospheric scattering. The traditional DOS algorithm involves calculation of haze radiance ($\boldsymbol{L_{haze\ b}}$) which is then subtracted from the top of atmosphere (TOA) spectral radiance ($\boldsymbol{L_b}$) during the calculation of surface reflection (SR) product. This process can be expressed as:

$$\boldsymbol{SR_b = \frac{(L_b - L_{haze\ b}) \cdot d^2 \cdot \pi}{E_b \cdot cos\theta_s}} \qquad \textbf{(1)}$$

where $\boldsymbol{L_b}$ is the top-of-atmosphere (TOA) spectral radiance for the spectral band b (in units of W m$^{-2}$ sr$^{-1}$ µm$^{-1}$); $\boldsymbol{d}$ is the Earth-Sun distance in astronomical units; $\boldsymbol{E_b}$ is the band-averaged solar exoatmospheric irradiance (in units of W m$^{-2}$ µm$^{-1}$); $\boldsymbol{\theta_s}$ is the solar zenith angle; $\boldsymbol{L_{haze\ b}}$ is the top-of-atmosphere spectral radiance for the darkest objects on the image. Methods in determining how the $\boldsymbol{L_{haze\ b}}$ is estimated vary from one implementation to another [34] [35] [36]. Generally, it is always based on some empirical assumptions and may either be set as a static value for each band and sensor or computed dynamically on the basis of image properties.

In this project the DOS correction is applied on the TOA reflectance product and extended with histogram stretch to increase the image contrast. The algorithm consists of the following steps:

**Step 1:** Convert original image DN (digital numbers) into TOA spectral radiance. The conversion formula can be slightly different for every specific sensor. For the VHR satellite sensors of MAXAR family, which include WorldView-[1-3] and GeoEye-1 satellites, the conversion formula can be expressed as:

$$\boldsymbol{L_b = DN_b \cdot Gain_b \cdot \frac{absCalFactor_b}{effectiveBandwidth_b} + Offset_b} \qquad \textbf{(2)}$$

and for the MR satellite Landsat-8 as:

$$\boldsymbol{L_b = Gain_b \cdot DN_b + Offset_b} \qquad \textbf{(3)}$$

where $\boldsymbol{L_b}$ is the TOA spectral radiance for a given band $\boldsymbol{b}$ (in units of Wμm$^{-1}$ m$^{-2}$ sr$^{-1}$); $\boldsymbol{DN_b}$ - digital number – is the pixel value of original L1 satellite image; $\boldsymbol{Gain_b}$ and $\boldsymbol{Offset_b}$ are the absolute radiometric calibration adjustment factors that are sensor and band specific; $\boldsymbol{absCalFactor_b}$ is the absolute radiometric calibration factor and $\boldsymbol{effectiveBandwidth_b}$ is the effective bandwidth of the spectral band. $\boldsymbol{Gain_b}$ and $\boldsymbol{Offset_b}$ are either available in the image metadata or published by the vendor separately. The values $\boldsymbol{absCalFactor_b}$ and $\boldsymbol{effectiveBandwidth_b}$ are available in the image metadata.

**Step 2**: Calculate the TOA reflectance:

$$\boldsymbol{p(TOA)_b = \frac{L_b \cdot d^2 \cdot \pi}{E_b \cdot cos\theta_s}} \qquad \textbf{(4)}$$

where $\boldsymbol{L_b}$ is the TOA spectral radiance for the spectral band b derived in (1); $\boldsymbol{d}$ is the Earth-Sun distance in astronomical units; $\boldsymbol{E_b}$ is the band-averaged solar exoatmospheric irradiance (in units of W m$^{-2}$ μm$^{-1}$); $\boldsymbol{\theta_s}$ is the solar zenith angle.

It is worth to note, that steps 1 and 2 are presented here in order to simplify readability, in the software implementation (processor ImageHandler, chapter 4.2.1) they are merged into a single processing step. Furthermore, the TOA reflectance is calculated only once for every unique DN in the histogram and stored as look-up table.

**Step 3**: Determine the haze reflection $\boldsymbol{p(TOA)_{haze\,b}}$ which is the maximum reflectance $\boldsymbol{p(TOA)_b}$ of the 0.001 percentile of the image pixels which have the **lowest** reflectance values. It must be calculated for every band separately.

**Step 4**: Determine the maximum reflection threshold $\boldsymbol{p(TOA)_{maxTh\,b}}$ in order to increase the image contrast. It is the **minimum** reflectance $\boldsymbol{p(TOA)_b}$ of the 0.01 percentile of the image pixels which have the **highest** reflectance values. It must be calculated for every band separately.

The percentile values for steps 3 and 4 were determined empirically after examining dozens of MR and VHR images.

**Step 5**: The SR product is calculated as following:

$$\boldsymbol{SR_b} = \mathbf{min}(\,\boldsymbol{p(TOA)_{maxTh\,b}},\ \boldsymbol{p(TOA)_b} - \boldsymbol{p(TOA)_{haze\,b}}) \qquad \textbf{(5)}$$

The resulting $\boldsymbol{SR_b}$ is clamped to fit into the value range of 0 and $\boldsymbol{p(TOA)_{maxTh\,b}}$ and remapped to an 8-bit unsigned integer raster image.

Figure 3.2 shows some examples of images before and after the application of proposed AC. One of the most important effects achieved is higher contrast between the vessels and surrounding background (mostly water). Furthermore, the haze effect is minimized which exposes more objects on the surface.

The described AC method cannot compensate for atmospheric absorption. It is designed for production of visualization-friendly products without the use of ancillary datasets within NRT applications. It should not be considered for land cover classification and time series analysis. For those tasks physical-based AC methods are recommended. However, resulting images are very suitable for object detection and visualization tasks.

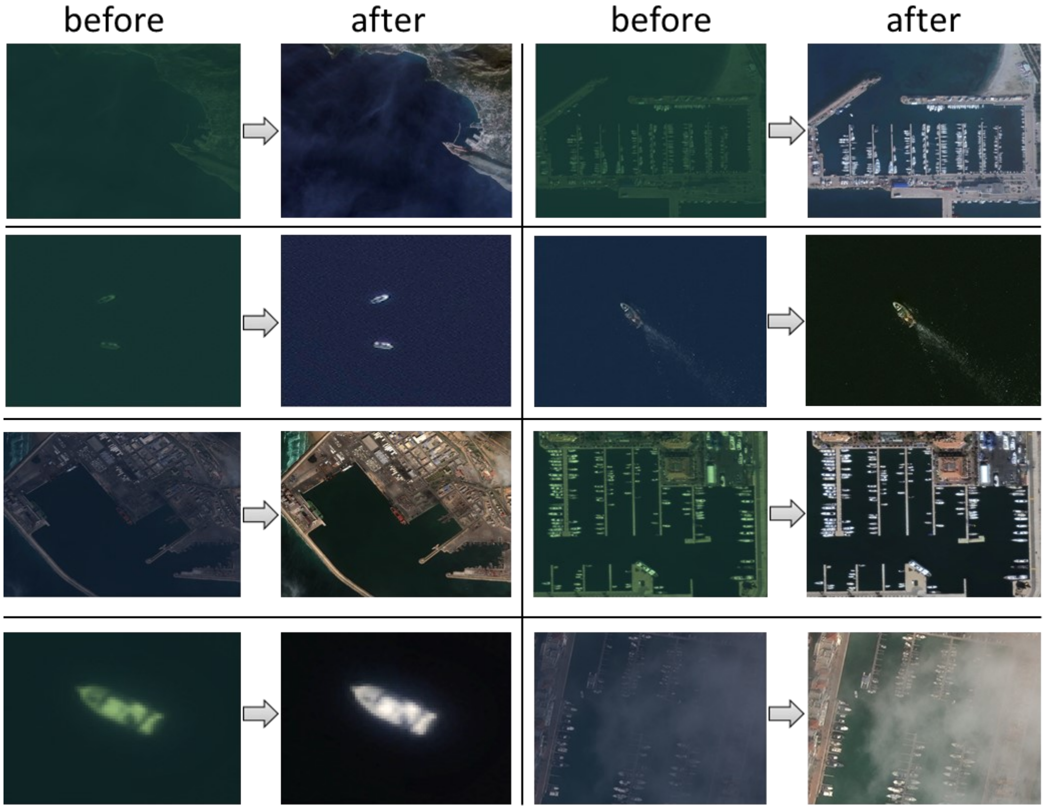


**Figure 3.2: Examples of atmospheric correction algorithm results.**
This figure shows the differences between corrected and uncorrected VHR images taken at different locations and within different atmospheric conditions.
Image credit: WorldView-3/GeoEye-1 © 2020 European Space Imaging / Maxar

### 3.1.2 Image orthorectification

Locations of objects shown in the VHR optical satellite images are highly affected by two factors: terrain relief and the tilt of the sensor [29]. Orthorectification aims at removing or reducing the spatial distortions on the image which occur due to the deviations in terrain relief and viewing angles [37]. The result of this operation is a planimetric map-like image (also called as orthoimage or orthophoto) with precise positions for the objects and consistent scaling throughout the image.

The common orthorectification approach involves using a mathematical model to describe physical relationship between 2D image space and 3D ground relief.

The RPC (rational polynomial coefficients) model enables transforming the image coordinates (row and column) into the Earth's surface coordinates [37]. It is computed from the satellite's position in orbit, its sensor orientation and on the basis of the physical sensor model. Most modern VHR optical images, like those from MAXAR's WorldVIew-[1-3] and GeoEye-1 satellites (and many others) are supplied with an RPC model.

Combining RPC model data with precise terrain elevation information such as DEM (Digital Elevation Model) significantly minimizes the spatial distortions on the image. Orthorectification operation is crucial to retrieving the precise positions of the objects, especially in the port areas. Depending on the terrain relief type and the image off-nadir view, the position error in the VHR image without orthorectification may reach up to 100 meters. In the orthorectified image this error may be reduced to a few meters or less, depending on the imaging angles, RPC quality and the DEM applied. Figure 3.3 shows one example of an image before and after the application of RPC based orthorectification.

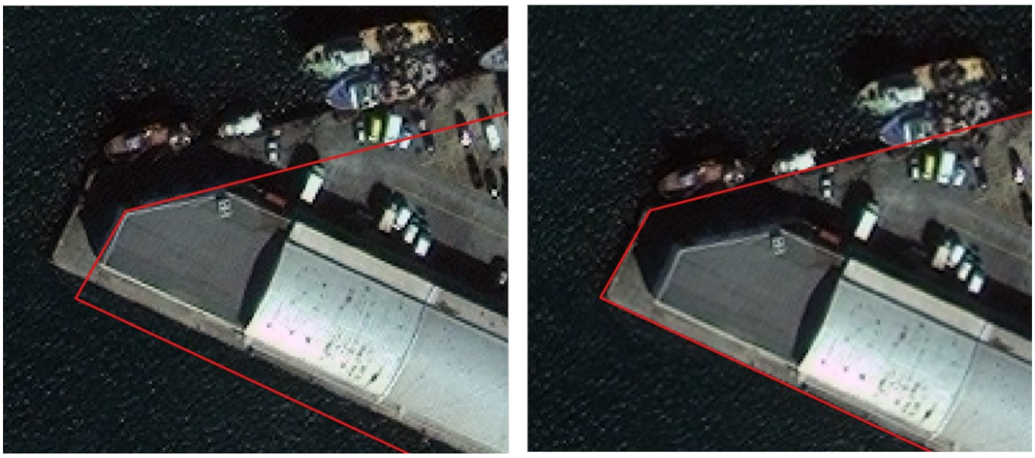


**Figure 3.3 Example of orthorectification results**
This figure shows the differences before and after image orthorectification. The overlaid red line is the land mask derived from the OSM dataset. In this particular example the position difference is approximately 10 meters. This visualization is based on the GeoEye-1 satellite image © 2020 European Space Imaging / Maxar

In the open seas, orthorectification is equally important to the harbor or land areas. Instead of DEM a constant averaged elevation on open water is used. This method still allows reducing geometrical distortions caused by the sensor itself, which is especially useful for image mosaicking in case of multi-strip acquisitions.

## 3.2 AIS interpolation

In order to improve detection results as well as to possibly identify detected vessels, it is possible to involve ancillary data acquired from position reporting systems. Current implementation utilizes AIS data for that purpose, but the described method can be applicable to other position reporting systems as well.

Within the vessel detection framework described in this thesis the L1 image is usually available within 10-20 minutes after the downlink. Because of this delay, it is possible to collect AIS data not only for the time before the image acquisition, but also up to 20 minutes afterwards. This allows reconstruction of vessel tracks in order to get an AIS ship position of higher accuracy according to the imagery time. The implemented solution enables precise AIS to image data fusion which is based on the exact imagery time for every ship and accounts for the following problems:

1) Low AIS update rates. The AIS reports for the moving vessels are being broadcasted in different reporting intervals. The transmission of updates depends on the SOG of the vessel. Furthermore, updates occur at different intervals depending on whether a Class A or Class B transponder is used. Unfortunately, complete transmitted reports are not always collected due to various technical reasons. Furthermore, AIS providers often aggregate the data by time. As a result, in some cases, it could take a few minutes for available AIS reports to be updated.

2) Timestamp deviations within the image. Depending on the satellite sensor type and covered spatial extent different image locations may have big deviations in collection timestamps. With VHR sensors these deviations are not significant and typically are about a couple of seconds or less. However, with MR sensors, such as Landsat-8, time deviations may be up to 30 seconds, which could be a problem with moving objects. For example, a cargo vessel with the speed of 20 knots (approx. 10m/sec) would have covered a distance of 300 meters on the open sea. Therefore, for prediction of the AIS position at imagery time, a specific timestamp at any position in the image should be considered.

Initially the AIS track is filled with interpolated points so that the minimum time gap between positions will not exceed 10 seconds. The new calculated positions are derived using "dead reckoning" concept, based on the closest known positions from both sides (when available), and accounting their speed and course over ground as illustrated in Figure 3.4.

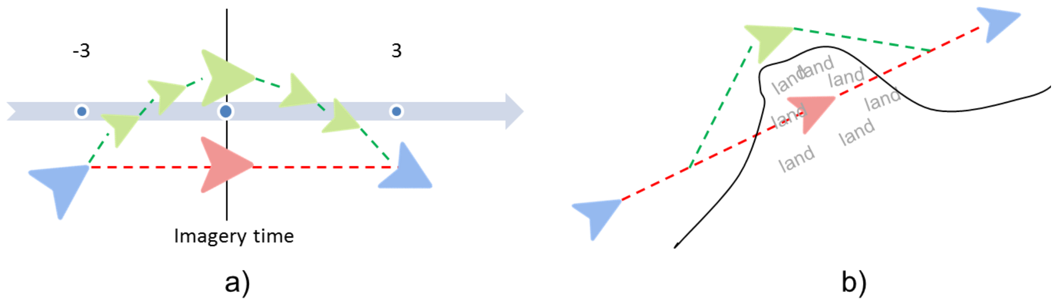


**Figure 3.4: AIS track reconstruction.**

If the new position intersects the coast line it will be iteratively corrected until it meets the following two conditions: 1) it is located on water and 2) it is located in the minimum possible distance between known positions under valid condition 1 (see Figure 3.4-b). The water polygons from OpenStreetMap (OSM) project [38] are used in this project.

The new attribute values for speed and course over ground are calculated by linear interpolation from known surrounding points, which can be expressed as:

$$\boldsymbol{a_i = a_{i-1}w_{i-1} + a_{i+1}w_{i+1}} \qquad \textbf{(6)}$$

where $\boldsymbol{a_i}$ is the attribute value for a new point to be calculated, $\boldsymbol{a_{i-1}}$ and $\boldsymbol{a_{i+1}}$ are the attribute values from neighboring points, $\boldsymbol{w_{i-1}}$ and $\boldsymbol{w_{i+1}}$ are corresponding weights for the known attributes and are based on their distance from predicted point:

$$\boldsymbol{w_{i-1} = 1 - \frac{d_{i-1}}{d_{i-1}+d_{i+1}}} \text{ and } \boldsymbol{w_{i+1} = 1 - \frac{d_{i+1}}{d_{i-1}+d_{i+1}}} \qquad \textbf{(7)}$$

where $\boldsymbol{d_{i-1}}$ is the distance from the "left" known point to predicted, and $\boldsymbol{d_{i+1}}$is the distance from the "right" known point to predicted.

If the predicted point is outside of the known track, meaning it has only one known neighbor, then the last known attribute is taken as the new point.

AIS positions at imagery time are derived in the following way. The reference time for the whole image must be defined. The optimal choice is to take a middle timestamp between the start and stop image collection timestamps. Then, it is necessary to find the closest AIS reports in time relative to the reference time. In the next step, AIS geographical coordinates that are reported closest to the reference time are used to determine on which part of the image they would appear. Corresponding image pixel coordinates are used to calculate the exact timestamp for this location. Finally, the new AIS report for the extracted timestamp is estimated in accordance to the procedure described above.

## 3.3 Vessel detection from VHR and MR optical satellite images

In this subchapter the two-stage vessel detection algorithm is presented. The first stage is responsible for fast preselection of vessel containing image parts, whereas the second stage is the actual vessel detection and parameter estimation. Both detection stages are based on the use of convolutional neural networks (CNN), therefore a short introduction to CNNs is given. Afterwards the detection stages in the sequence of their occurrence in the algorithm are described.

### 3.3.1 Introduction to Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) is a class of deep neural networks mostly applied for the image content analysis in the computer vision domain. In particular it is very effective for image classification and object detection tasks. The goal of image classification is assigning one (or more) label(s) to the entire image to describe its content. This is different from the commonly used definition in the remote sensing community, where the image classification means labelling every single pixel in the image. For example, a satellite image may be classified in a computer vision way with a single label "vessel" indicating that there is actually a vessel on this image. CNN based image classification assumes prediction of the image label by the neural network model which is trained in advance with labelled data of some categories. Object detection is a more complex process which extends the image classification problem: its objective is to detect the presence of different objects in the image, to find their locations and to predict their labels.

The process by which CNN works is inspired by the organization of the visual cortex of animals [39] [40] [41], whose individual neurons respond on receptive fields of restricted regions which form the entire field of vision. Similarly to that, the general idea of CNN is to split an image into small subsets (regions) and then, with the help of series of convolutions, to extract the low-level features such as lines, edges, and colors. At the later steps, low-level features are forming the high-level features such as car wheels or zebra lines and are then used to identify the object. The CNN's typical architecture contains three types of building blocks: convolution layer, pooling layer and fully connected layers. Figure 3.5 is showing a very simple CNN architecture containing sequences of convolutional, pooling and fully connected layers.

The process by which the input data (image) passes through the CNN in the forward direction is called inference or forward propagation. During this process each layer in the network produces the inputs for the successive layer. The loss function is used to evaluate the CNN performance and to adjust the parameters which learn the CNN - learnable parameters (also referred as weights). This process is called back propagation. The training process of the CNN consists of a large number of full cycles of the forward propagation followed by the back propagation aiming at minimizing the difference between the network outputs and the ground truth annotations. The parameters which are set manually and define the network configuration are called hyperparameters.

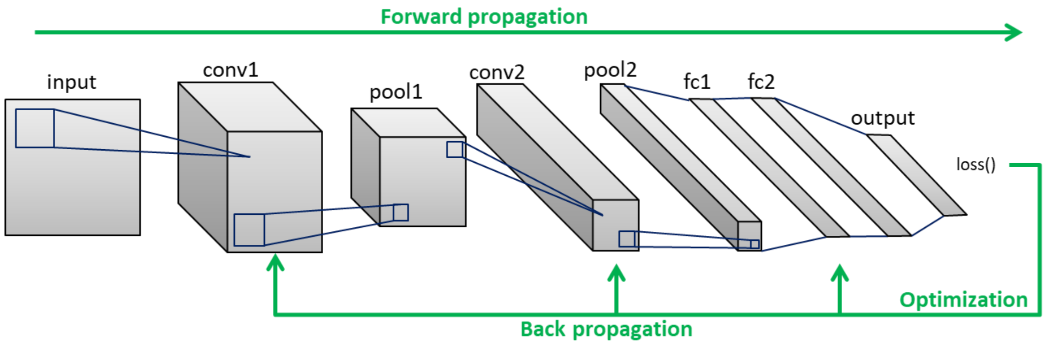
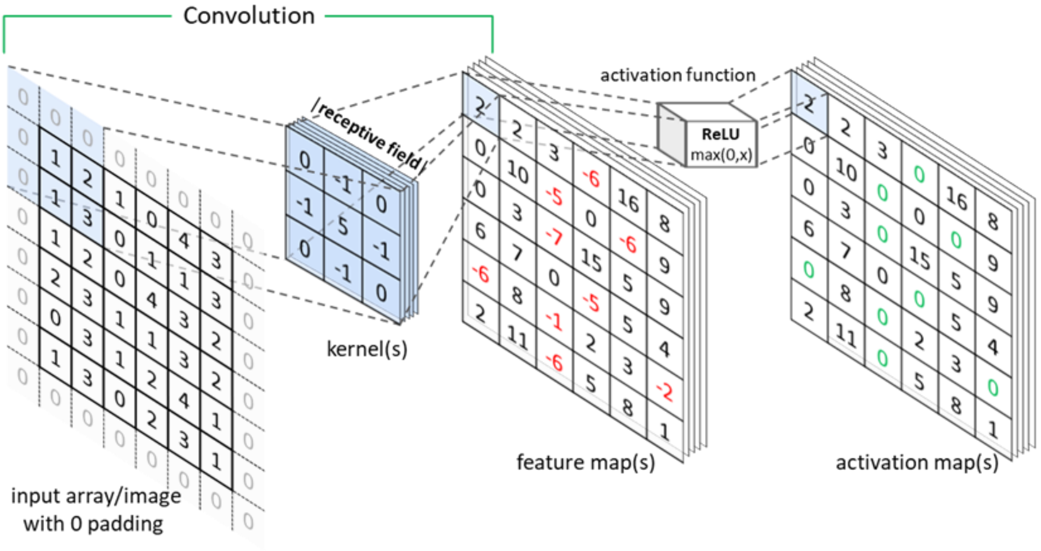


**Figure 3.5: An example of simple CNN architecture.**

The convolutional layer (conv) is the main element in the CNNs and it is responsible for the feature extraction. Typically, a convolutional layer combines two mathematical operations: linear convolution and activation function. This process is visualized in Figure 3.6.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 2 | 1 | 0 | 4 | 3 | 0 |
| 0 | 1 | 3 | 0 | 1 | 1 | 3 | 0 |
| 0 | 1 | 2 | 0 | 4 | 3 | 2 | 0 |
| 0 | 2 | 3 | 1 | 1 | 3 | 2 | 0 |
| 0 | 0 | 3 | 1 | 2 | 4 | 1 | 0 |
| 0 | 1 | 3 | 0 | 2 | 3 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| 2 | 2 | 3 | -6 | 16 | 8 |
| 0 | 10 | -5 | 0 | -6 | 9 |
| 0 | 3 | -7 | 15 | 5 | 9 |
| 6 | 7 | 0 | -5 | 5 | 4 |
| -6 | 8 | -1 | 2 | 3 | -2 |
| 2 | 11 | -6 | 5 | 8 | 1 |

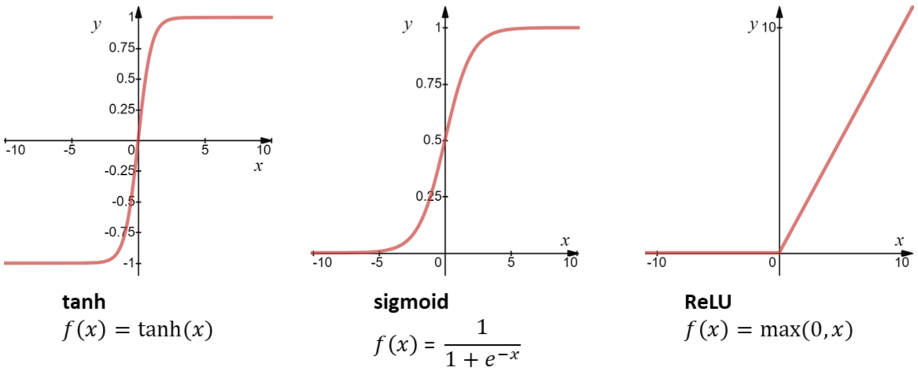
| | | | | | |
|---|---|---|---|---|---|
| 2 | 2 | 3 | 0 | 16 | 8 |
| 0 | 10 | 0 | 0 | 0 | 9 |
| 0 | 3 | 0 | 15 | 5 | 9 |
| 6 | 7 | 0 | 0 | 5 | 4 |
| 0 | 8 | 0 | 2 | 3 | 0 |
| 2 | 11 | 0 | 5 | 8 | 1 |

**Figure 3.6: Convolution layer.**

Convolution is the linear operation used for feature extraction. It involves the set of learnable kernels to determine the presence of different patterns or features in the input array. The kernel is usually expressed as a small square matrix with the same depth as the input array. The width and height of the kernel can be treated as the receptive field. Each element within the receptive field in the input array is multiplied by the corresponding element in the kernel and then summed up to obtain the value at the corresponding location in the output feature map. The kernel is sliding across the input array until the complete feature map is generated.

The sliding step of the kernel is called a stride. The stride, the receptive field size and the number of kernels are the hyperparameters, whereas the coefficients in the kernels are the learnable parameters. In order to allow the center of the kernel to overlap the outermost elements in the input array a padding technique is used. This ensures the same dimension of the feature map as in the input array. The most popular padding technique in modern CNNs is zero padding, which is visualized in Figure 3.6. Basically, it pads additional rows and columns filled with zero values to all four sides of the input array.

An activation function is used to add non-linearity to the network. The most popular functions are sigmoid, hyperbolic tangent (tanh) and rectified linear unit (ReLU). Due to its simplicity and computational efficiency the ReLU is the most popular activation function used today [8]. The main advantage of ReLU is that it truncates all negative values which reduce the number of activated neurons. This leads to dramatic performance increase and it is several times faster than tanh and sigmoid.



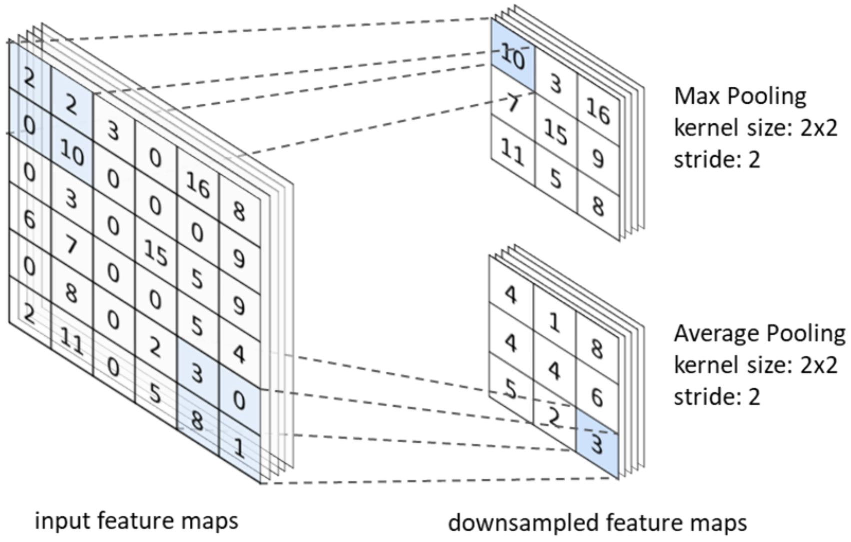
**tanh**
$f(x) = \tanh(x)$

**sigmoid**
$f(x) = \frac{1}{1 + e^{-x}}$

**ReLU**
$f(x) = \max(0, x)$

**Figure 3.7: Activation functions.**

The Pooling layer (pool) is frequently used between the convolutional layers to reduce the dimensionality of the activated feature maps in order to decrease the number of learnable parameters in the network. Additional effects of this operation are noise reduction and feature position invariance in the output layer. The pooling layers have no learnable parameters, whereas their kernel size, stride and padding are hyperparameters which are set by the network architect. The two most popular pooling types Max Pool and Average Pool are visualized in Figure 3.8.



|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 2 | 2 | 3 | 0 | 16 | 8 |
| 0 | 10 | 0 | 0 | 0 | 9 |
| 0 | 3 | 0 | 15 | 5 | 9 |
| 6 | 7 | 0 | 0 | 5 | 4 |
| 0 | 8 | 0 | 2 | 3 | 0 |
| 2 | 11 | 0 | 5 | 8 | 1 |

|  |  |  |
|---|---|---|
| 10 | 3 | 16 |
| 7 | 15 | 9 |
| 11 | 5 | 8 |

|  |  |  |
|---|---|---|
| 4 | 1 | 8 |
| 4 | 4 | 6 |
| 5 | 2 | 3 |

**Figure 3.8: Pooling layer.**

The pooling operation results in a downsampled feature map where each element corresponds to the maximum or averaged (depending on selected method) value within the kernel at the appropriate location in the input feature map. The most frequent configuration for pooling is the kernel size of 2x2 and the stride of 2, which produces the downsampled feature maps by a factor of 2.

The fully connected layers are responsible for classification of extracted features in convolution layers and therefore are usually placed at the end of the network. Derived feature maps from the series of convolution and pooling layers are transformed into one-dimensional vector. Elements of this vector are then connected to each neuron in the fully connected layer. The strength of these connections between the inputs and the outputs is dependent on the learnable parameters, which are adjusted during the training process. The network architecture may contain sequence of several fully connected layers as shown in Figure 3.9.

All fully connected layers are usually followed by the activation function, for example ReLU. Functionally, the fully-connected layers are attempting to describe non-linear connections between the detected features. The activation function applied in the last layer is usually different from the ones used in the previous layers. For multiclass classification tasks, the most suitable function is Softmax, which outputs the probability distributions between the classes. The Softmax function can be expressed as:

$$\boldsymbol{\sigma(z_j) = \frac{e^{z_j}}{\sum_{c=1}^{M} e^{z_c}}} \tag{8}$$

where $\boldsymbol{z}$ is the inputs to the output layer; $\boldsymbol{M}$ is the total number of output classes; $\boldsymbol{j}$ and $\boldsymbol{c}$ are the class indexes.

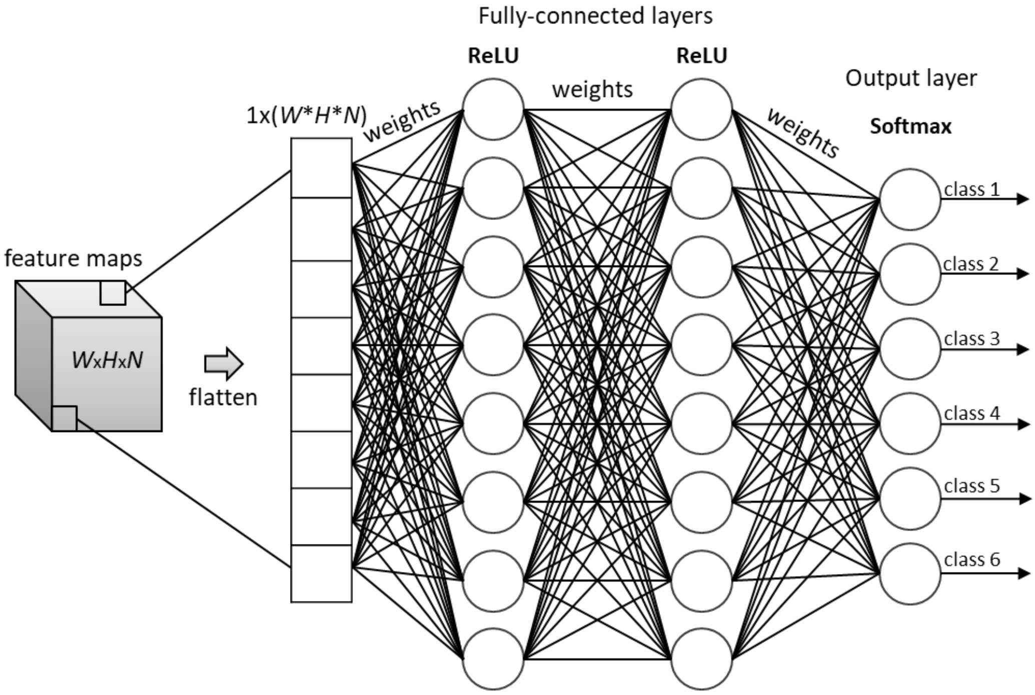


**Figure 3.9: Fully-connected layers.**

The loss function (also referred as cost function) is used to evaluate the deviations between the predicted output by the network and the expected (ground truth) value. The value derived by the loss function is used by the optimization algorithm in the back propagation process to adjust the learnable parameters. The common choice of the loss function for multiclass classification is the cross entropy, which can be expressed as:

$$\boldsymbol{loss(x) = -\sum_{c=1}^{M} log\ (y_{c,p}) y'_{c,p}} \tag{9}$$

where $\boldsymbol{x}$ is the input image (in case of image classification CNN); $\boldsymbol{M}$ is the number of classes; $\boldsymbol{c}$ is the class index; $\boldsymbol{y_{c,p}}$ is predicted probability (score) for class $\boldsymbol{c}$; and $\boldsymbol{y'_{c,p}}$ is the true probability for the class $\boldsymbol{c}$ (0 or 1).

The optimization algorithm gradient descent [42] [43] [44] is used in backpropagation for updating the learnable parameters in the CNNs towards the minimization of the loss function. The calculated gradient of the loss function defines the direction for iterative update of the learnable parameters with the step size according to the predefined hyperparameter learning rate. This process can be expressed as:

$$\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{e}\frac{\partial \boldsymbol{loss}(\boldsymbol{x})}{\partial \boldsymbol{w}} \qquad \textbf{(10)}$$

where $\boldsymbol{w}$ is the learnable parameter (weight); $\boldsymbol{e}$ is the learning rate; $\boldsymbol{loss}(\boldsymbol{x})$ is the loss from (9).

It is worth to mention, that all learnable parameters in the network are initially randomly set. The initial value for parameter *learning rate* is usually selected empirically for the specific CNN architecture and the dataset. Typically, it is set in the range between 0.0 and 1.0. Setting this parameter too small may require more training time, whereas setting it too large may result in not finding the optimal values for learnable parameters. In some applications the learning rate remains the same during the entire training process (static), whereas in other applications it is possible to find scheduled as well as adaptive modification of the learning rate.

The hyperparameter *batch size* of the gradient descent defines the number of training samples to process through the network before learnable parameters are updated. If the batch size equals to the number of training samples the algorithm is called batch gradient descend [44]**.** This type of optimization algorithm is required to process the entire training dataset before every update of the learnable parameters. This scenario can be very slow and may require enormous hardware resources. Therefore, the more optimized stochastic gradient descent (SGD) (batch size = 1) or mini-batch gradient descend (batch size > 1) and number of their improved versions such as Momentum, Adam, Adagard and others are frequently used [44]. Many of the optimization algorithms from the gradient descent family are available by default in all modern machine learning software frameworks, such as TensorFlow [45]. The parameter controlling which optimization algorithm to use is also one of the hyperparameters.

The hyperparameter *number of epochs* controls how many times the training algorithm will work through the entire training dataset. Alternatively, the amount of training cycles may be controlled by the hyperparameter *number of training steps* which corresponds to the number of gradient updates. One epoch is comprised of (number of training samples) / (batch size) training steps.

The decision on how each of the hyperparameters is set is usually based on empirical findings taking into consideration such factors as CNN overall architecture, quality and amount of training data samples as well as hardware resources available.

In the situations when the amount of training data samples is not sufficient a transfer learning technique can be used. In the image classification CNN world, transfer learning assumes training the CNN on the large training dataset (like ImageNet [19]), and retraining the last layers of the CNN on the new dataset.

The described CNN architecture (shown in Figure 3.5) in this chapter contains seven layers only. Its purpose is to demonstrate the main components of CNNs and not to solve any real problem. In modern image classification and object detection applications much deeper CNN architectures are used.

Although the general concept of CNN was proposed back in the late 1980s, the hardware resources were the biggest limitation factor in its application. Since the mid-2000s, there has been increased interest in the use of CNNs for visual recognition problems. The tipping point in the use of CNNs was the introduction of the image classification network AlexNet in 2012 [8]. Since then a number of new deeper and more accurate networks have been introduced such as Inception [9] [46]. The extended models for object detection task were invented afterwards. In particular the region-based convolutional neural networks (RCNN), such as very popular Fast(er)-RCNN [21] which can predict bounding boxes of the objects, and later Mask-RCNN [23] which can also label the pixels belonging to the detected objects.

### 3.3.2 Global region search

The Global Region Search is the first stage of the two-stage vessel detection algorithm. The goal of this step is to detect potential regions with vessel presence prior to the object detection, thus to reducing the computation cost of the entire processing chain.

Frequently, the observation areas in the maritime domain are covering ports and coastal areas. In that scenario a significant part of the image may be occupied by land. For these kinds of situations, the initial step of the global region search is land discrimination. With help of the geographically annotated ancillary data, the land areas are excluded from the detection. The water polygons available from the OSM project [38] are used for this purpose. The OSM dataset provides a very good level of detail for both ports and rural coastal areas. To avoid any information loss along the coastline, a buffer of 50 meters in the direction to the land is considered in the case of VHR scenarios. For the MR scenarios, due to the lower image resolution, the coastal areas are excluded from detection by applying a buffer of 50 meters in the opposite direction from land. The resulting vector layer is rasterized with the same pixel size and spatial extent as the input image. The final land masking raster layer is used to exclude land areas from the detection process. The land masking process is shown in Figure 3.10-a.

After the land areas are excluded, the remaining image parts are divided into small tiles of size 224x224 pixels with stride of 180 pixels. Then, every tile is classified by the image classification CNN whether it contains vessel/vessel parts or not (binary classification). In some way this approach is following the idea presented in the R-CNNs [47], where the features extracted from the image classification CNN are used for binary tests of the predefined set of regions indicating the presence of an object. The tile size is chosen on the basis of the selected CNN architecture which is described below.

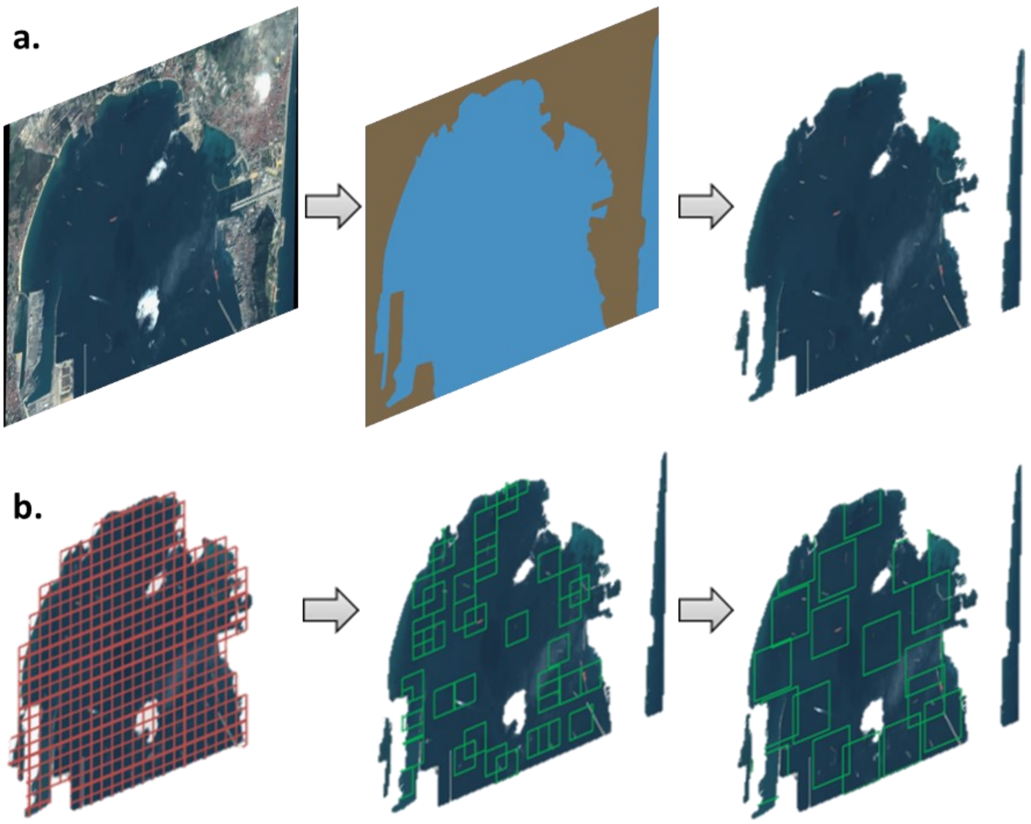


**Figure 3.10: Global region search workflow.**
a. Land masking with OSM water polygons
b. Extracting region proposals with image classification CNN
This visualization is based on the WorldView-3 satellite image © 2020 European Space Imaging / Maxar

Because of the NRT-oriented nature of this project, the processing timeline is very crucial. This was a motivating factor to search for a CNN architecture which is proven to be efficient under computational constraints. The choice was made in favor of the image classification CNN MobileNet [10]. The MobileNet is a lightweight CNN which is designed to work under computationally limited conditions, for example to run on mobile devices. Instead of standard convolutions it applies the depthwise separable convolutions [48]. As described in the introduction to CNN section of this thesis a standard convolution applies a set of filters (kernels) on every input channel to generate a set of output channels in one layer. In depthwise separable convolutions this operation is divided into two layers: depthwise convolutions (filtering) and pointwise convolutions which combine the results into the final output layer. For both layers batch normalization [49] and ReLU activation function are used. Furthermore, in MobileNet only one filter per channel is applied. Although this approach may potentially lead to accuracy degradation, it allows for a reduction in the number of parameters in the network which consequently leads to a reduction in the total number of floating-point multiplications.

The accuracy assessment provided in [10] demonstrates the reasonable performance for the image classification task (up to 70.7% on ImageNet dataset) while at the same time being up to 30x computationally cheaper as compared to the full-size models, such as Inception (for example v3 showed accuracy of 78.1% on ImageNet dataset).

The regions which are classified by MobileNet as “vessel” class are merged into the clusters. Every cluster is formed by the set of overlapping and neighboring regions. The cluster sizes are set to 1500x1500 pixels for VHR images and 600x600 pixels for MR images. The cluster origin is the top-left corner with the first detected region.

Other positively classified regions which are overlapping the cluster with an intersection area of more than 70% from their own area are included in this cluster. Regions with intersection of less than 70% or completely outside are becoming members of the new cluster. This process is visualized in Figure 3.11.

Besides the MobileNet classification results, additional region proposals extracted from the two ancillary data sources: 1) AIS signals acquired or interpolated at imaging time and location and 2) previous vessel detection results for imaging location if any. If any of the points extracted from these two sources do not intersect with MobileNet classified region proposals, they are used to create additional region proposals whose sizes are dependent on the vessel sizes reported by the AIS or estimated by the previous detection at current location. These region proposals contribute to cluster generation as well.

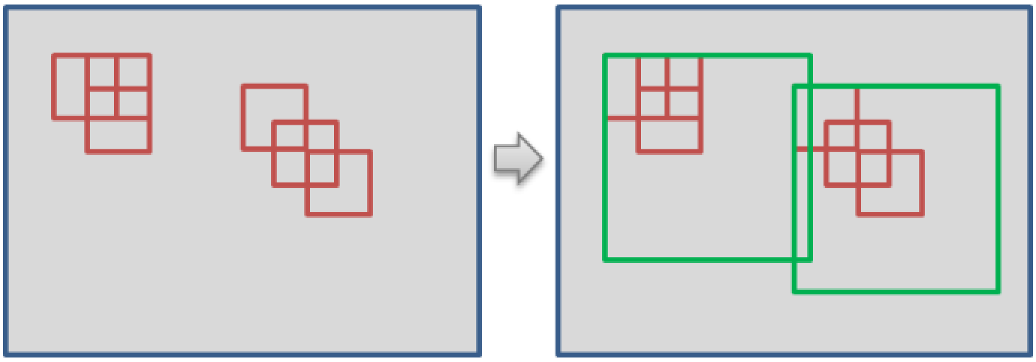


**Figure 3.11: Clustering of preselected regions.**
Preselected regions by MobileNet are shown in red and the resulting clusters are in green.

### 3.3.3 Vessel detection and parameter estimation

The object detection network Faster R-CNN [21] is applied on the regions extracted in the global region search step. The Faster R-CNN can be described as architecture with two modules: the region proposal network (RPN) and the Fast R-CNN [47]. For computational efficiency, both modules share the same feature maps extracted only once by the backbone image classification CNN. The overall architecture of the Faster R-CNN is illustrated in Figure 3.12.

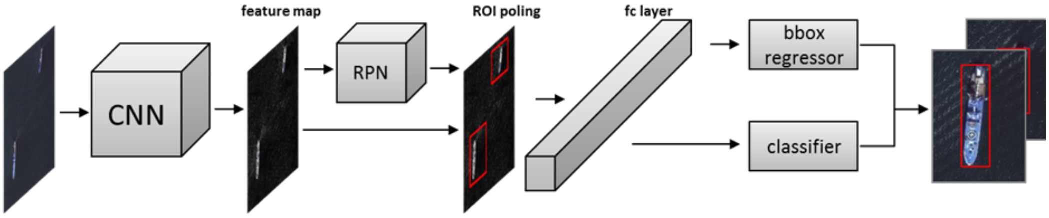


**Figure 3.12: The Faster R-CNN network architecture.**

The RPN is responsible for the prediction of class-agnostic local region proposals, which afterwards are used by the Fast R-CNN. The RPN is testing the CNN's feature maps against the set of spatially distributed anchor boxes with different sizes and aspect ratios. The anchor boxes are built around the anchor points which are equally distributed over the feature map with vertical and horizontal stride set to 16 or 8 pixels for VHR and MR images correspondingly. For VHR images, the anchor boxes are generated at scales [0.5, 1.0, 2.0] and aspect ratios [0.5, 1.0, 2.0]. For MR images scale [0.5, 1.0] and aspect ratios [0.5, 1.0, 2.0] are used. To deal with overlapping region proposals the soft non-maximum suppression (S-NMS) [24] with IoU threshold of 0.6 is applied. After the S-NMS, the top 300 proposals (ranked by their score) are used for further processing. The Fast R-CNN has two inputs: region proposals from the RPN and CNN feature maps. Its objective is to predict object classes/labels and to refine detected bounding boxes.

As the backbone CNN one of the state-of-the-art models ResNet [46] is chosen. During the last few years the deep CNNs have shown very impressive results in image classification [9]. At that time the general trend formed to produce ever deeper networks. On one hand this trend improved classification accuracy, but on the other hand it has its limits; such networks due to their complexity became difficult to train and the accuracy degraded after a certain point. The ResNet is partly solving this problem by skipping connections between some layer stacks which forms residual blocks, as it is shown in Figure 3.13. This approach allows designing deeper networks without the dramatic increase of their complexity. Currently, several ResNet configurations such as ResNet50, ResNet101 and ResNet152 are popular choices to use as a standalone image classification CNN or as a backbone for the object detection frameworks [11]. The main difference between them is the number of layers they have, which is indicated by the numbers 50, 101 and 152 in their names. More layers mean more parameters in the network which would require more hardware resources. The decision on the choice of a network is always a trade-off. The general rule of thumb is that more detailed images and complex objects require more parameters in the CNN in order to build non-linear descriptors. Based on this generalization ResNet50 for MR images and ResNet101 for VHR images have been chosen.

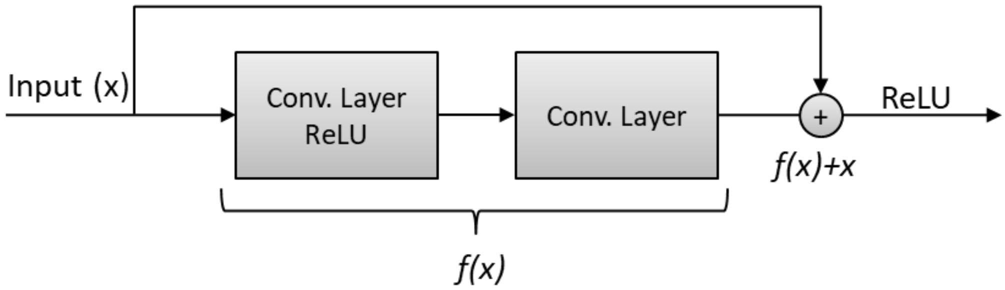


**Figure 3.13: Residual CNN block.**
reproduced from [46]

The Faster R-CNN model does not account for orientation of detected objects. Its outputs are objects horizontal bounding boxes (HBB), their type labels and detection scores. Since detected vessels are always arbitrary oriented, it is not possible to estimate their size directly from the HBB. Several attempts were made to estimate rotated bounding boxes (RBB) by modifying RPNs as presented in [50] [51] [52], but these approaches, while being accurate, involve an additional parameter to the network – angle for region proposals, which dramatically increase the amount of region proposals to test, thus increasing the inference time. In the RS context with enormous large images, especially in the VHR domain, it dramatically increases the overall processing time. In this thesis an alternative solution is proposed, where the RBBs are estimated only on confirmed detections with HBBs. The proposed method is designed specifically to detect vessels, whose generalized shape has rectangular representation, and not applicable for other object types. The process is carried out outside the Faster R-CNN, but it involves its backbone CNN as well as a priori knowledge about the vessel-type dependent dimensional characteristics.

It is assumed that RBB is a close-enough vessel parameter representation. The width and length of the RBB are equal to the beam (width) and length of the vessel; its orientation corresponds to the heading of the vessel normalized to [0..170] degrees range without differentiating where the bow and stern are. To estimate the RBBs, the algorithm accounts for vessel type specific size characteristics which are shown in the Table 3.1 and Table 3.2. Provided numbers are estimated during the collection of training datasets (described in chapters 4.3 and 0), they are based on the AIS reported vessel sizes as well as on manual annotations. These numbers agree with vessel construction standards, which are particularly discussed in [53] [54] [55].

In the first step, the initial size of the rotated bounding box (RBB) is estimated. It is assumed that the vessel length $\boldsymbol{L}$ will not exceed 0.95 of the HBB diagonal and will fit in the length limits of its type, which are shown in the Table 3.1.

**Table 3.1: Vessel type-dependent dimensional parameters.**
Note that all the values in this table are estimated statistically from the VHR training dataset. They are mostly based on vessel appearance on satellite images in combination with supplied AIS (when available) and do not represent the precise dimensional characteristic of any particular vessel class. These values must be interpreted as reference only.

| Class label | min length | max length | min beam | max beam |
|---|---|---|---|---|
| Generic cargo vessel | 10 | 400 | 4 | 60 |
| Oil tanker | 35 | 299 | 6 | 50 |
| Service/tug boat | 8 | 143 | 2 | 70 |
| Generic Leisure boat / skiff / speedboat /rib | 3 | 75 | 2 | 14 |
| Generic Passenger ship / Ferry | 14 | 330 | 4 | 52 |
| Generic Warship | 18 | 206 | 4 | 36 |
| Container carrier | 37 | 400 | 7 | 73 |
| Generic fishing boat | 3 | 65 | 1 | 17 |
| Sloop /Sailing boat | 4 | 94 | 2 | 17 |
| LNG/Chemical/Gas tanker | 20 | 290 | 5 | 74 |
| Patrol vessel | 6 | 90 | 2 | 20 |
| Yacht / Superyacht | 4 | 103 | 2 | 15 |
| Cruise boat | 25 | 297 | 4 | 45 |
| Catamaran | 6 | 21 | 2 | 10 |
| Unknown | 2 | 157 | 1 | 71 |

Then, the vessel beam is calculated as:

$$\boldsymbol{B} = \frac{\boldsymbol{L}}{\boldsymbol{mLBR}} \tag{11}$$

where $\boldsymbol{B}$ is the vessel beam and $\boldsymbol{mLBR}$ represents the mean length-beam ratio. The length-beam ratio is dependent on specific ship design and also not linearly changing within different vessel sizes. Therefore the $\boldsymbol{mLBR}$ was calculated not only for every vessel class separately, but also for six different size classes as it is shown in the Table 3.2.

In addition, if AIS signal at imaging time within the HBB is available and if its reported type can be mapped to the detection type; and difference between the reported beam and length and estimated values is less than the predefined threshold (20% for VHR and 40% for MR scenarios) it is assumed to be the same object with correctly reported dimensional attributes. The reported beam and length in that case are used as $\boldsymbol{L}$ and $\boldsymbol{B}$ for further processing.

**Table 3.2: Vessel type and size dependent *mLBR.***
Note that the ***mLBR*** values are estimated statistically and do not represent precise dimensional characteristic of any particular vessel class, it should be interpreted as reference only. Vessel length ranges is given in meters.

| Length<br>Class label | very small < 10 | small [10-15) | medium [15-20) | medium-large [20-50) | large [50-100) | very large [100-450] |
|---|---|---|---|---|---|---|
| Generic cargo vessel | - | 3.31 | 3.75 | 4.37 | 6.06 | 5.65 |
| Oil tanker | - | - | - | 4.98 | 6.98 | 6.09 |
| Service/tug boat | 4.34 | 3.37 | 3.48 | 3.71 | 5.14 | 6.59 |
| Generic Leisure boat / skiff / speedboat /rib | 2.94 | 3.17 | 3.52 | 4.14 | - | - |
| Generic Passenger ship / Ferry | - | 3.25 | 3.17 | 3.95 | 5.44 | 5.72 |
| Generic Warship | - | - | 2.59 | 4.48 | 5.87 | 4,38 |
| Container carrier | - | - | - | 4.11 | 5.75 | 5.55 |
| Generic fishing boat | 2.8 | 3.31 | 3.25 | 4.01 | 5.04 | - |
| Sloop / Sailing boat | 2.87 | 3.13 | 3.23 | 5.5 | 7.42 | - |
| LNG/Chemical/Gas tanker | - | - | - | 4.05 | 5.91 | 4.35 |
| Patrol vessel | - | 3.35 | 3.3 | 4.15 | 5.5 | - |
| Yacht / Superyacht | 2.71 | 3.31 | 3.47 | 4.14 | 5.48 | - |
| Cruise boat | - | - | - | 7.22 | 6.61 | - |
| Catamaran | 3.04 | 3.45 | 2.74 | 2.14 | - | - |
| Other / Unknown | 3.06 | 3.39 | 3.71 | 3.91 | 4.96 | - |

In the next step, the RBBs with the derived size and rotation from 0 to possible 170 degrees with step of 10 degrees are generated. The amount of generated RBBs is limited to the number that fit into the HBB. The RBB is considered to fit in the HBB if 80% of its area is inside the HBB, thus smaller aspect ratio of the HBB leads to smaller amount of RBBs to test. The center of the RBBs corresponds to the center of the HBB. The resulting RBBs are used to create the square image subsets and to mask out everything outside of their bounds. Then, the cropped image subsets are tested with the backbone CNN that is used in the Faster R-CNN. The RBB with the highest score is considered to be the correct one.

In the next step, the image subset within the correct RBB is binarized. Afterwards the vessel contour is extracted to derive precise vessel length and width. The contour search is based on the algorithm proposed in [56]. With help of principle component analysis (PCA), the extracted contour is used to refine the final vessel heading. The entire process of parameter estimation is visualized with two different examples in Figure 3.14.

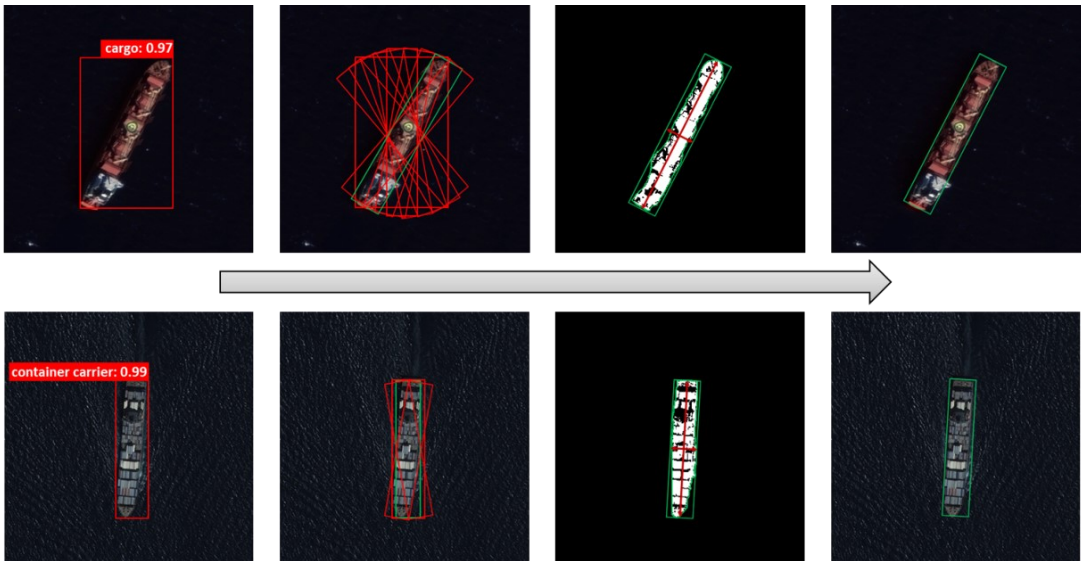


**Figure 3.14: Vessel parameter estimation workflow.**
This visualization is based on the WorldView-3 satellite image © 2020 European Space Imaging / Maxar

The final vessel location is considered to be the centroid of the extracted vessel contour. With the help of the affine transformation of the input satellite image the pixel coordinates are translated into geographical coordinates. Detected objects are matched with AIS reports using nearest neighbor search within the pre-defined distance of 10 meters in open waters and 3 meters in port areas. In the case of successful AIS match, all AIS attributes are assigned to the detected objects. Furthermore, if any of the reported attributes does not match the detected values, it is marked as a probable anomaly which is of potential interest to the end-users.

# 4 Software Implementation and Hardware Environment

This chapter describes the implementation of the vessel detection framework deployed at DLR's Ground Station Neustrelitz. Subchapter 4.1 covers short descriptions of the processing chain and the hardware setup. Subchapter 4.2 describes the main tools developed by the author. Subchapters 4.3 and 4.4 present generated training datasets. Subchapter 4.5 provides descriptions about CNN training configurations.

## 4.1 Earth Observation Maritime Surveillance System (EO-MARISS)

The vessel detection framework is implemented as part of the EO-MARISS (Earth Observation Maritime Surveillance System) [57] which architecture is shown in Figure 4.1.

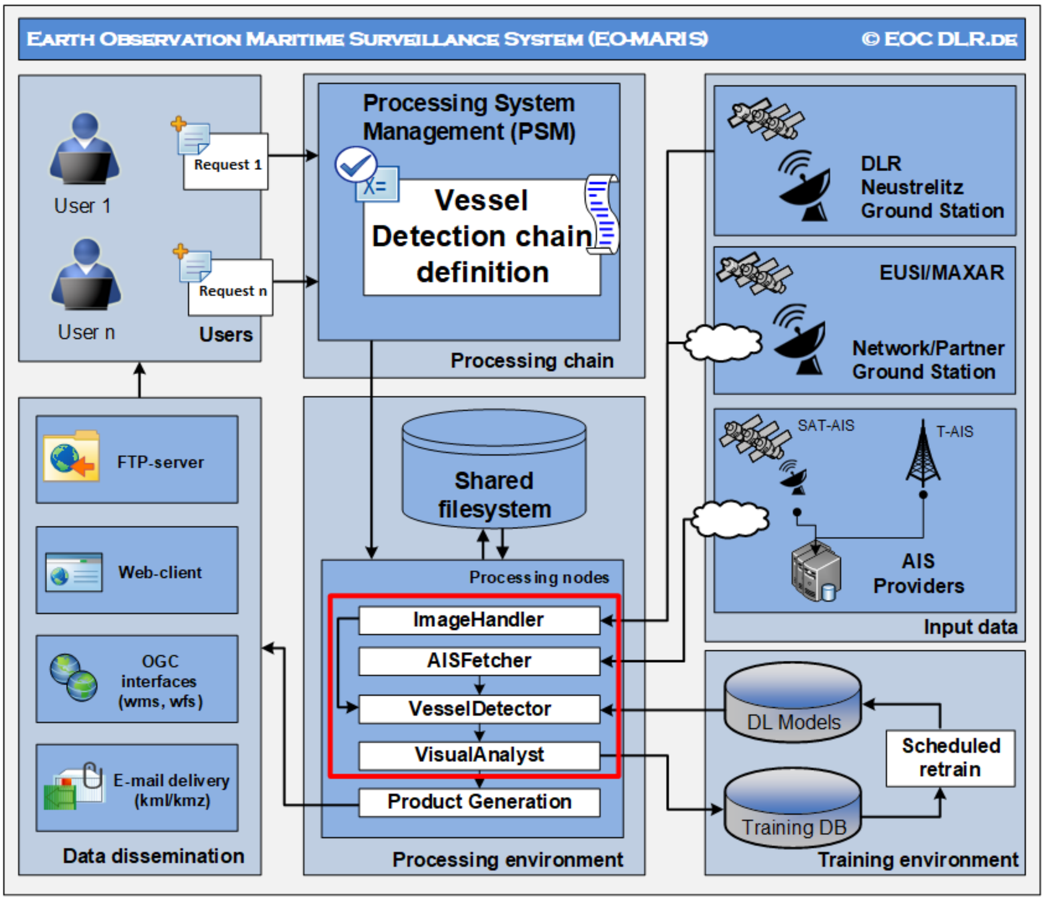


**Figure 4.1: EO-MARISS Vessel Detection Chain.**
The red frame highlights the core processors of the vessel detection framework developed by the author

The system supports request- and/or data-driven scenarios for processing initialization. In the case of the request-driven scenario, a special request file must be filled by the user which indicates the observation AOI, possible time window and which data source (satellite) should be used. The request file triggers the system to generate the runtime workflow and initiate the processing chain. The input satellite imagery is then automatically collected via dedicated pickup points, either from the local ground station or from the partner ground stations and data providers. For the data-driven scenario, the system can be pre-configured for the dedicated data source(s) and AOI(s). In that case, the processing chain is triggered every time upon the new data arrives to the pick-up point.

The processing chain is managed by the Processing System Management (PSM), the software developed jointly by the DLR’s institute DFD (German Remote Sensing Data Center) and private company Werum Software & Systems AG [58]. The PSM is the orchestration software for the hardware resources and the processing workflow. The PSM supports sequential as well as parallel and/or mixed processing workflows.

The vessel detection processing sequence is configured in accordance to the workflow presented in the chapter 3 and shown in Figure 3.1. It starts with simultaneously running the image processing software “ImageHandler” (described in 4.2.1) and the processor “AISFetcher” (described in 4.2.2) which is responsible for collection and interpolation of AIS data. The outputs produced by these processors are used in the next step by the software “VesselDetector” (described in 4.2.3). After that, vessel detection results can be optionally validated and enhanced within the special interactive tool “Visual Analyst” (described in 4.2.4). In the final step, results are distributed to the end users via different dissemination options, which include OGC-standardized raster and vector file formats; file formats specified by the EMSA; and web-based mapping services as described in [57].

The vessel detection system has a self-learning design. This means that all validated detections are constantly filling the training dataset. Once the amount of new training datasets is considered to be sufficient, the system triggers to retrain the CNN models. In the current implementation the sufficiency criterion is set to acquire at least 200 new samples per class.

The processing system is deployed in a cluster of virtual machines sharing the following hardware resources: GPU NVIDIA Tesla™ T4 16GB, 128 GB RAM and 96 physical cores of Intel(R) Xeon(R) CPU E7-4870 v2 2.30GHz. Every processor is encapsulated together with its external dependencies into Docker [59] container.

The training environment is configured on independent server with following hardware: 2x Eight-Core Intel® Xeon® Processor E5-2667 v4 3.20GHz, 256 GB RAM and NVIDIA® Tesla™ V100 GPU card.

## 4.2 Tools developed

### 4.2.1 Processor “ImageHandler”

The processor “ImageHandler” is responsible for image enhancement, orthorectification and mosaicking (in case of multi-strip acquisition). For performance reasons, the processor is written in C++ programming language with use of APIs from the GDAL (Geospatial Data Abstraction Library) [60] and the Intel TBB (Thread Building Blocks) [61] libraries. The GDAL library is used mainly for raster read/write operations as well as for image warping while the Intel TBB serves for parallelization of the application.

The functional diagram of the ImageHandler is visualized in Figure 4.2. In case of a multi-strip acquisition, the processor runs AC in parallel threads for every image strip. One additional thread is run for production of Digital Elevation Model (DEM) mosaic, which will be used for the orthorectification process.

The implemented AC algorithm is described in the chapter 3.1.1. For computation efficiency the entire image is read into the RAM buffer. All pixel modifications are implemented in place without any intermediate data copying. The image histogram is used to generate a look-up table with atmospherically corrected values. This approach allowed to run iteration over the image pixels only twice and to avoid complex computations within the loop. At first iteration the image histogram is calculated. The histogram for every band is calculated in parallel threads. In the second iteration the pixel values are replaced with calculated look-up tables (for each band and raster line in parallel). With this implementation the entire AC of one single image file has duration that is slightly longer than file copy on the SSD (Solid-State Drive).

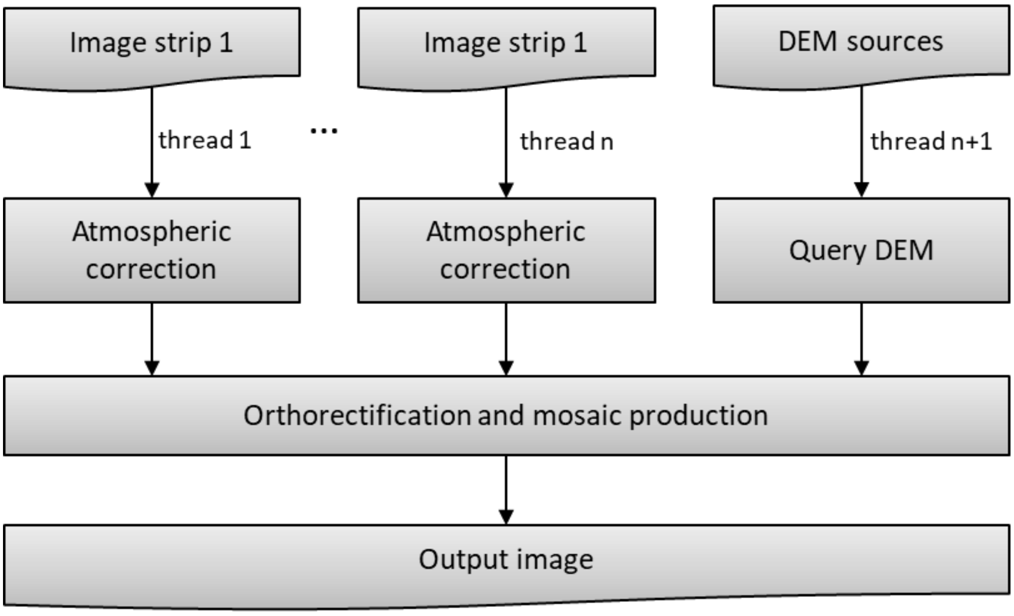


**Figure 4.2: ImageHandler functional diagram.**

The orthorectification (described in chapter 3.1.2) is only applied when the RPC model for the image is available (usually the case for VHR images).

The RPC-based orthorectification and image mosaicking processes (in case of multi-strip collection) are carried out with the help of the Warp C++ API, part of the GDAL library. The following DEMs are used on the basis of the best available resolution at location and available license for specific end-user: Copernicus DEM GLO-90 [62], DLR’s SRTM-X DEM [63], Copernicus EU-DEM [64], Copernicus DEM GLO-30 [62] and Copernicus DEM EEA-10 [62].

The resulting image is used in the VesselDetector processor, and also delivered to the end-users by request.

### 4.2.2 Processor “AISFetcher”

The purpose of the processor “AISFetcher” is to collect and preprocess the AIS data in accordance to the image extent and acquisition time. The processor is written in Java programming language, which offers a rich API and extensions for processing of http data streams and database interactions. In addition, it allows deeper integration with PSM, which is also a Java application. The processor’s functional diagram is visualized in Figure 4.3.

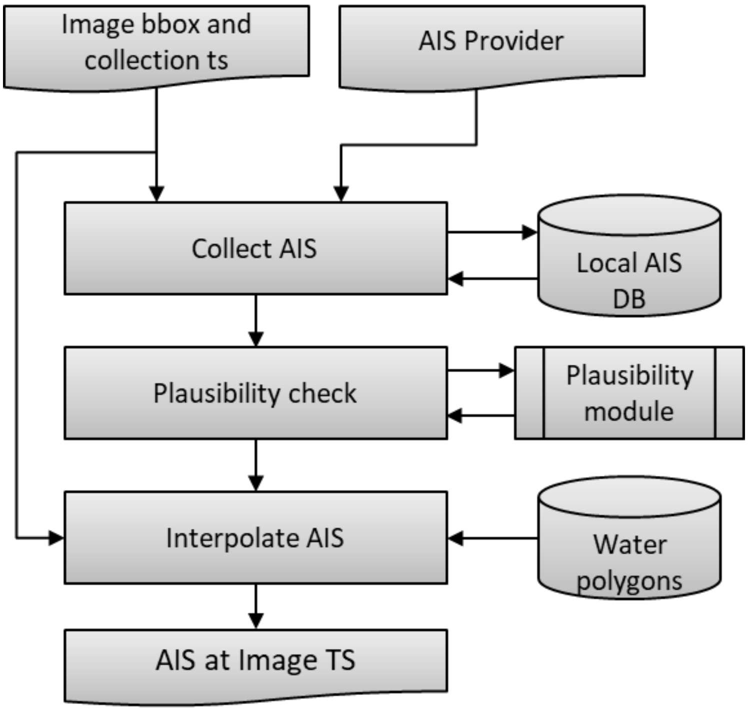


**Figure 4.3: AISFetcher functional diagram.**

Current implementation supports http query APIs from several AIS providers as well as decoding raw AIS data from the receiver (in NMEA format). Selection of any particular AIS data source is a subject of agreement with any particular end-user. All results of successful requests from the AIS providers are stored for reproduction capacities in the local SQL DB.

Every sequence of extracted AIS messages are processed with embedded AIS Plausibility Module [65] developed in the Institute of Communications and Navigation, part of DLR. The module estimates plausibility of vessel movement taking into account dynamic attributes of AIS messages, such as speed or course over ground. In case of suspicious behavior, corresponding messages are marked as to be potentially anomalous.

The implemented AIS interpolation algorithm is described in the chapter 3.2. The processor fills the time gaps between the real AIS reports and then calculates positions for the exact acquisition time within the image. Resulting point dataset is used in the VesselDetector processor for the two following purposes: to attract an attention at global region search stage and to identify detected targets if their positions correlate with AIS reports.

### 4.2.3 Processor “VesselDetector”

The vessel detection processor “VesselDetector” implements the algorithm described in the chapter 3.3. It is written in C++ programming language and uses the following software libraries: GDAL [60], OpenCV [66], TensorFlow [45] and Intel TBB [67]. The processors functional diagram is shown in Figure 4.4.

During the Global Region Search stage, as it is described in the chapter 3.3.2, the land areas are masked out with the help of water polygon layer. The vector dataset containing water polygons is extracted for the exact geographical extent that corresponds to the input image. Afterwards, a binary raster mask with the same GSD (ground sampling distance) and dimensions as in the input image is generated with help of GDALRasterize API [60]. Resulting water mask is used to set pixel values in the input image to zero so that they will not produce any false recalls by the CNNs in the later processing. Furthermore, image tiles containing zero pixels only are completely excluded from the remaining processing steps. The image classification CNN is used to classify the image tiles as vessel containing or not containing regions. In addition to the image classification CNN results, complementary region proposals are generated on the basis of AIS reports and historical detections within the geographical bounds of the input image.

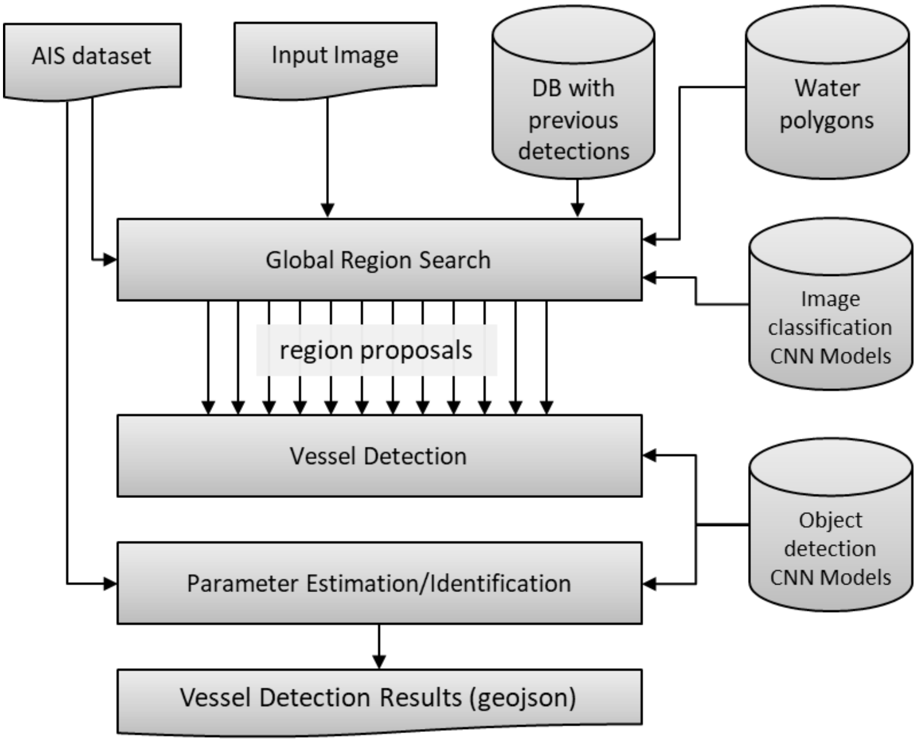

AIS dataset
Input Image
DB with previous detections
Water polygons
Global Region Search
Image classification CNN Models
region proposals
Vessel Detection
Object detection CNN Models
Parameter Estimation/Identification
Vessel Detection Results (geojson)

**Figure 4.4: VesselDetector functional diagram.**

Vessel detection and parameter estimation methods are described in the chapter 3.3.3. Object detection CNN is used to detect vessels without their orientations; however, its backbone CNN is used once again in the post-processing step for vessel parameter estimation. This process accounts ancillary AIS dataset for vessel identification and/or potential anomaly detection.

For performance reasons, image tiling and region selection for the image classification and object detection CNNs is done virtually, without copying the image subsets from the original input image. Instead, a pointer to the fraction of RAM containing the selected image is fed directly to TensorFlow [45] session object with activated CNN model to perform inference.

In the current implementation, the image classification CNN MobileNet [10] and object detection Faster R-CNN [18] with backbone ResNet101 (for VHR) and ResNet50 (for MR) [43] are used. Nevertheless, the software design allows to use any image classification and object detection CNN model compatible with the machine learning framework TensorFlow.

The output produced by the VesselDetector is a geojson file, which contains all estimated attributes and rectangular vessel polygons with geographical and image pixel coordinates.

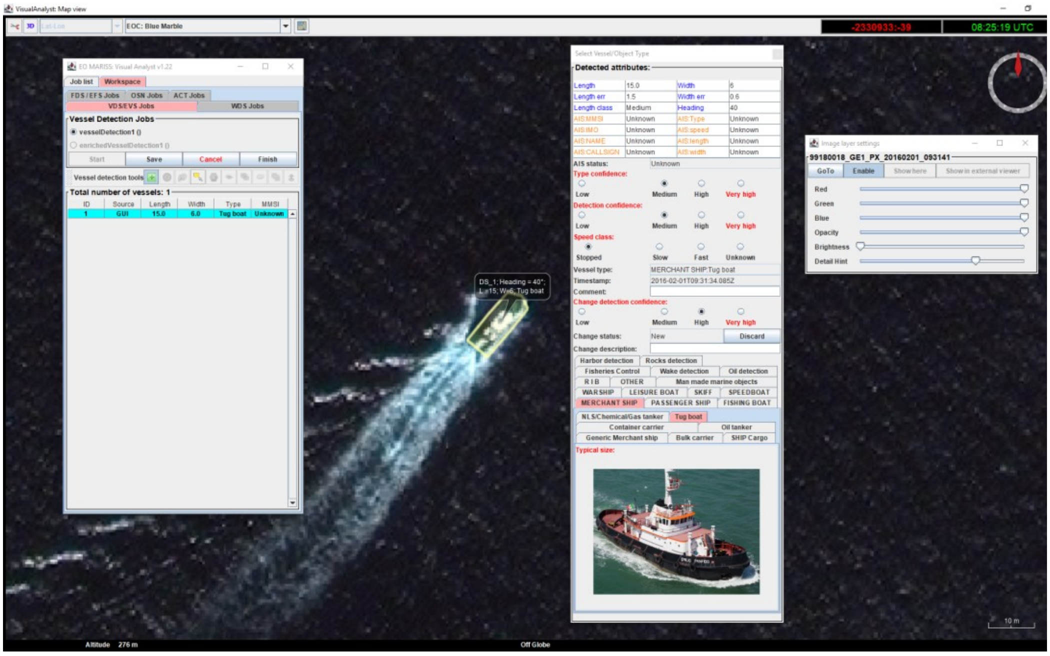
### 4.2.4 Analysis tool “Visual Analyst”

The Visual Analyst tool is the only (and optional) manual step in the entire processing chain. It serves for the following purposes: manual vessel detection and validation/correction of automatically detected vessels. It is an interactive tool that combines GIS and image analysis functionalities. The software is written in Java and based on the NASA WorldWind Java API [68]. Figure 4.5 shows a screenshot of the Visual Analyst user interface.

The Visual Analyst is composed of different modules which can be divided in two main groups: Visualization and Analysis. The Visualization modules are responsible for visual representation of EO derived image products. The internal map frame supports different configuration options for visualization such as specifying different band combinations, change brightness or opacity and others.

The Analysis modules provide all the functionalities needed for analysis and classification, for example such as adding/removing or modifying detected vessel geometries as well as setting different annotation attributes.

The Visual Analyst is the client application which is running on the remote working stations on a thin client. Multiple requests can be processed by different operators simultaneously. The application uses socket-based connection with PSM for the data exchange. The full resolution image data is served to the client via WMS [69] interface from the Geoserver [70] instance located on the same host as the PSM.



Vessel Detection Jobs

● vesselDetection1 ()

○ enrichedVesselDetection1 ()

Total number of vessels: 1

| ID | Source | Length | Width | Type | MMSI |
|---|---|---|---|---|---|
| 1 | GUI | 15.0 | 6.0 | Tug boat | Unknown |

Detected attributes:

| | | | |
|---|---|---|---|
| Length | 15.0 | Width | 6 |
| Length err | 1.5 | Width err | 0.6 |
| Length class | Medium | Heading | 40 |
| AIS:MMSI | Unknown | AIS:Type | Unknown |
| AIS:IMO | Unknown | AIS:speed | Unknown |
| AIS:NAME | Unknown | AIS:length | Unknown |
| AIS:CALLSIGN | Unknown | AIS:width | Unknown |

AIS status: Unknown

Type confidence: ○ Low ● Medium ○ High ○ Very high

Detection confidence: ○ Low ● Medium ○ High ○ Very high

Speed class: ● Stopped ○ Slow ○ Fast ○ Unknown

Vessel type: MERCHANT SHIP:Tug boat

Timestamp: 2016-02-01T09:31:34.085Z

Comment:

Change detection confidence: ○ Low ○ Medium ● High ○ Very high

Change status: New

Change description:

99180018_GE1_PX_20160201_093141

**Figure 4.5: Visual Analyst user interface.**
This Visual Analyst tool allows modifying and creating new vessel objects and setting different attributes. For better classification by the operator, an embedded photograph for every vessel class is available.

## 4.3 VHR Training dataset

The VHR vessel detection training dataset was generated from the collection of WorldView-[1-3] and GeoEye-1 satellite images acquired around the world in the densest shipping areas. These images were provided by the European Space Imaging company and restricted exclusively for use within the joint projects. The first version of the dataset was presented in [71] and in [72]. It contained more than 36 000 of annotated vessels of 9 different classes: yacht, sailing boat, passenger ship, service/tug, fishing boat, container carrier, tanker, cargo and warships. The annotated attributes were vessel classes and horizontal bounding boxes (HBB). Selected classification was based mainly on vessel classes present in AIS specifications [73]. Later on, it was discovered, that presented classification can be extended, for example class “tanker” can be divided into “oil tanker” and “LNG/chemical tanker”. Furthermore, additional attributes about the vessel, for example its geographical coordinates, dimensions, orientation as well as AIS-derived information might be of high interest for further research.

The second version (v.2) of the dataset is initially generated out of 100+ scenes and is continuously extended with the new samples produced by the system itself. The images have different sizes (from about 5 000 x 5 000 to 200 000 x 100 000 pixels), different aspect ratios and a spatial resolution of 0.3-0.5m per pixel. The dataset has 14 vessel classes (as shown in Figure 4.6) plus 1 additional "unknown" class for those vessels which were hard to classify. In the current state (06-2020) the dataset contains nearly 40 000 of unique annotated vessels, with 500 - 5 000 for every class. One unique annotation means one unique appearance on the satellite image.

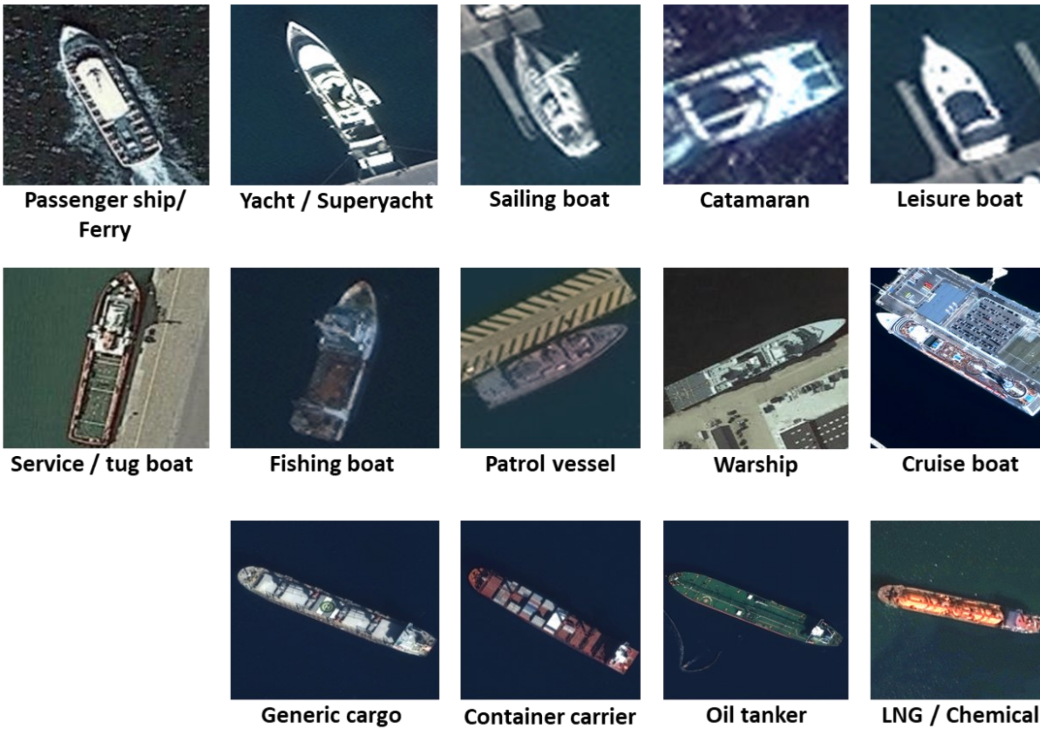


**Figure 4.6: VHR Vessel detection training dataset classification.**
This visualization is based on the collection of WorldView-3 and GeoEye-1 satellite images © 2020 European Space Imaging / Maxar

The annotations in the v.2 dataset include the following information: pixel coordinates, geographical coordinates, horizontal bounding box, rotated bounding box, true heading, object state (stopped or moving), object dimensions and all available attributes derived from the AIS (except identification). One training image sample is a 1500x1500 pixels crop from the full resolution satellite scene which can be either PAN or RGB images and may contain one or more vessels. Image samples are created using sliding window over the full satellite scene with the step size of 300 pixels in both, x and y directions. If any annotated object found within the sliding window it is used to create an image crop. As the result, the same object/object part may appear within different image crops, but it will have always different pixel coordinates. This can be considered as sort of data augmentation. Depending on the vessel classes appearing within the image crops, up to 15 additional augmented samples may be produced in order to harmonize the distribution of training samples between the classes. For augmentation different color filters, rotations and flips were applied. Several color filtering strategies have been developed: addition of green or blue colors in order to imitate possible atmospheric effects; transformation to black and white to imitate PAN (in case of original RGB image); and smoothing with averaging filter with kernel size of 5. For rotated augmentation random angles in the range of [0-359] degrees were used.

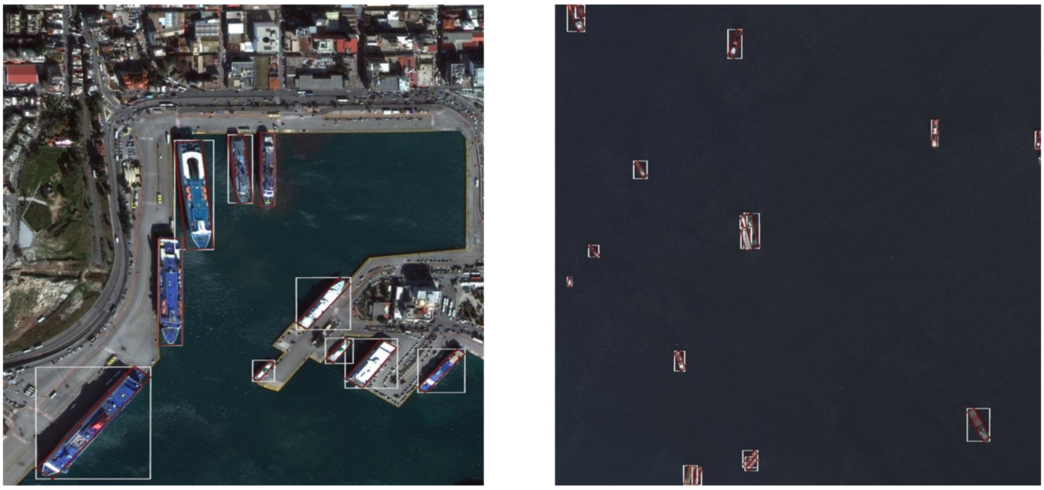
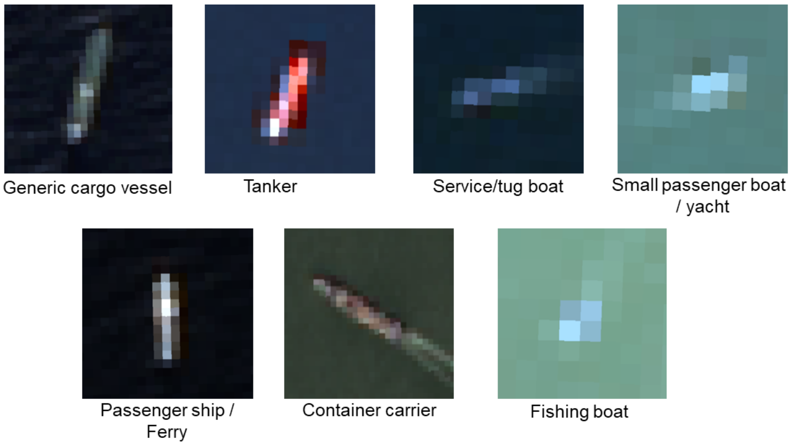


**Figure 4.7: Examples of VHR training samples.**
overlaid horizontal bounding boxes (white) and rotated bounding boxes (red)
This visualization is based on the collection of WorldView-3 and GeoEye-1 satellite images © 2020 European Space Imaging / Maxar

## 4.4 MR Training dataset

The MR vessel detection training dataset is initially generated from the collection of 100 pan-sharpened Landsat-8 images. All images have similar sizes, which is in average 16 000 x 16 000 pixels. The dataset has the same structural design as VHR Training dataset v.2, but different vessel classification and image crop sizes.

One training image sample is a 300x300 pixels crop from the full resolution satellite images with RGB color space and may contain one or more vessels. Image samples are created using sliding window over the full satellite scene with step size of 100 pixels in both, x and y directions. The augmentation strategy is the same as in VHR v.2. The dataset has 7 vessel classes (as shown in Figure 4.8). In the current state (06-2020) the dataset contains nearly 14 000 of unique annotated vessels, with 1 000 - 3 000 for every class.


Generic cargo vessel

Tanker

Service/tug boat

Small passenger boat / yacht

Passenger ship / Ferry

Container carrier

Fishing boat

**Figure 4.8: MR Vessel detection training dataset classification.**
This visualization is based on the Landsat-8 satellite images ©2020 USGS

## 4.5 Training configuration

All CNN models used in this project are pre-trained on the ImageNet [17] dataset. Two instances of MobileNet models are retrained, one for VHR and another for MR data. Both MobileNet instances trained for the two classes only: vessel and non-vessel. To train them the TF-slim API [74] has been used.

The VHR MobileNet instance was retrained on 400 000 full resolution image crops with the size of 224x224 pixels. Additional 50 000 image snippets were used for training evaluation. This dataset is derived from the original multiclass dataset described in the subchapter 4.3. The samples which are labelled as vessels may contain entire vessel(s) or just some parts of it. In particular, this is mostly the case for very large vessels, such as cargo or tankers. The network is re-trained for 30 epochs with constant learning rate of 0.001 and mini-batch gradient descend optimizer with batch size of 64.

The MR MobileNet instance was retrained on 10 000 scaled image samples to the size of 224x224 pixels. Another 2 000 samples were used for evaluation. The dataset for MR MobileNet is derived from the original multiclass dataset described in the chapter 4.4. The network is re-trained for 20 epochs with constant learning rate of 0.001 and mini-batch gradient descend optimizer with batch size of 64.

The object detection CNNs are trained with the TensorFlow Object Detection API [11] [75]. For VHR vessel detection, the Faster R-CNN ResNet-101 was retrained on 90% of the complete VHR v.2 dataset. Another 10% is used for evaluation. The network is re-trained for 20 epochs with batch size of 1, initial learning rate of 0.0003 with scheduled learning rate decrease (on epoch 5 → 0.00003 and on epoch 10 and onwards → 0.000003) and SGD optimizer with momentum 0.9.

The Faster R-CNN ResNet-50 that is used for MR vessel detection was retrained on 90% of the MR dataset and another 10% is used for evaluation. The network is re-trained for 15 epochs with the learning rate set to 0.0001 and decreased to 0.00001 on epoch 5, and mini-batch gradient descend optimizer with batch size of 2.

The initial learning rates in this project are set to small values because all models are pre-trained on large dataset ImageNet [17] and higher values may introduce a risk to lose previously acquired knowledge. Other training hyperparameters are determined empirically by using different commonly used configurations and analyzing the loss values and evaluation results after every epoch. It is a still an open research question to find the best configuration. Furthermore, the increasing amount of training data samples may require to revise hyperparameters accordingly.

# 5 Results and Discussions

The result of this dissertation is a methodology for vessel detection which is implemented as fully functional software framework. It operates with VHR images acquired from WorldView-[1-3] and GeoEye-1 satellites as well as with MR images acquired from Landsat-8 satellite. The developed method includes image and AIS data pre-processing, two-stage detection and fusion with AIS methods. The presented algorithms are implemented in the form of independent software processors which are configured to run in predefined sequence. One of the key requirements for the implementation was NRT applicability of developed methods. Therefore, the focus was on the efficient use of existing or newly created techniques.

The core algorithm for vessel detection is based on the use of artificial neural networks, namely CNNs. For that purpose, two versions of training datasets were generated: VHR and MR. The initial VHR training dataset is produced from the set of more than 100 of WorldView-[1-3] and GeoEye-1 images and contains about 40 000 of uniquely annotated vessels divided in 14 different classes. The initial MR training dataset is generated from the set of 100 of Landsat-8 images and contains about 14 000 of uniquely annotated vessels of 7 different classes. During the framework operation, both datasets are constantly filling with the new data in order to potentially increase detection accuracies in the future.

In order to provide the performance overview of the developed framework, a special benchmark with a set of 25 VHR and 25 MR images was accomplished. These images did not contribute to the training dataset and was not used for CNN training evaluation. However, most of them have the same geographical coverages as the images in the training dataset – the densest port and marine trafficking areas around the world, but acquired at different dates. All the vessels on the test images were with help of AIS or completely manually annotated. Even those objects which had AIS signals were manually validated. For the accuracy assessment following metrics has been calculated: detectability, F1 score and accuracies of the estimated vessel parameters length, beam and heading. The detectability represents a percentage of all the detected vessels on the image in relation to all manually annotated vessels. The F1 score is used to evaluate the classification accuracy among detected vessels. It combines classification precision and recall by taking their harmonic mean. The F1 score (presented in Table 5.1 and in Table 5.3) is calculated as follows:

$$\boldsymbol{F1 = 2 \cdot \frac{Precision\ \cdot\ Recall}{Precision\ +\ Recall}} \tag{12}$$

where

$$\boldsymbol{Recall = \frac{tp}{tp\ +\ fn}} \quad \textbf{and} \quad \boldsymbol{Precision = \frac{tp}{tp\ +\ fp}} \tag{13}$$

where ***tp*** is true positives, ***fn*** is false negatives and ***fp*** is false positives.

The parameter accuracies (presented in Table 5.2 and in Table 5.4) are estimated as:

$$\boldsymbol{pA = \frac{min(ev, tv)}{max(ev, tv)}} \tag{14}$$

where ***pA*** is estimated accuracy for particular parameter, ***ev*** is estimated value and ***tv*** is ground truth value.

Besides the accuracy metrics, the processing time of the processors VesselDetector and ImageHandler was measured. The processing time of AISFetcher is considered to be not critical point to evaluate, as it runs in parallel to ImageHandler, and during all the tests it was simply faster, and thus did not affect the overall processing time.

## 5.1 Framework performance on VHR images

The measured averaged detectability from the validation VHR satellite images was about 67%. In the open waters the detectability achieved 84%, whereas in the coastal areas it was 49% only. The coastal area is considered to be within the distance of 50 meters from the coastline or any man-made infrastructure. The lower detectability in the coastal areas was expected. The main obstacle in those regions is related to the high concentration of very small boats which sometimes occupying a few pixels only without clear visible pattern, like those shown in Figure 5.1.

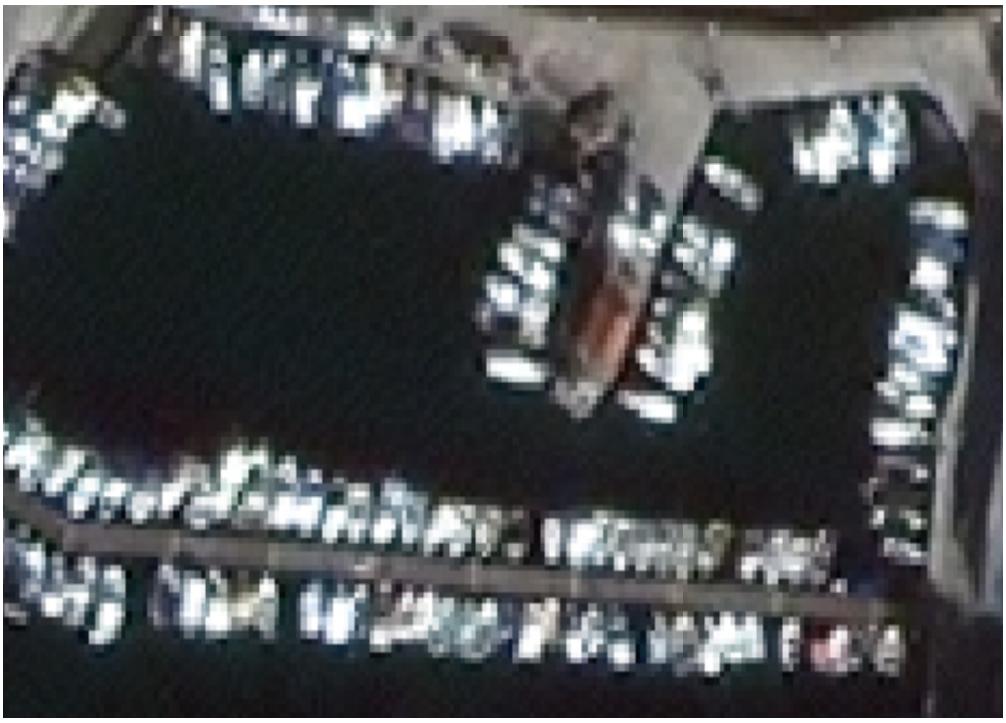


**Figure 5.1: Harbor area with high concentration of small boats.**
Image resolution per pixel is 0.5x0.5 m, all boats have length of less than 10 m.
Image credit: GeoEye-1 © 2020 European Space Imaging / Maxar

The F1 score is used to evaluate the classification performance. The classification confusion matrix as well as corresponding F1 scores for each class are shown in Table 5.1. The highest F1 score of 0.86 was achieved for the warship class, whereas the lowest F1 of 0.38 was for the yacht class. The warship class can be disregarded from the assessment because in the test dataset there were only 8 vessels of this class, however other classes with the highest F1 scores are representing the real picture. Larger vessel classes have higher F1 scores than smaller. Such diversity in F1 scores between the classes was expectable. In particular, large vessels, such as cargoes, container carriers or tankers may be treated as easy objects; they are mostly large and have some unique features present on them. On the other hand, yachts or sailing boats are mostly small white objects which are difficult to differentiate from other small boats such as leisure boats, although the latter ones showed relatively high F1 score.

**Table 5.1: Classification confusion matrix and F1 score of detected vessels from VHR validation dataset.**
**Class IDs:** 1 - passenger / ferry; 2 - yacht; 3 - sailing boat; 4 – catamaran;
5 - leisure boat; 6 - service / tug; 7 – fishing; 8 – patrol; 9 – warship; 10 – cruise;
11 – cargo; 12 - container carrier; 13 - oil tanker; 14 - LNG / Chemical tanker
The numbers in the main diagonal representing the amount of vessels whose predicted class match to the ground truth.

| | | Ground truth | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | Total | F1 |
| Predicted | 1 | 35 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 4 | 3 | 0 | 0 | 0 | 47 | 0.84 |
| | 2 | 0 | 33 | 16 | 11 | 42 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 103 | 0.38 |
| | 3 | 0 | 12 | 53 | 1 | 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 87 | 0.55 |
| | 4 | 0 | 6 | 7 | 28 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 48 | 0.61 |
| | 5 | 0 | 21 | 31 | 4 | 312 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 368 | 0.83 |
| | 6 | 0 | 0 | 0 | 0 | 0 | 121 | 62 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 186 | 0.73 |
| | 7 | 0 | 0 | 0 | 0 | 0 | 23 | 95 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 118 | 0.69 |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0.83 |
| | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 6 | 0.86 |
| | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 2 | 0 | 0 | 0 | 17 | 0.76 |
| | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 72 | 16 | 8 | 4 | 103 | 0.73 |
| | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 55 | 0 | 0 | 66 | 0.80 |
| | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 0 | 41 | 6 | 53 | 0.77 |
| | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 4 | 33 | 38 | 0.81 |
| | Total | 36 | 72 | 107 | 44 | 387 | 144 | 157 | 14 | 8 | 20 | 94 | 71 | 53 | 43 | 1250 | |

To evaluate accuracy of estimated vessel parameters the averaged accuracies for every class have been calculated. Table 5.2 provides the mean accuracies of estimated parameters per vessel class.

**Table 5.2: Averaged accuracy of estimated parameters of detected vessels from VHR validation dataset.**
The values are calculated in accordance to (14)

| | length | beam | heading (0..180)° |
|---|---|---|---|
| passenger / ferry | 0.88 | 0.79 | 0.81 |
| yacht | 0.68 | 0.58 | 0.72 |
| sailing boat | 0.71 | 0.86 | 0.6 |
| catamaran | 0.83 | 0.6 | 0.5 |
| leisure boat | 0.69 | 0.66 | 0.81 |
| service / tug | 0.88 | 0.83 | 0.92 |
| fishing | 0.85 | 0.89 | 0.82 |
| patrol | 0.81 | 0.76 | 0.91 |
| warship | 0.88 | 0.86 | 0.8 |
| cruise | 0.87 | 0.82 | 0.9 |
| cargo | 0.92 | 0.94 | 0.91 |
| container carrier | 0.87 | 0.88 | 0.92 |
| oil tanker | 0.95 | 0.91 | 0.96 |
| LNG / Chemical | 0.9 | 0.89 | 0.95 |

Provided mean accuracies showing similar picture to that with F1 scores. The general trend is that larger vessels are not only easier to detect and classify, but also to estimate their dimensions and orientations. Some detection results from VHR satellite images are demonstrated in Figure 5.4, in Figure 5.3 and in Figure 5.4.

Due to very different VHR image sizes in the validation dataset, which are varying from 10 000 pixels to maximum 200 000 pixels in one dimension, it is hard to impossible to estimate the absolute processing speed metrics. Because of this reason, it was decided to calculate the average processing speed of one square image block (evaluation block) with 10 000 pixels in one dimension, for every scene and then for the entire dataset. For that, the overall processing time is divided by the amount of evaluation blocks that would fit into the scene.

The average processing time for one evaluation block needed by the ImageHandler was 31 seconds and additional 42 seconds needed by the VesselDetector. Nevertheless, for VesselDetector this value should not be treated as an absolute metric, as it highly depends on several unpredictable factors, such as: the number of the detected regions at the global region search stage as well as the number of objects in particular region.

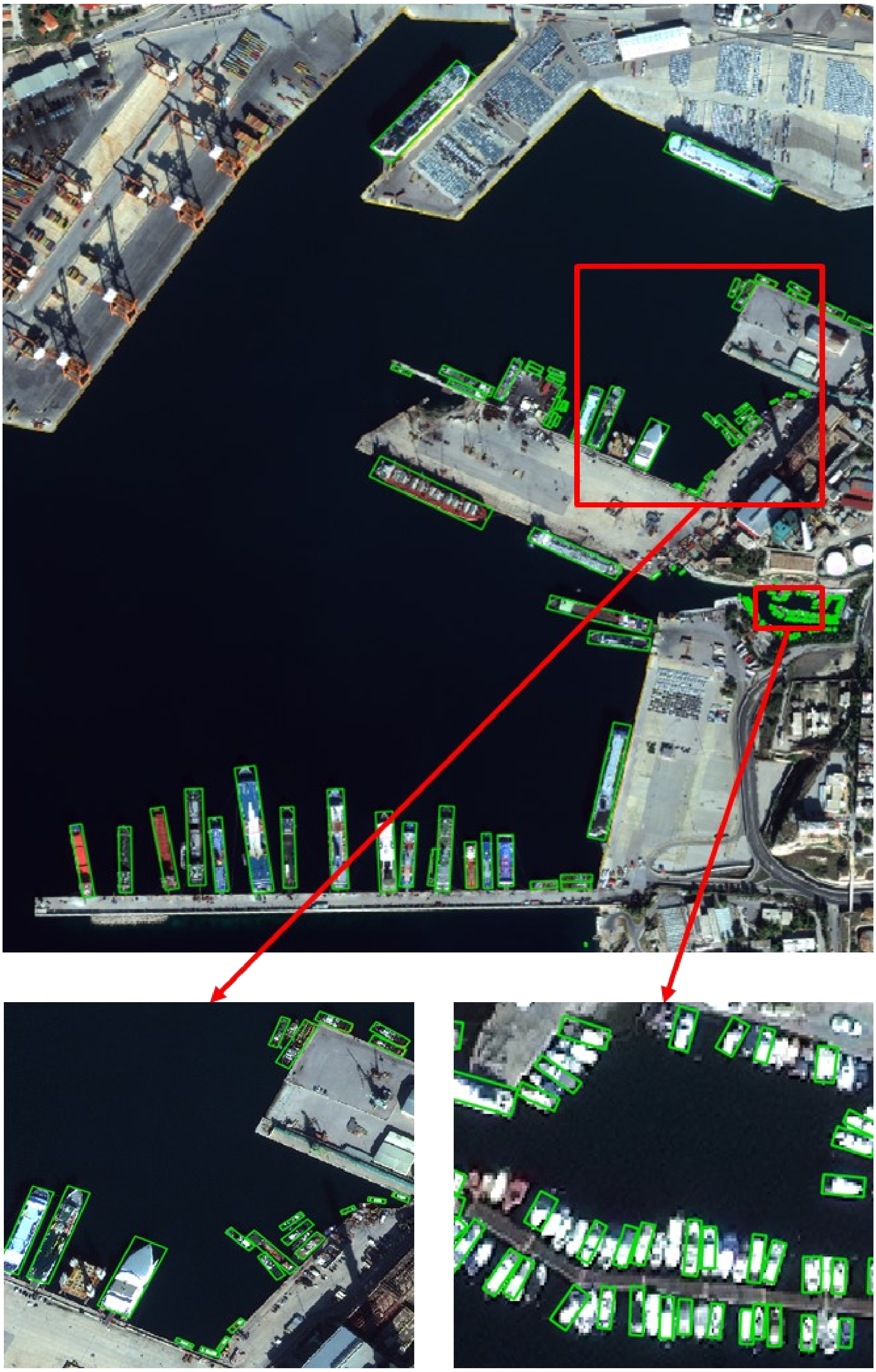


**Figure 5.2: Example of VHR vessel detection results in port and harbor areas.**
Greece, 2016, zoomed-in image clip
Image credit: GeoEye-1 © 2020 European Space Imaging / Maxar

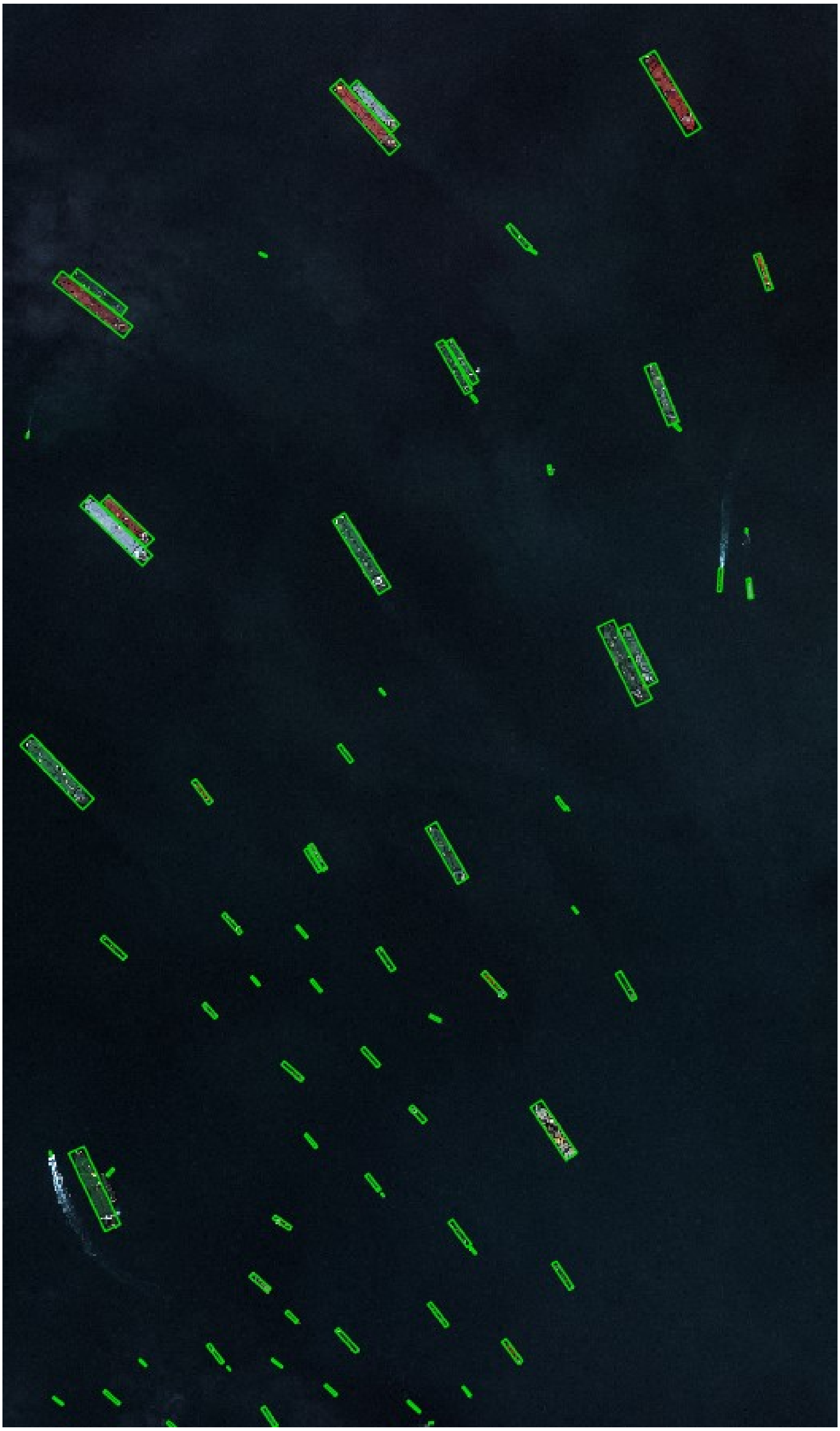


**Figure 5.3: Example of VHR vessel detection results in open sea.**
Singapore, 2015, zoomed-in image clip
Image credit: WorldView-2 © 2020 European Space Imaging / Maxar

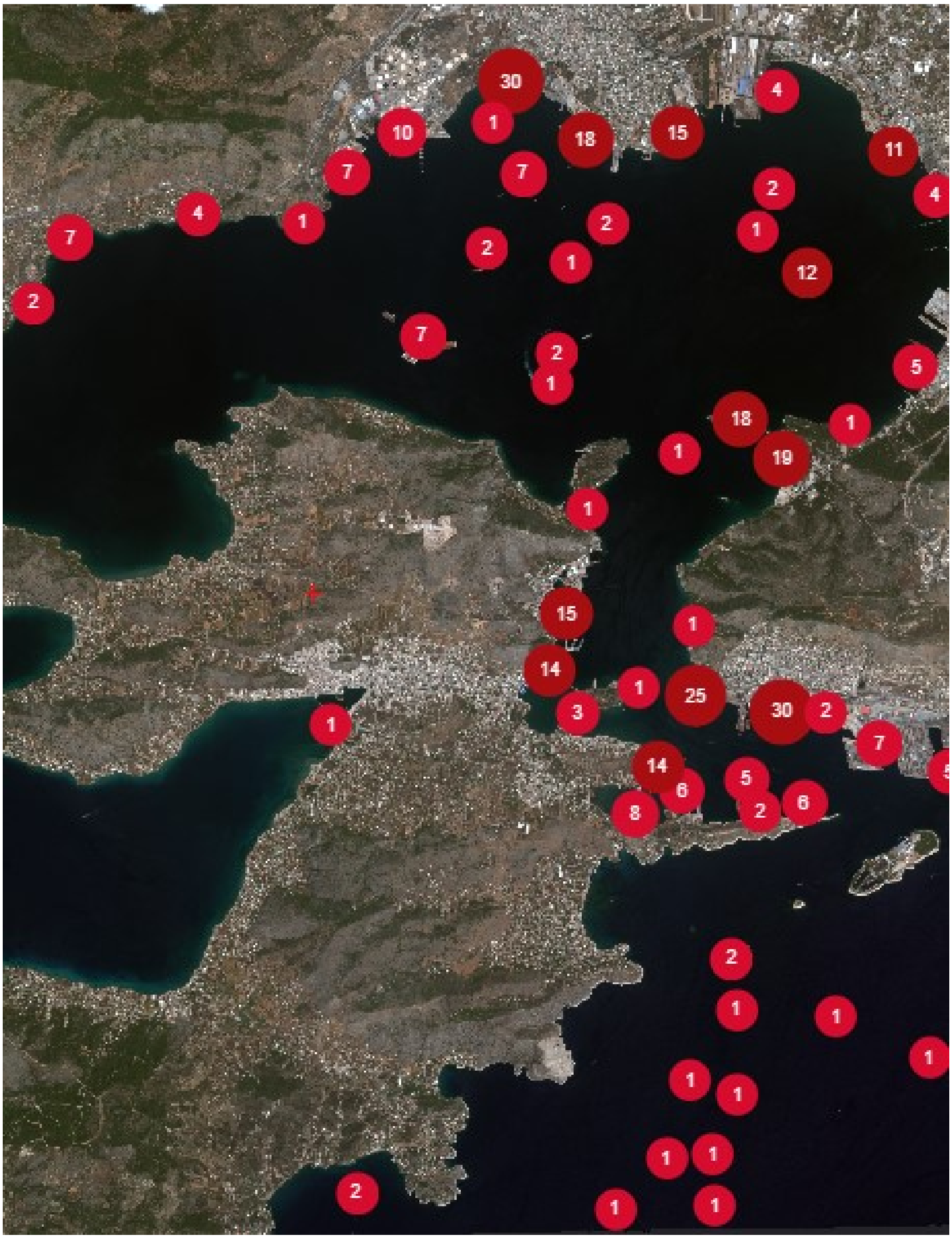


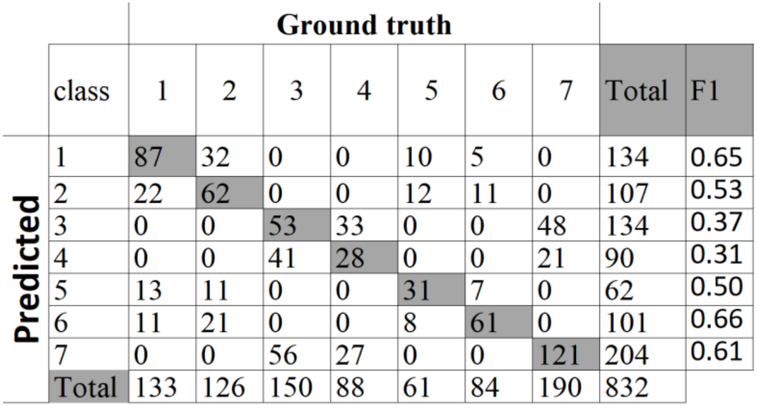
**Figure 5.4: Example of VHR vessel detection results – full satellite scene.**
Greece, 2017, full satellite image scene. Clustered visualization of detection results. The number in the red circles represents the amount of vessels detected at particular image region.
Image credit: GeoEye-1 © 2020 European Space Imaging / Maxar

## 5.2 Framework performance on MR images

The measured averaged vessel detectability from the validation MR satellite images was about 62%. Since the coastal areas are excluded from the detection in the MR scenario, there was no additional location dependent evaluation. After the detailed inspection, it was observed one expectable effect that larger vessels are easier to detect.  This is a similar problem to that occurred with the VHR scenario, but with shifting low detection scores to larger physical vessel sizes. The lower image resolution of Landsat-8, which is 15 meters, sets the limitation on detectable object sizes. For example, a medium sized fishing or tug boat with length of 30 meters would appear as a few bright pixels only as it demonstrated in Figure 4.8. However, under certain circumstances they are still detectable. When these kinds of targets are in the open waters and in the moving states they produce unique patterns. Thus, the threshold of 30 meters is considered to be the minimum detectable vessel size on the MR Landsat-8 images.

The F1 score is used to evaluate the classification performance. The classification confusion matrix as well as corresponding F1 scores for each class are shown in Table 5.3.

**Table 5.3: Classification confusion matrix and F1 score of detected vessels from MR validation dataset.**
**Class IDs:** 1 – generic cargo; 2 - tanker; 3 - service / tug; 4 – small passenger; 5 - passenger / ferry; 6 - container carrier; 7 – fishing
The numbers in the main diagonal representing the number of vessels whose predicted class match to the ground truth.



| | | Ground truth | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Total | F1 |
| Predicted | 1 | 87 | 32 | 0 | 0 | 10 | 5 | 0 | 134 | 0.65 |
| | 2 | 22 | 62 | 0 | 0 | 12 | 11 | 0 | 107 | 0.53 |
| | 3 | 0 | 0 | 53 | 33 | 0 | 0 | 48 | 134 | 0.37 |
| | 4 | 0 | 0 | 41 | 28 | 0 | 0 | 21 | 90 | 0.31 |
| | 5 | 13 | 11 | 0 | 0 | 31 | 7 | 0 | 62 | 0.50 |
| | 6 | 11 | 21 | 0 | 0 | 8 | 61 | 0 | 101 | 0.66 |
| | 7 | 0 | 0 | 56 | 27 | 0 | 0 | 121 | 204 | 0.61 |
| | Total | 133 | 126 | 150 | 88 | 61 | 84 | 190 | 832 | |

The highest F1 score of 0.66 was achieved for the container class, whereas the lowest F1 of 0.31 was for the small passenger class. As it was with VHR scenario, the larger vessel classes have higher F1 scores than the smaller. Small objects of classes 3, 4 and 7 are mostly having very similar appearance and therefore most of inter-class confusions occurred between them.

To evaluate accuracy of estimated vessel parameters the averaged accuracies for every class have been calculated. Table 5.4 provides the mean accuracies of estimated parameters per vessel class. Comparing to VHR results, parameter estimation from MR detections have significant lower accuracy. This is especially notable on smaller objects, which apparent size (mostly in width dimension) is larger than their physical (up to by factor 2). Some detection results from the MR satellite images are demonstrated in Figure 5.5 and in Figure 5.6.

**Table 5.4: Averaged accuracy of estimated parameters of detected vessels from MR validation dataset.**
The values are calculated in accordance to (14)

| | length | beam | heading (0..180)° |
|---|---|---|---|
| Generic cargo | 0.71 | 0.61 | 0.79 |
| Tanker | 0.73 | 0.58 | 0.87 |
| service / tug | 0.61 | 0.34 | 0.51 |
| Small passenger | 0.54 | 0.51 | 0.49 |
| Passenger / ferry | 0.75 | 0.66 | 0.83 |
| Container carrier | 0.83 | 0.69 | 0.81 |
| Fishing | 0.48 | 0.31 | 0.54 |

Due to the similar sizes of Landsat-8 scenes which is in average 16 000 x 16 000 pixels the processing time was measured for the entire image. The average processing time for one image needed by the ImageHandler was 35 seconds and additional 31 seconds needed by the VesselDetector. Nevertheless, for VesselDetector this value should not be treated as an absolute metric, as it highly depends on several unpredictable factors, such as: the number of the detected regions at the global region search stage as well as the number of objects in particular region.

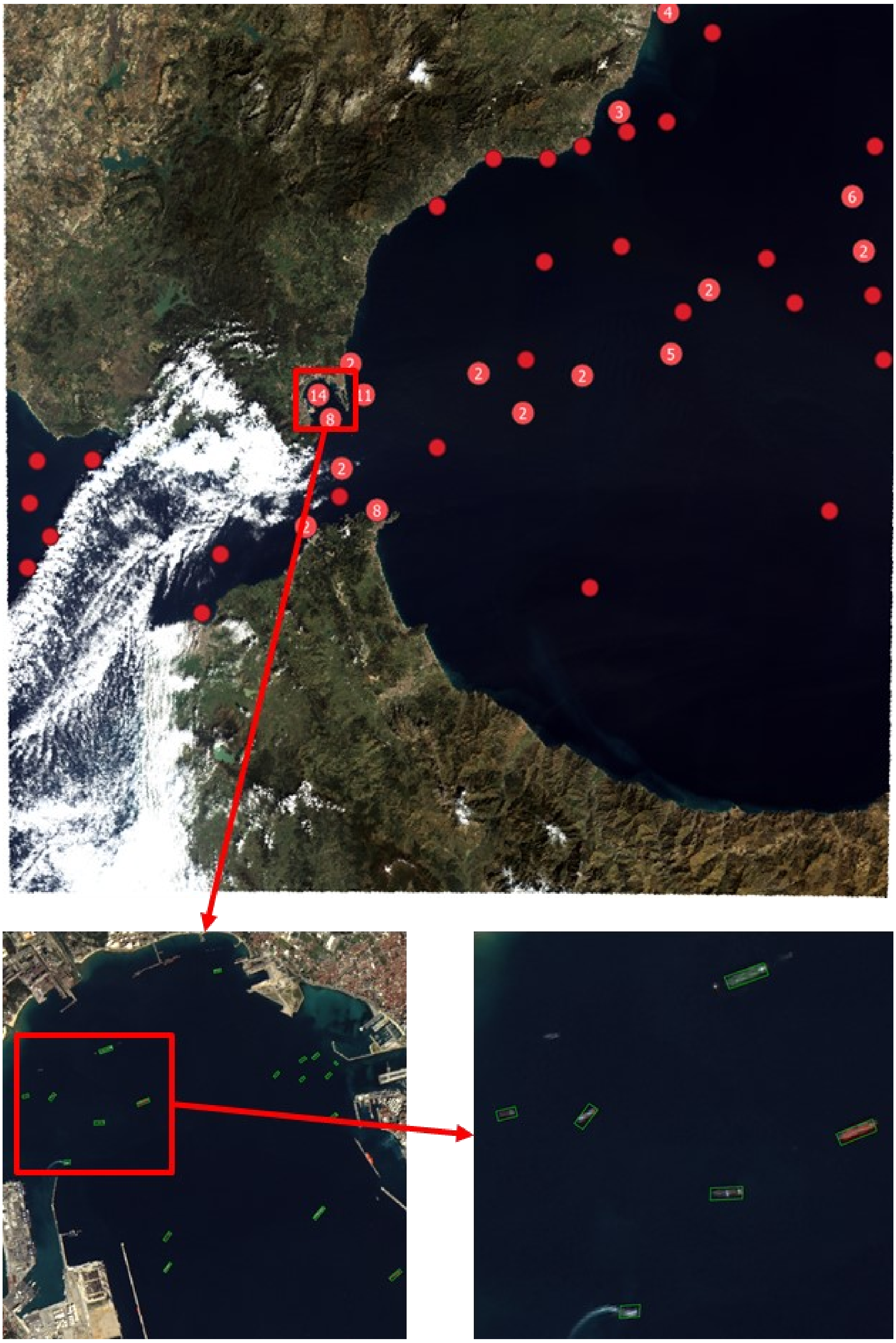


**Figure 5.5: Example of MR vessel detection results: Gibraltar.**
Gibraltar, 2020
Clustered visualizations of detection results. The number in the red circles represents the number of vessels detected at particular image region.
Image credit: Landsat-8 © 2020 USGS

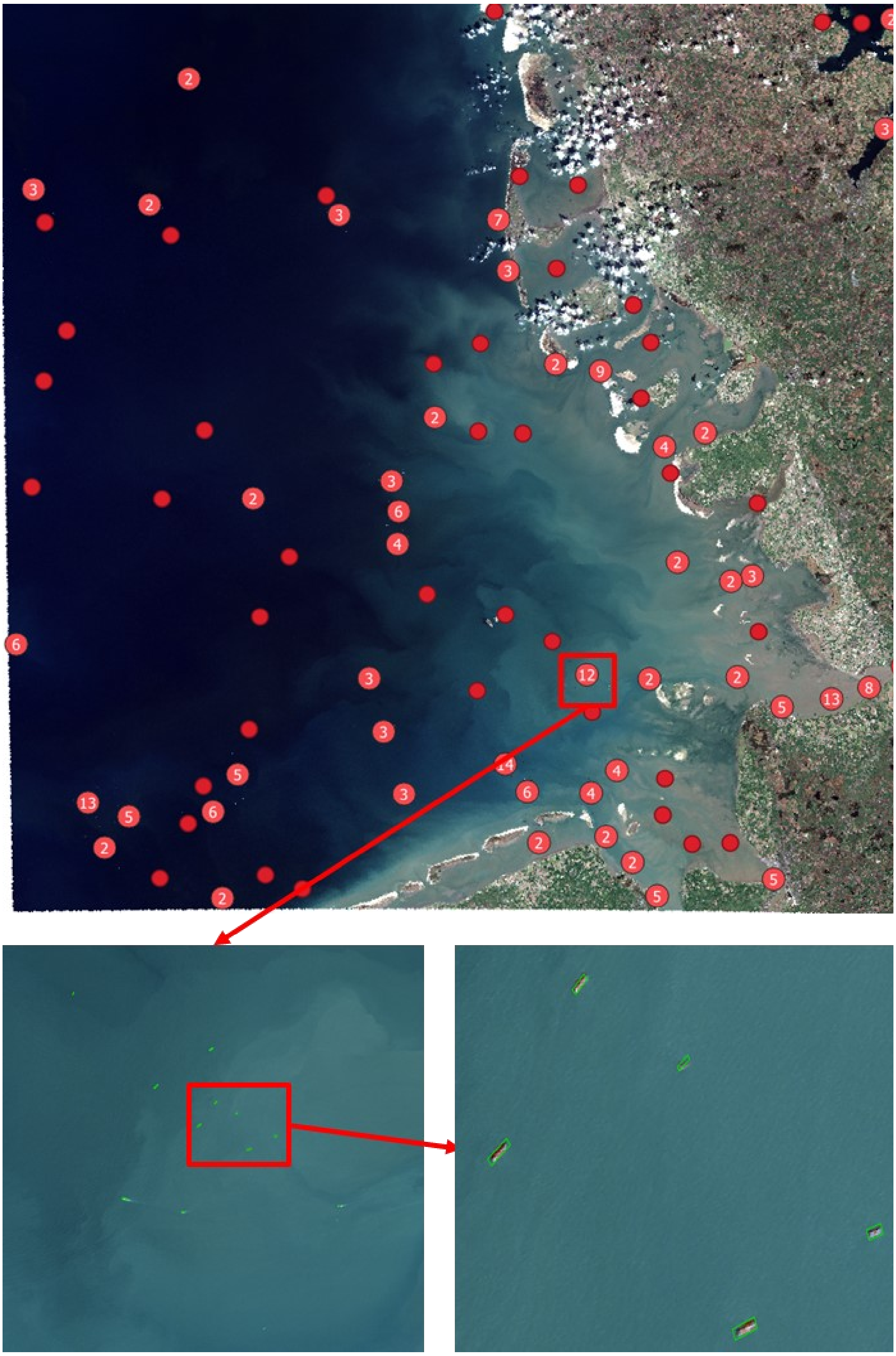


**Figure 5.6: Example of MR vessel detection results: German Bight.**

German Bight, 2020

Clustered visualizations of detection results. The number in the red circles represents the number of vessels detected at particular image region.

Image credit: Landsat-8 © 2020 USGS

# 6 Conclusion

The satellite remote sensing data is valuable source of information for ensuring maritime situational awareness. In particular, the vessel detection product derived from VHR and MR optical satellite sensors can serve as a standalone or a complementary tool for sea traffic monitoring systems.

The goal of this thesis was to develop a method for vessel detection from VHR and MR optical satellite images that would be applicable in real life applications. The presented approach covers the complete processing chain and involves rapid image enhancement techniques, the fusion with automatic identification system (AIS) data, and the detection algorithm based on convolutional neural networks (CNN). Besides the vessel's position and its type, the method allows extracting its dimensions and orientation.

One of the key factors for successful machine learning projects is the availability of the training datasets. This project was not an exception from this rule. To train the CNNs, two versions of training datasets were generated. The VHR training dataset was produced from the set of WorldView-[1-3] and GeoEye-1 images and initially contained about 40 000 of uniquely annotated vessels divided into 14 different classes. The MR training dataset was generated from the set of Landsat-8 images and initially contained about 14 000 of uniquely annotated vessels of 7 different classes.

The presented algorithms are implemented in the form of independent software processors and integrated in an automated processing chain as part of the Earth Observation Maritime Surveillance System (EO-MARISS) [57]. The system has a self-learning design, which means that any successfully processed and validated image will contribute back to the training dataset. It is expected that in the future it will improve accuracies of predictions by the CNNs. In addition, historical datasets are directly used within the detection algorithm in order to extract potentially vessel-containing regions. Furthermore, constantly extending training datasets may offer some new research directions, for example analysis of vessel type dependent traffic patterns and many others.

The processing performance and accuracy assessment conducted on validation datasets showed promising results for maritime NRT surveillance applications, although there are opportunities for improvements. Not surprisingly, vessel detection from MR images has shown lower accuracies of estimated parameters as well as classification scores. This is due to the significantly lower image resolution. A similar situation exists with VHR based scenario in dense harbor areas with high concentration of small boats. One probable research topic in the direction to solve this problem may be selective image resampling to higher resolution employing the CNNs. Further research topics may include experiments with different CNN architectures as well as modelling additional parameters. For example, vessel speed estimation based on dependencies of its type, size and apparent wake patterns.

The solution presented in this thesis has proven its usability within different projects and is used and further developed at the ground station of the German Aerospace Center (DLR) in Neustrelitz.

# List of figures

# List of tables

# References

[1] Transport and Trade Faciliation Series No. 13, "Digitalizing The Port Call Process," United Nations, Geneva, 2020.

[2] L. Feldt, P. Roell and R. D. Thiele, "Maritime Security – Perspectives for a Comprehensive Approach," *ISPSW Strategy Series: Focus on Defense and International Security,* no. 222, 2013.

[3] European Commission / Joint Research Centre, "Working Document III on Maritime Surveillance Systems," European Commission, Ispra, 2008.

[4] EMSA, "CleanSeaNet," [Online]. Available: http://emsa.europa.eu/csn-menu.html. [Accessed 27 01 2020].

[5] EMSA, "Copernicus Maritime Surveillance Service," [Online]. Available: http://www.emsa.europa.eu/copernicus.html. [Accessed 27 01 2020].

[6] S. Brusch, S. Lehner, T. Fritz, S. M., A. Soloviev and B. van Schie, "Ship Surveillance With TerraSAR-X," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 49, pp. 1092-1103, 2011.

[7] B. Tings, C. A. Bentes da Silva, D. Velotto and S. Voinov, "Modelling Ship Detectability Depending On TerraSAR-X-derived Metocean Parameters," *CEAS Space Journal,* vol. 11, pp. 81-94, 2019.

[8] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems 25,* p. 1106–1114, 2012.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," arXiv:1409.4842v1, 2014.

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861v1, 2017.

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," arXiv:1611.10012v3, 2017.

[12] D. Marmanis, M. Datcu, T. Esch and U. Stilla, "Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks," *IEEE Geoscience and Remote Sensing Letters,* vol. 13, no. 1, pp. 105-109, 2016.

[13] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu and U. Stilla, "Semantic Segmentation of Aerial Images with an Ensemble of CNNS," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Prague, 2016.

[14] S. Lars, S. Tobias and B. Jürgen, "Deep learning based multi-category object detection in aerial images," in *SPIE Defense and Security (2017)*, Anaheim, 2017.

[15] Y. Li, . H. Zhang, X. Xue, Y. Jiang and Q. Shen, "Deep learning for remote sensing image classification: A survey," *WIREs Data Mining Knowl Discov. 2018;8:e1264.,* 2018.

[16] K. Rainey, J. D. Reeder and A. G. Corelli, "Convolution neural networks for ship type recognition," in *Proc. SPIE 9844, Automatic Target Recognition XXVI*, Maryland, United States, 2016.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV,* vol. 15, no. 3, p. 211–252, 2015.

[18] T. Yamamoto and Y. Kazama, "Ship detection leveraging deep neural networks in WorldView-2 images," in *Image and Signal Processing for Remote Sensing XXIII*, 2017.

[19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.

[20] Y. Yao, Z. Jiang, H. Zhang, D. Zhao and B. Cai, "Ship detection in optical remote sensing images based on deep convolutional neural networks," *Journal of Applied Remote Sensing,* vol. 11, no. 4, 2017.

[21] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv:1506.01497v3, 2016.

[22] S. Nie, Z. Jiang, H. Zhang, B. Cai and Y. Yao, "Inshore Ship Detection Based on Mask R-CNN," in *2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, 2018.

[23] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017.

[24] N. Bodla, B. Singh, R. Chellappa and L. S. Davis, "Soft-NMS — Improving Object Detection with One Line of Code," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.

[25] D. Štepec, T. Martinčič and D. Skočaj, "Automated System for Ship Detection from Medium Resolution Satellite Optical Imagery," in *OCEANS 2019 MTS/IEEE SEATTLE*, Seattle, WA, USA, 2019.

[26] Kaggle, "Airbus Ship Detection Challenge," [Online]. Available: https://www.kaggle.com/c/airbus-ship-detection.

[27] International Maritime Organization (IMO), "Automatic Identification Systems," 2020. [Online]. Available: http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx.

[28] International Maritime Organization (IMO), "SOLAS Chapter V - Safety of Navigation," IMO, 2002.

[29] W. G. Rees, Physical Principles of Remote Sensing, Cambridge University Press, 2012.

[30] OpenStreetMap, 2019. [Online]. Available: https://www.openstreetmap.org.

[31] A. Pleskachevsky, S. Jacobsen, B. Tings and E. Schwarz, "Estimation of Sea State from Sentinel-1 Synthetic Aperture Radar Imagery for Maritime Situation Awareness," *International Journal of Remote Sensing,* vol. 40, pp. 4104-4142, 2019.

[32] E. F. Vermote, D. Tanre and J. L. Deuze, "Second Simulation of the Satellite Signal in the Solar Spectrum 6S: An Overview," *IEEE Transactions onGeoscience and Remote Sensing,* vol. 35, no. 3, pp. 675-686, 1997.

[33] R. Richter, "A Spatially Adaptive Fast Atmospheric Correction Algorithm," *International Journal of Remote Sensing,* vol. 17, no. 6, pp. 1201-1214, 1996.

[34] P. S. Chavez, "Image-Based Atmospheric Corrections. Revisited and Improved," *Photogrammetric Engineering & Remote Sensing,* vol. 62, no. 9, pp. 1025-1036, 1996.

[35] J. A. Sobrino, J. C. Jiménez-Muñoz and L. Paolini, "Land surface temperature retrieval from LANDSAT TM 5," *Remote Sensing of Environment,* vol. 90, no. 4, pp. 434-440, 2004.

[36] Y. J. Kaufman and C. Sendra, "Algorithm for automatic atmospheric corrections to visible and near-IR satellite imagery," *International Journal of Remote Sensing,* vol. 9, no. 8, pp. 1357-1381, 1988.

[37] S. Liang and J. Wang, Advanced Remote Sensing : Terrestrial Information Extraction and Applications, Elsevier Science & Technology, 2020.

[38] OpenStreetMap, 2020. [Online]. Available: https://www.openstreetmap.org.

[39] D. H. Hubel and T. N. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *The Journal of Physiology,* vol. 160, pp. 106-154, 1962.

[40] D. H. Hubel and T. N. Wiesel, "Receptive Fields and Functional Architecture of Monkey Striate Cortex," *The Journal of Physiology,* vol. 195, pp. 215-243, 1968.

[41] K. Fukushima , "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics,* vol. 36, p. 193–202, 1980.

[42] H. Robbins, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics,* vol. 22, no. 3, pp. 400-407, 1951.

[43] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function," *The Annals of Mathematical Statistics,* vol. 23, no. 3, pp. 462-466, 1952.

[44] L. Bottou, F. E. Curtis and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning," 2016.

[45] TensorFlow, 2020. [Online]. Available: https://www.tensorflow.org/.

[46] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385v1, 2015.

[47] R. Grishick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015.

[48] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," arXiv:1610.02357, 2016.

[49] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv:1502.03167, 2015.

[50] S. Azimi, E. Vig, R. Bahmanyar, M. Körner and P. Reinartz, "Towards Multi-class Object Detection in Unconstrained Remote Sensing Imagery," in *Asian Conference of Computer Vision 2018 (ACCV)*, Perth, Western Australia, 2018.

[51] Z. Zhang, W. Guo, S. Zhu and W. Yu, "Toward Arbitrary-Oriented Ship Detection With Rotated Region Proposal and Discrimination Networks," *IEEE Geoscience and Remote Sensing Letters,* vol. 15, no. 11, pp. 1745-1749, 2018.

[52] L. Li, Z. Zhou, B. Wang, L. Miao and H. Zong, "A Novel CNN-based Method for Accurate Ship Detection in HR Optical Remote Sensing Images via Rotated Bounding Box," arXiv:2004.07124v2, 2020.

[53] J. K. Waters, R. H. Mayer and D. L. Kriebel, "Shipping Trends Analysis," USACE, 2000.

[54] United States. Dept. of the Interior. Office of Economic Analysis, Final Environmental Impact Statement: Deepwater Ports: To Accompany Legislation to Authorize the Secretary of the Interior to Regulate the Construction and Operation of Deepwater Port Facilities, vol. 2, U.S. Department of the Interior, 1974.

[55] Maritime Connector, 2020. [Online]. Available: http://maritime-connector.com/wiki/ships/. [Accessed 25 05 2020].

[56] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing,* vol. 30, no. 1, pp. 32-46, 1985.

[57] S. Voinov, E. Schwarz, D. Krause and B. Tings, "Earth Observation Maritime Surveillance System," in *GeoForum MV 2020 - Geoinformation als Treibstoff der Zukunft*, Rostock, 2020.

[58] M. Boettcher, R. Reißig, E. Mikusch and C. Reck, "Processing Management Tools for Earth Observation Products at DLR-DFD," in *DASIA 2001 Data Systems in Aerospace*, Nice, 2001.

[59] Docker, 2020. [Online]. Available: https://www.docker.com/.

[60] GDAL/OGR contributors, "GDAL/OGR Geospatial Data Abstraction software Library," Open Source Geospatial Foundation, 2020. [Online]. Available: https://gdal.org.

[61] Threading Building Blocks, 2019. [Online]. Available: https://www.threadingbuildingblocks.org/.

[62] Copernicus Space Component Data Access, "Copernicus DEM - Global and European Digital Elevation Model (COP-DEM)," European Space Agency, [Online]. Available: https://spacedata.copernicus.eu/web/cscda/dataset-details?articleId=394198.

[63] German Aerospace Center (DLR), "SRTM X-SAR - Digital Elevation Mode," 2020. [Online]. Available: https://download.geoservice.dlr.de/SRTM_XSAR/.

[64] Copernicus Land Monitoring Service, "EU-DEM," 2020. [Online]. Available: https://land.copernicus.eu/imagery-in-situ/eu-dem.

[65] F. Heymann, T. Noack and P. Banyś, "Plausibility analysis of navigation related AIS parameter based on time series," in *European Navigation Conference*, Vienna, 2013.

[66] OpenCV, 2020. [Online]. Available: https://opencv.org/.

[67] Threading Building Blocks, 2020. [Online]. Available: https://www.threadingbuildingblocks.org/.

[68] NASA, "WorldWind Java," 2020. [Online]. Available: https://worldwind.arc.nasa.gov/java/.

[69] OGC, "Web Map Service," 2020. [Online]. Available: https://www.ogc.org/standards/wms.

[70] Open Source Geospatial Foundation, "GeoServer," 2020. [Online]. Available: http://geoserver.org/.

[71] S. Voinov, D. Krause and E. Schwarz, "Towards Automated Vessel Detection and Type Recognition from VHR Optical Satellite Images," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, 2018.

[72] S. Voinov, "Deep Learning-Based Multiclass Vessel Detection from Very High Resolution Optical Satellite Images," *gis.Science - Die Zeitschrift fur Geoinformatik,* vol. 2020, no. 1, pp. 10-17, 2020.

[73] International Telecommunications Union, "Technical Characteristics for an Automatic Identification System Using Time Division Multiple Access in the VHF Maritime Mobile Frequency Band," 2014.

[74] S. Guadarrama and N. Silberman, "TensorFlow-Slim," 2019. [Online]. Available: https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/slim.

[75] Tensorflow Object Detection API, 2020. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection.

[76] European Space Agency (ESA), "EC unveils new EU maritime policy," 2007. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/EC_unveils_new_EU_maritime_policy.

[77] S. Voinov, E. Schwarz, D. Krause and M. Berg, "Processing framework to support maritime surveillance applications based on optical remote sensing images," in *RSCy2018*, Paphos, Cyprus, 2018.

[78] S. Voinov, F. Heymann, R. Bill and E. Schwarz, "Multiclass Vessel Detection From High Resolution Optical Satellite Images Based On Deep Neural Networks," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, 2019.

[79] S. Voinov, E. Schwarz and D. Krause, "Automated Processing System for SAR Target Detection and Identification in Near Real Time Applications For Maritime Situational Awareness.," in *Maritime Knowledge Discovery and Anomaly Detection Workshop Proceedings*, Ispra, Italy, 2016.

[80] DLR. [Online]. Available: http://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-5515/9214_read-17716/.

[81] S. Guadarrama and N. Silberman, "TensorFlow-Slim," 2020. [Online]. Available:

https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/slim.

[82] H. Kaiming, G. Gkioxari, D. Piotr and G. Ross, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017.

[83] Geospatial Data Abstraction Library, 2019. [Online]. Available: http://gdal.org/.

[84] Tensorflow Object Detection API, 2019. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection.

# Publication Record

**Voinov, S. and Schwarz, E. and Krause D. and Tings, B. (2020):** "Earth Observation Maritime Surveillance System.", In: *GeoForum MV 2020 - Geoinformation als Treibstoff der Zukunft*, Rostock, 2020.

**Voinov, S. (2020):** "Deep Learning-Based Multiclass Vessel Detection from Very High Resolution Optical Satellite Images.", In: *gis.Science - Die Zeitschrift fur Geoinformatik,* vol. 2020, no. 1, pp. 10-17, 2020. ISSN: 18699391

**Voinov, S. and Heymann, F. and Bill, R. and Schwarz, E. (2019):** "Multiclass Vessel Detection From High Resolution Optical Satellite Images Based On Deep Neural Networks.", In: IGARSS 2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 166-169, 28 July-2 Aug. 2019, Yokohama, Japan. DOI: 10.1109/IGARSS.2019.8900506 ISBN 978-1-5386-9154-0 ISSN 2153-7003

**Tings, B. and Bentes da Silva, C. A. and Velotto, D. and Voinov, S. (2019):** "Modelling Ship Detectability Depending On TerraSAR-X-derived Metocean Parameters.", In: CEAS Space Journal, 11 (1), pp. 81-94. Springer. DOI: 10.1007/s12567-018-0222-8 ISBN (Online ISSN 1868-2510) ISSN 1868-2502

**Tings, B. and Velotto, D. and Voinov, S. and Pleskachevsky, A. and Jacobsen, S. (2019):** "Comparison of detectability of ship wakes between C-Band and X-Band SAR.", In: TerraSAR-X/TanDEM-X Science Team Meeting, 21.-24. Okt. 2016, DLR Oberpfaffenhofen, Germany.

**Tings, B. and Velotto, D. and Voinov, S. and Pleskachevsky, A. and Jacobsen, S. (2019):** "Non-Linear Modelling of Detectability of Ship Wakes and Comparison between X-Band and C-Band SAR.", In: TerraSAR-X/TanDEM-X Science Team Meeting, 21.-24. Okt. 2016, DLR Oberpfaffenhofen, Germany.

**Voinov, S. and Krause, D. and Schwarz, E. (2018):** "Towards Automated Vessel Detection and Type Recognition from VHR Optical Satellite Images.", In: IGARSS 2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 4823-4826., 22-27 July 2018, Valencia, Spain. DOI: 10.1109/IGARSS.2018.8519121 ISBN 978-1-5386-7150-4 ISSN 2153-7003

**Schwarz, E. and Berg, M. and Krause, D. and Voinov, S. (2018):** “Near Real Time Processing Framework for Remote Sensing Based Maritime Surveillance Applications.”, In: 69th International Astronautical Congress, IAC 2018, 01.-05. Okt. 2018, Bremen, Germany.

**Voinov, S. and Schwarz, E. and Krause, D. and Berg, M. (2018):** “Processing framework to support maritime surveillance applications based on optical remote sensing images.”, In: Proceedings of SPIE - The International Society for Optical Engineering, 10773, pp. 1-6. SPIE. Sixth International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2018), 26-29 March 2018, Paphos, Cyprus. DOI: 10.1117/12.2326058 ISBN 978-151062117-6 ISSN 0277786X (Best paper award)

**Krause, D. and Schwarz, E. and Voinov, S. and Damerow, H. and Tomecki, D. (2018):** “Sentinel-1 near real-time application for maritime situational awareness.”, In: CEAS Space Journal, pp. 1-9. Springer. DOI: 10.1007/s12567-018-0210-z ISSN 1868-2502

**Schwarz, E. and Voinov, S. and Frauenberger, O. and Krause, D. and Tings, B. (2018):** “Remote Sensing Analysis Framework for Maritime Surveillance Application.”, In: CMRE Maritime Big Data Workshop, pp. 1-6. Maritime Big Data Workshop CMRE, 09.-10. Mai 2018, LaSpezia, Italien.

**Voinov, S. and Schwarz, E. and Krause, D. and Berg, M. (2018):** “Identification of SAR Detected Targets on Sea in Near Real Time Applications for Maritime Surveillance.”, In: Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings, 16 (1), pp. 39-48. ScholarWorks@UMass Amherst. Free and Open Source Software for Geospatial (FOSS4G) Conference, 24.-26. Aug. 2016, Bonn, Deutschland. DOI: 10.7275/R5125Q

**Schwarz, E. and Voinov, S. and Krause, D. and Daedelow, H. and Tings, B. (2017):** “Applications for Maritime Situational Awareness.”, In: Asian Space Technology Summit, Kuala Lumpur, Malaysia.

**Schwarz, E. and Krause, D. and Voinov, S. and Daedelow, H. and Tings, B. and Pleskachevsky, A. and Singha, S. and Jacobsen, S. (2017):** “Applications for Maritime Situational Awareness.”, In: C-SIGMA VII, 19.-20. Apr. 2017, Lissabon.

**Schwarz, E. and Krause, D. and Voinov, S. and Daedelow, H. and Jacobsen, S. and Tings, B. (2016):** “Research and pre-operational application of near real time services for maritime situational awareness.”, In: TerraSAR-X/TanDEM-X Science Team Meeting 2016, 17. - 20. Oct 2016, Oberpfaffenhofen, Deutschland.

**Daedelow, H. and Schwarz, E. and Voinov, S. (2016):** “Near Real Time Applications to retrieve Wind Products for Maritime Situational Awareness.”, In: DLRK 2016, 13.-15. September, Braunschweig.

**Voinov, S. and Schwarz, E. and Krause, D. (2016):** “Automated Processing System for SAR Target Detection and Identification in Near Real Time Applications For Maritime Situational Awareness.”, In: Maritime Knowledge Discovery and Anomaly Detection Workshop Proceedings Publications Office of the European Union. pp. 66-68. ISBN 978-92-79-61301-2.

**Schwarz, E. and Krause, D. and Voinov, S. and Daedelow, H. and Lehner, S. (2016):** “Near Real Time Applications for Maritime Situational Awareness.”, In: Fourth International Conference on Remote Sensing and Geoinformation of Environment (RSCY2016), 04.-08. Apr. 2016, Paphos, Cyprus.

**Tings, B. and Bentes da Silva, C. A. and Frost, A. and Velotto, D. and Voinov, S. and Wiehle, S. (2016):** “NRT Vessel detectability on TerraSAR-X and Sentinel-1.”, In: TerraSAR-X/TanDEM-X Science Team Meeting, 17.-20. Okt. 2016, DLR Oberpfaffenhofen, Germany.

**Schwarz, E. and K., Detmar and Daedelow, H. and Voinov, S. (2015):** “Near Real Time Applications for Maritime Situational Awareness.” In: Deutscher Luft- und Raumfahrtkongress 2015, 22. Sep. - 24. Sep 2015, Rostock, Deutschland.

**Keil, M. and Esch, T. and Divanis, A. and Marconcini, M. and Metz, A. and Ottinger, M. and Voinov, S. and Wiesner, M. and Wurm, M. and Zeidler, J. (2015):** Updating the Land Use and Land Cover Database CLC for the Year 2012 - „Backdating“ of DLM-DE of the Reference Year 2009 to the Year 2006. Project Report. Umweltbundesamt, Dessau-Roßlau, Germany.

**Spivak, L. and Spivak, I. and Sokolov, A. and Voinov, S. (2014):** “Comparison of Digital Maps: Recognition and Quantitative Measure of Changes.” In: Journal of Geographic Information System, 6 (5), pp. 415-422. Scientific Research Publishing. DOI: 10.4236/jgis.2014.65036 ISSN 2151-1950

**Voinov, S. (2014):** “Modeling Population Distribution Based on EO-Derived Data on the Built-Environment.” Master's thesis, Hochschule für Technik Stuttgart.