TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

LEVERAGING MACHINE LEARNING FOR MARITIME OBJECT DETECTION AND PEATLAND CLASSIFICATION

Harnessing the Power of Machine Learning for Precise Maritime Object Detection and Peatland Classification

Luca Zelioli

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 42 | TECHNICA – INFORMATICA | TURKU 2024

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

LEVERAGING MACHINE LEARNING FOR MARITIME OBJECT DETECTION AND PEATLAND CLASSIFICATION

Harnessing the Power of Machine Learning for Precise Maritime Object Detection and Peatland Classification

Luca Zelioli

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 42 | TECHNICA – INFORMATICA | TURKU 2024

University of Turku

Faculty of Technology
Department of Computing
Computer Science
The Doctoral Programme in Technology

Supervised by

Professor, Jukka Heikkonen
Research director
Department of Computing
University of Turku

Docent, Fahimeh Farahnakian
Supervisor
Department of Computing
University of Turku

Reviewed by

Professor, Tapio Elomaa
Department of Computer Science
University of Tampere

Assistant Professor, Jaakko Suutala
Faculty of Information Technology and Electrical Engineering
University of Oulu

Opponent

Professor, Heikki Kälviäinen
Department of Computational Engineering
Lappeenranta-Lahti University of Technology LUT

ISBN 978-951-29-9796-1 (PRINT)
ISBN 978-951-29-9797-8 (PDF)
ISSN 2736-9390 (PRINT)
ISSN 2736-9684 (ONLINE)
Painosalama Oy, Turku, Finland, 2024

My dedication goes to my family and to the University of Turku.

Abstract

This thesis deals with two application sectors where Machine Learning (ML) approaches have a central role. The first one is the maritime environment, where one of the tasks is to create the Situational Awareness (SA) model, showing what is happening in the environment around a vehicle. The second sector focuses on peatland classification to characterize and differentiate various peatland types.

The maritime study proposed in this thesis investigates the most relevant target predictors in the maritime environment, focusing on different Convolutional Neural Network (CNN) architectures. Additionally, Transfer Learning (TL) is implemented to determine if its use enhances object recognition performance. Subsequently, a maritime dataset is developed. The dataset is precisely manually annotated and can be used for two main computer vision tasks: object detection and tracking. The purpose of the dataset is to provide a solid basis for the development of efficient ML-based approaches for SA modeling in maritime environments. Article I evaluates the performance of three state-of-the-art object detection algorithms using datasets collected in the Finnish archipelago. Best method is Faster R-CNN with ResNet101 as feature extractor achieving the highest accuracy at 74.0%. Article II addresses the limited availability of domain-specific datasets in maritime environments. For this purpose a new dataset was tested with various detectors, revealing that Faster R-CNN (35.18%) and EfficientDet (55.48%) achieved the highest average precision. Article III explores the performance of Faster R-CNN, R-FCN, and SSD using different feature extractors, with Faster R-CNN achieving the highest mean average precision at 75.2%.

The objective of the remote sensing part of the thesis is to create and evaluate a methodology that starts from a set of Geographic Information System (GIS) data input and finishes with the output of a soil-type classification map, especially focusing on pixel-wise soil-type classification. The proposed peatland methodologies summarizes the accumulation of decay material. Peatland areas are mainly found where vegetation decomposition exists. Peatland areas help to regulate the vegetation state and water availability. Article IV proposes a CNN fusion approach for peatland site type classification by integrating multi-source and multi-resolution data, achieving an accuracy of approximately 32%. Article V investigates the performance of CNNs

when trained with a high number of synthetic aperture radar (SAR) and visual bands (51.06%) compared when trained with only the best bands (56.73%). Article VI extends the methods used in Articles IV to different zones in Finland, achieving a classification accuracy ranging from 26.9% to 33.6%.

KEYWORDS: Deep learning, object detection, satellites imagery, peatland classification, autonomus driving, sensor fusion, remote sensing.

Tiivistelmä

Tämä työ käsittelee kahta sovellusaluetta, joissa kummassakin koneoppiminen on keskeisessä asemassa. Ensimmäinen on merelliseen ympäristöön liittyvä aihe, jossa yhtenä tehtävänä on luoda tilannetietoisuusmalli (SA -malli), joka esittää merenkulkualuksen ympärillä tapahtuvat kehityskulut ja ilmiöt. Toinen aihe keskittyy turvemaiden luokitteluun, jossa tehtävänä on luonnehtia ja eritellä useita erilaisia turvemaatyyppejä.

Tämän työn merenkulkuun liittyvä tutkimus tutkii mereliseen ympäristöön kaikkein soveltuvimpia kohteentunnistukseen liittyviä ennustusmenetelmiä keskittyen lähinnä erilaisiin konvoluutiohermoverkkojen (CNN) arkkitehtuureihin. Lisäksi toteutettiin siirto-oppiminen sen selvittämiseksi, auttaako se kohteiden tunnistamisen suoritustasoa. Näiden ohessa kehitettiin merenkulkuun liittyvä tietovaranto. Kyseinen datajoukko on annotoitu käsin hyvin tarkasti ja sitä voidaaan käyttää kahteen tietokonäön tehtävään: kohteiden tunnistamiseen ja seuraamiseen. Datajoukon tarkoituksena on tarjota yhtenäinen pohja tehokkaille koneoppimiseen perustuville lähestymistavoille, joilla luodaan SA -malleja meriympäristöä varten.

Artikkeli 1 arvioi kolmea tämänhetkistä huipputasoa edustavaa kohteentunnistusalgoritmia käyttäen Suomen saaristoalueelta kerättyä datajoukkoa. Paras menetelmä on Faster R-CNN, joka käyttää esiopetettua ResNet101 -verkkoa piirreirrotukseen saavuttaen parhaimmillaan luokitustarkkuuden 74.0 %. Artikkeli II käsittelee sovellusaluekohtaisten merellistä ympäristöä koskevien tietovarantojen rajallista määrää. Tästä syystä uutta datajoukkoa kokeiltiin useilla kohteentunnistimilla, jolloin paljastui, että Faster R-CNN (35.18%) ja EfficientDet (55.48%) saavuttivat parhaan keskimääräisen luokittelutarkkuuden. Artikkeli III tutkii menetelmien Faster R-CNN, R-FCN, ja SSD suorituskykyä käyttäen erilaisia piirreirrottimia. Tällöin Faster R-CNN saavutti parhaan keskimääräisen luokittelutarkkuuden 72.5%.

Kaukokartoitusta koskevan osuuden tavoitteena on luoda menetelmäjoukko, joka alkaa geografisen tietämysjärjestelmän (GIS) syötekanavista ja ja päättyy karttaan, joka esittää maaperätyyppien luokittelua, joka keskittyy erityisesti pikselitason suotyyppiluokkiin. Toinen tavoite on näiden menetelmäjoukkojen toiminnallisuuden arviointi. Ehdotetut turvemaita koskevat menetelmät koostavat yhteen maatuvan

aineksen kertymisen. Turvemaat löytyvät pääasiassa sieltä, missä kasvillisuuden maatumista tapahtuu. Turvemaat auttavat säätelemään kasvillisuuden tilaa ja kasvillisuuden vedensaantia. Artikkeli IV ehdottaa lähestymistapaa, jossa käytetään CNN -fuusiota tavalla, jossa yhdistetään monilähteistä, monitarkkuuksista dataa. Tällöin saavutetaan luokittelutarkkuus, joka on noin 32%. Artikkeli V tutkii CNN -menetelmiä, jotka opetetaan hyvin suurella määrällä SAR- ja visuaalisen alueen satelliittilähteiden taajuusalueita (näin saavutetaan 51.06% luokittelutarkkuus), ja näitä verrataan opettamiseen vain parhailla taajuusalueilla (56.73%). Artikkeli VI laajentaa artikkelissa IV käytetyt menetelmät eri alueille Suomea. Tällöin saavutetaan luokittelutarkkuus, jonka vaihteluväli on 26.9 ... 33.6%.

KEYWORDS: koneoppiminen, kohteentunnistus, satelliittikuvantaminen, maaperän luokittelu, autonominen ajo, anturifuusio, kaukokartoitus.

Acknowledgements

I would like to express my heartfelt gratitude to my two advisors, Professor Jukka Heikkonen and Dr. Fahimeh Farahnakian, who have played an indispensable role in the competition of my study. I am immensely thankful for their constant encouragement throughout this research exploration. Their membership and their guidance have shaped my knowledge of computer science and, in particular, in Computer Vision. Without their precious advice, I did not really know if I would have survived this journey. My special thanks go to Professor Olli Nevalainen (UTU) who helped me in my thesis evaluations.

My studies were not possible without the help of my friend Dr. Jonne Pohjankukka, who helped me when I needed it most. He guided me through the most ambitious projects of my entire study, The Maati project, together with GTK and LUKE institutes.

There are several researchers I had the honour to work with. These are Doctors Maarit Middleton (GTK), Paavo Nevalainen (UTU), and Sakari Tuominen (LUKE);. Thanks to their expertise, I grew professionally and for their insightful feedback and constructive criticism, which greatly enriched the quality of my study.

I extend my sincere appreciation to the staff and faculty of the Department of Computing of the University of Turku, whose support in various administrative and academic matters facilitated a conducive research environment.

My special thanks go to my friends Pouya Jafar Zadeh and Adrian Borzyszkowski who provided stimulating discussions and shared resources that made the research journey more rewarding.

I can not close this acknowledgement without remembering Professor Johan Lilius, who introduced me to the academic world. Without his contribution and his help, I never took the first steps in my journey.

I would like to express my gratitude to my girlfriend, Anita Ylitalo, my family, Rosella, Paolo, Mauro, and Santa Caterina's friends for their unwavering support,

patience, and encouragement. Your belief in my abilities has been a constant source of motivation. I could not have done this without you.

Grazie.

Luca Zelioli
Eura, Finland
May 2024

Table of Contents

Abbreviations

RS	Remote Sensing
ML	Machine Learning
SA	Situational Awareness
CNN	Convolutional Neural Network
GIS	Geographical Information System
PR	Pattern Recognition
PCA	Principal Component Analysis
ICA	Independent Component Analysis
EEG	Electroencephalogram
ECG	Electrocardiogram
GLCM	Grey-Level Co-occurrence Matrix
SOM	Self Organizing Maps
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
MLP	Multilayer Perceptron
CART	Classification and Regression Trees
SS	Selective Search
SSD	Single Shot Detectors
MAD	Mean Absolute Deviation
MAE	Mean Absolute Error
MSE	Mean Squre Error
FLIR	Forward Looking Infrared
IR	Infrared
LSTM	Long Short Term Memory

List of Original Publications

This dissertation is based on the following original publications, which are referred to in the text by their Roman numerals:

I Soloviev, V., Farahnakian, F., Zelioli, L., Iancu, B., Lilius, J. and Heikkonen, J., Comparing CNN-based object detectors on two novel maritime datasets, in IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1-6, 2020

II Iancu, B., Soloviev, V., Zelioli, L. and Lilius, J., ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations, in Remote Sensing, pp. 13(5), 2021.

III Farahnakian, F., Zelioli, L. and Heikkonen, J., Transfer Learning for Maritime Vessel Detection using Deep Neural Networks., in IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 1-6, 2021.

IV Farahnakian, F., Zelioli, L., Pitkänen, T. Pojankukka, J., Middleton, M., Tuominen, S., Nevaleinen, P. and Heikkonen, J., Multistream Convolutional Neural Network Fusion for Pixel-wise Classification of Peatland, in 26th International Conference on Information Fusion (FUSION), pp. 1-8, 2023.

V F. Farahnakian, L. Zelioli, M. Middleton, I. Seppä, T. P. Pitkänen and J. Heikkonen, CNN-based Boreal Peatland Fertility Classification from Sentinel-1 and Sentinel-2 Imagery, 2023 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Tokyo, Japan, 2023, pp. 1-7.

VI Zelioli, L., Farahnakian, F., Middleton, M., Pitkänen, T., Tuominen, S., Nevaleinen, P., Pojankukka, J. and Heikkonen, J., Peatland Pixel-level Classification via Multispectral, Multiresolution and Multisensor data using Convolutional Neural Network, Elsevier Remote Sensing Environment, 2024 (submitted).

1 Introduction

We live in a world dominated by data. Data is currently provided by numerous different types of devices, such as smart televisions, cookers, cameras, mobile phones, and laptops. These devices collect data with minimal human intervention. Machine Learning (ML) has a central role in Artificial Intelligence (AI). The ML methods are designed to process large datasets, extract, and learn to recognize patterns in data to provide accurate predictions or classifications [1].

The proper manipulation of the datasets produces information used for various purposes, such as real time decision-making. The correct data handling methodologies are key elements when building efficient ML applications. Thanks to low-cost internet access, information can be available everywhere, and the continuous process of data collection makes the data practically an inexhaustible resource. However, in some cases data access is limited due to privacy issues or rules. On the other hand, open data has the idea of free usage without restrictions. Using data correctly helps the user to predict future outcomes; therefore, people or machines are able to make better decisions. In addition, it is crucial to ensure good quality of the data; working with bad data tends to give wrong or inaccurate predictions or classifications.

AI applications, and especially ML rely on datasets [2], and due to the types of the available data, ML provides two different main types of algorithms. If the dataset comes with labels, *supervised learning* [3] assigns for each data point a label. If the labels are not provided with the dataset, *unsupervised learning* [4], can undercover patterns and structures from the data. ML systems perform well when they gain the ability of generalizing from the training data and make accurate predictions from unseen data.

Real time decision making, especially in the maritime environment, is becoming more and more vital. If the data provided to a ML system is reliable, the AI system can make decisions more rapidly than humans. Real-time data may be used to generate automated decisions and successively predict action, enhancing Situational Awareness (SA), which broadly refers to the understanding of the environment around the system.

In Remote Sensing (RS) [5], ML is used to fuse multi-spectral, multi-resolution, spatial data [6]. Data are collected from different sources, and they include also the time dimension. One of the common data type in RS is satellite data. Similarly, planes and drones can be used for data generation.

For application purposes, the knowledge provided by the data is a valuable asset. It enables ML to be smarter and build robust models that can be generalized with unseen data. Generalization of the model [7] helps the process to make intelligent decisions. Thanks to various abilities such as real-time decision-making and continuous improvement ML, without any doubts, ensures a positive impact on society. This thesis contributes to the ML based data analysis in two sectors, maritime and remote sensing.

1.1 Motivation and objectives of this research

This thesis focuses on two main topics. SA modelling in maritime environment [8][9][10] and peatland classification [11][12][13] in RS. Predictive SA has the objective of predicting the movements of the recognized objects. This includes object tracking approaches and classification of the observations. In SA, it is crucial to take into consideration the influence of all external factors (weather and light conditions, etc.) during building and evaluation of a predictive model.

Sensor or data fusion [14][15][16] is an essential technology in both applications where it is necessary to combine data from different sources, in order to get information from the environment. The data may come from different RS sources, such as satellites, and aeroplanes or drones.

The research motivation for this thesis originates from the desire to enhance the level of security in the maritime environment. The coastal zone of Finland has a complicated structure with its bays, capes and islands. For this reason, a specialized dataset, e.g. in maritime environment, helps autonomous vehicles to navigate in the Turku archipelago in security, avoiding collisions with other vehicles and with the geographic objects. Thanks to performance analysis of different detectors, it is possible to select which ones are better suited for autonomous navigation.

The land-cover in Finland has various types of fertility levels and soil types. For instance, in those study areas, Keminmaa, Eastern Finland and Southern Ostrobothnia, which are the target pilot areas of this thesis, it was observed that Keminmaa region has different fertility classes compared to other pilot regions, whereas Eastern Finland has poor fertility soil types due to agricultural land development. The Southern Ostrobothnia, instead, is full of abandoned agricultural fields. This already

sets a lot of challenges for RS for soil-type classification tasks.

It is a proper practice to combine different RS data sources in order to understand the varieties of the land types. Institutes such as LUKE[1] and GTK[2] periodically collect data from various sources; it is therefore possible to consequently update the peatland maps.

Maritime study objectives

The main objective of the maritime study proposed in this thesis is to investigate which are the most relevant target predictors in the maritime environment. In the present work we primarily focuses on different Convolutional Neural Network (CNN) models (article I). In addition, transfer learning is studied as well in order to know if its use enhances object recognition accuracy (article III). Subsequently, as a part of this thesis, a maritime dataset is created (article II). The dataset is precisely manually annotated and it can be used for two main computer vision task: object detection and tracking. The objective of the dataset is to provide a solid basis for the development of efficient machine learning based approaches for SA modelling in maritime environments.

Remote sensing study objectives

Usually, the peatland areas are divided into *undrained* portions of the map, which act as a region with poor water levels and *drained*, where the level of water is abundant. In this study, the three pilot areas have between 35 and 39 different soil-type classes (article VI). In particular, the study focuses on combining and normalization of different remote sensing sources with different spatial dimensionalities. This leads to a proposition of a CNN architecture for peatland classification (article IV). Furthermore, article V explores the analysis of two input sources, namely SAR Sentinel-1 and optical Sentinel-2 imagery, employing both early and late fusion techniques. Notably, the article elucidates the disparity in performance between fusion strategies utilizing all bands from Sentinel-1 and Sentinel-2 versus employing only the most salient bands. For comparative purposes, the analysis also includes scenarios where fusion is not performed.

[1] LUKE: https://www.luke.fi/en

[2] GTK: https://www.gtk.fi/en/

1.2 Organization of the thesis

The thesis is organized into two separate parts. Part 1 consists in 6 chapters, and part 2 consists of the 6 original articles. Chapter 1 describes a brief introduction of the thesis and the motivation of the thesis. Chapter 2 includes the theoretical background of pattern recognition. Chapter 3 reports our maritime study, in particular, SA is described. A literature review and a number of maritime dataset are described in this section as well. In Chapter 4, RS application and a description of major satellite data and specifications are given. The chapter also describes the role of CNNs in RS. Chapter 5 gives a summary of included research publications, including the main results and contributions. Chapter 6 outlines the conclusions of this thesis.

2 Pattern recognition

The aim of Pattern Recognition (PR) is to identify patterns and regularities in the data. The identification method varies depending on the application. Typical applications are related to image recognition, audio detection, and signal processing. Patterns and regularities in the data are discovered by statistical analysis of the data combined with ML approaches such as decision trees [17], nearest neighbors techniques [18], or neural networks [19].

Pattern recognition shares many similarities with statistics, and they even share the same objective: understanding the data. However, statistics is more about formal inference. It creates hypothesis and tests it, and pattern recognition is focused on data predictions. *Statistical pattern recognition* uses statistical techniques to learn from observations [20]. These observations are used to create a general model used to recognize unseen samples. The unseen patterns are classified by finding relationships between the new sample and the ones previously observed. These relationships are discovered without committing explicit rules. The role of statistics in statistical pattern recognition is the identification of data regularities in an autonomous manner, without direct human labour. *Syntactic pattern recognition* uses descriptors of the pattern's structure. Descriptors help to create a complex observation structure from multiple simpler patterns. In addition, descriptors have the capacity to describe aspects of the pattern, making the pattern unique. Syntactic pattern recognition is conducted by setting grammar rules that describe the pattern to discover and rule sets that avoid errors and encourage the generalization of the model [21]. Syntactic patterns will be parsed by the model. A common approach is the decomposition of the complex rules in a hierarchical structure.

PR often includes a process that assigns a label to each input value. The label is the one that most probably corresponds to the given input. PR differs from pattern matching (PM), where the algorithm assigns the label to the predicted pattern that exactly matches the input pattern [22]. Pattern recognition algorithms can be trained with labelled data, but they can also exploit features without labels, using unsupervised learning algorithms [23].

Patterns can manifest as objects or events, distinguished by the nature of what they

represent. An *object* refers to a tangible entity characterized by its specific shape, form, and structure. PR techniques are commonly employed to identify and categorize such entities. On the other hand, an *event* denotes a specific occurrence or happening within a defined time-frame, often involving interactions between multiple objects over time. PR techniques are commonly used in multiples sectors such as, detecting car accident, identification of suspicious activities in surveillance cameras. Differences between objects and events are summarized in the Table 1.

Table 1. This table summarize the difference between objects and events in PR.

Feature	Object	Event
What is it	Physical entity	Occurrence or happening
Focus	Shape, form, structure	Interaction, change over time

Objects are various, such as a handwritten number, car, vessel, cat, and dog. Moreover, objects can belong also to specific disciplines, for example, in the medical field, a pattern recognition problem can be the identification of cancer in the human body [24]. In the food quality assessment discipline, a pattern recognition problem can be the identification of traces of metal in the tea [25]. Some of the different types of pattern recognition tasks are summarized in the Figure 1. PR is often integrated with intelligent systems, such as in the case of autonomous vehicles or remote sensing. Autonomous vehicle systems use cameras and possibly other sensors to sense the external environment for detecting objects. In addition, pattern recognition may be related to recognizing events, for example, in human gesture recognition [26]. In some cases, the PR can be simplified by removing the not relevant parts of the current pattern vector. For example, in objects classification, the background of the image is commonly subtracted from the original image in order to better isolate the objects to classify.

Deep learning (DL) can also be used to automatize the pattern recognition problem. The main task of DL is object localization [27] and classification [28]. Currently, one of the most used methods is Convolutional Neural Network (CNN) and its variants. They are based on different types of convolution layers. For example, convolution layers use complex filters to create a feature map of the sensed object. Feature map shows the location and accuracy of the sensed input. Convolution layers automate feature extraction and classification processes. This is especially beneficial in maritime environments, where situational awareness can be determined by sensor fusion. In this thesis, CNN sensor fusion is applied on remote sensing articles number IV, V, and VI.
PR has multiple goals and objectives, such as objects classification and localization, semantic segmentation, and event recognition. Let us assume that the current pattern recognition task is related to image analysis. In object classification, the algorithm

Figure 1. The figure shows some of the different types of pattern recognition tasks. Pattern recognition is able to detect or recognize objects and events as well.

determines the class of an image. Object segmentation is the process where the pixel boundaries of the object are determined to segment the object of interest out of the image. In semantic segmentation, the algorithm predicts the category of each image pixel, dividing the images into multiple classes of segments. Region-based approaches study patterns in images, revealing regions of images containing specific objects. Usually, these regions of importance are surrounded by bounding boxes. In event recognition, complex algorithms investigate the identification of specific sequences of actions, enabling the system to identify event patterns. In addition, the identification algorithm often captures the temporal and spatial information of the event. In real-time applications, event recognition is used in surveillance applications to identify abnormal human behaviour. These tasks are summarized in the Table 2.

Table 2. Different pattern recognition tasks.

Tasks	Solutions
Object classification	The object is classified into a category
Object segmentation	Partition of the image in multiple segments
Region based approach	Object detection with a bounding box
Semantic segmentation	Classification of each pixel to a class
Event recognition	Recognition of events with spatial and temporal information

Pattern recognition systems are normally developed with the following steps: 1) problem formulation, 2) data collection and annotation, 3) initial examination of the

data (e.g. visualization), 4) feature extraction/selection (in classical PR), 5) model building (e.g. based supervised or unsupervised approach), and 6) assessment of the results. The steps are close to CRISP-DM (Cross-industry standard process for data mining) [29] that can be used to guide the pattern recognition system development. The steps of PR are summarized in the Figure 2.

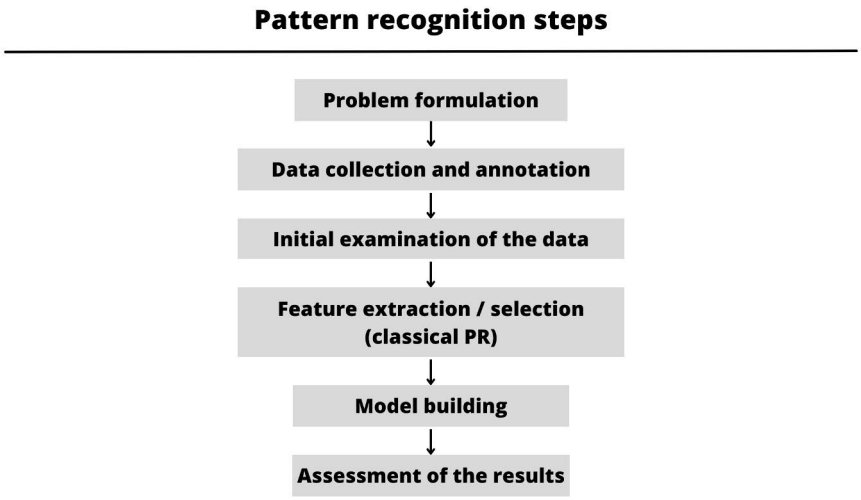

Figure 2. Pattern recognition steps.

The *problem formulation* stage aims to understand which type of investigation should be addressed in order to solve the problem. Usually, this stage is formulated by setting research questions or setting the goals of the functionality of the application. In this stage, problem characteristics are defined, as well as which approaches have been adopted in similar study cases. It is noticeable that a problem statement should always be clear and concise.

In the second step, the *data collection and annotation*, the data that is assumed to solve the problem will be collected. The data can be found from public and already existing datasets or it can be collected by yourself. Already made datasets have the advantage that the labelling process, also known as data annotation, has already been taken care of. However, if the problem to be solved is specific and requires dedicated data collection, the annotation step for ground truth should be taken into consideration. The dataset annotation is the process of assigning a specific value to a label in each collected data point. In article II, a comprehensive data annotation process was conducted. Depending on the type of pattern recognition application, labels can have categorical or numeric values, but they may also contain bounding boxes given by their coordinates. In some cases, the data needs further pre-processing steps. Points

positioned far away from the mean line of all the data points usually produce large errors and should be handled in a special way or eliminated. The authors of [30] suggested the usage of clustering algorithms or minimum volume estimators to handle the population of these so called outliers in the data. The authors also used the Mahalanhobis distance [31] to calculate if a data point has large enough distance from other data to become an outlier.

Data normalization is used to transform data that come from different scale ranges into a common range. In this way, the features are interconnected better discovering of the information is easier when using standardized the values of the data. In articles IV and VI, data normalization was employed to standardize the data obtained from various inputs to a consistent range. This was particularly crucial for derived images like Canopy Height Model (CHM, see section 4.2 for more details) and Digital Elevation Model (DEM, see section 4.2 for more details), where the value range of these inputs typically spans from 0 to 1, while other raster sources such as Sentinel-1 and Sentinel-2 exhibit a value range between 0 and 255 (see Section 4.2).

The *missing data handling* is used in order to reduce the feature errors generated by the training and evaluation of the model. A missing data point can be replaced with the mean or the median of the existing values for this data point in the whole data. The author of [32] describe a phenomenon called the picking value that should be avoided because increasing the size of the feature vector can initially increment the accuracy of the model, but it also raises the probability of the errors. The implementation of the missing raster data will be discussed in article IV, V, and VI.

The initial examination of the data includes commonly statistical measurements are often calculated in this step, such as calculation of the data correlation, which tackles how all the variables in the dataset are related to each others. In addition, the data distributions may be determined, showing how frequently different data points appear in the dataset. Various different plots, such as box plots, scatter plots, and heatmaps are commonly generated at this point because visual representations tend to be more informative than bare numbers.

When the data is visualized, especially in the classical pattern recognition it is the time to extract the features that are relevant for solving the problem. Feature extraction has the purpose to recognise features from the data that are most suitable in order to achieve the best accuracy. These data points identify the best possible combination that is used to identify the various classes in the data; the data points and the classes that have less relevance for the problem are discharged. These important features can be also obtained by the transformation of the original dataset before *feature extraction*.

Table 3. Commonly used visualization techniques in statistical PR.

Type of plot	
Scatter plot	It shows the relationship between two continuous variables
Histogram	It shows the distribution of a single variable
Box Plot	It provides a summary of the dataset
Heatmap	It shows the relation between multiples variables
Line Plot	It visualizes the change over time of a single variable
Bar plot	It is used to display categorical data
Radar chart	It compares multiples variables with multiples category
Density plot	It shows the probability density of a function

The data may be re-projected using dimensionality reduction techniques such as Principal Component Analysis (PCA) [33]. PCA is a linear projection approach that projects the original data along the direction of its maximum variance, thus having the ability to preserve the variance of the data. PCA is often a valuable tool to reduce the dimensionality of the data for further processing.

The model building step includes the determination of the model class to be trained for the pattern recognition task. In some cases, a simple linear model may be sufficient, but most probably, a non-linear model will outperform it. We should also remember that sometimes the pattern recognition tasks may be related to clustering, where the task is to group the data with statistical similarity together. The number of groups in which the data population is separated is usually the number of classes expected, or it can also be arbitrarily decided. If the data come with labels, a supervised classification process is adopted. Moreover, in classification tasks, a totally new input is given to the algorithm, and statistical distance is calculated between each trained input point and the new point. The label of the point with a lesser statistical distance is assigned to the tested point.

Model building is a crucial step, and picking the best model can be a difficult task because each model reacts differently to the data and can produce different results. In model building the generalization of the model aims to determine an optimal model complexity for the task. With the intention of increasing the model generalization as much as possible, ensemble approaches can be helpful [34]. The concept behind the ensemble approach is the combination of multiple models, combining their diversity and improving the performance. In bagging [35], multiple models are trained with the same dataset or with a dataset that is sampled with genetic algorithms [36]. When the training process is complete, in the case of regression, the average accuracy produced by each model is taken as the final result. In the case of classification, the

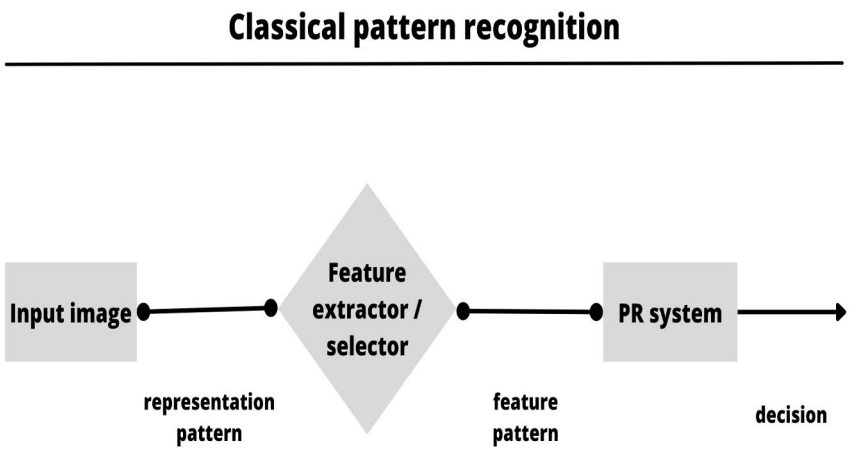

Figure 3. The main steps of classical pattern recognition.

class with the majority of the votes is the predicted class. An example of ensemble approaches is Random Forest [37], which creates multiple bagging decision trees to improve the performance of the models. Another ensemble process is boosting [38], which involves the weighting of samples of each iteration when determining the final output. Another model building technique is the committee approach [39], where multiple models are trained with the same data, and the committee determines the final output, e.g. based on the mean of the outputs that are obtained by a majority vote.

In order to make an *assessment of the performance* of the model, the simplest way to proceed is to divide the labelled data into an actual training and test set. There are also more sophisticated approaches, such as N-fold (see Section 2.4). When the training data has been applied into the model, the test data is used to check its performance. The test set shows how the model performs with data that is not seen by the model during the training phase. Articles IV, V, and VI utilize cross-validation to dynamically create test sets, ensuring robust performance evaluation of the models.

Pattern recognition systems are often divided into the following two categories: 1) classical ones and 2) deep learning based. Classical PR is described in more detail in Section 2.1 and deep learning is Section 2.2. Also, other types of PR system categorizations exist, such as division to statistical and syntactic pattern recognition, see above.

2.1 Classical PR

The classical pattern recognition (CPR) is based on the process shown in Figure 3. First, from the raw data suitable features are extracted. The goals of feature

extractions are 1) to find the most informative data, 2) to remove the redundancy of the data, and 3) to create classes that have small variations between the class members. The classical pattern recognition approach is based on a set of predefined features. Feature engineering is not a straightforward process, and it differs from case to case.

2.1.1 Feature extraction

In CPR the extracted features are often based on the statistical or texture properties of the input. In addition, especially in the case of image-based PR, the features may be based on different spatial filters such as the wavelets [40].

The mean, variance, standard deviation, skewness, kurtosis, and median constitute statistical features that are frequently employed in feature extraction. These features provide information about the distribution type of the input data. A more sophisticated approach to estimate the distribution is through histograms, wherein the selection of bin widths determines the number of features. Histograms convey valuable insights into the spread of values and the range they encompass. In addition, PCA, *Independent Component Analysis* (ICA) [41], and *Fourier Transform* [42] are more sophisticated forms of feature extraction. PCA transforms the original inputs into features, and it keeps only the feature along the direction of its maximum variance, preserving the most important information of the input. The transformation reduces the dimensionality of the data. Close to PCA, the ICA is used to compute a linear transformation of the independent component. The created independent components are statically independent. The primary distinctions between PCA and ICA lie in their respective functions. PCA compresses information while extracting features that exhibit the highest variance. In contrast, ICA focuses on disentangling information into independent components, revealing how features can be perceived as distinct entities. Fourier transform extracts features with the same frequency domain representation. In feature extraction, patterns are grouped by their specific frequency signature. Moreover, this approach may also be used in noise removal, removing unwanted frequency components. In the field of medicine, the Fourier approach can be used to analyse electroencephalogram (EEG) or electrocardiogram (ECG) to discover pattern frequency that can help Doctors formulate better diagnoses [43].

The statistical features, as those described above may be sufficient for simple applications, but often more sophisticated features are needed, e.g. texture features. A texture refers to a pattern or regular shape repetition (of patterns) that appears in an input image. *Texture features* are important in image processing and computer vision applications. Texture features refer to the spatial arrangement and distribution of patterns in an input image. The authors of [44] use texture features to subtract

background from image input data. Within texture, it is possible to derive properties such as a) the contrast that measures the intensity between neighbouring pixels, b) the correlation between different portions of the images, c) how neighbouring pixels are similar, and c) the uncertainty of neighbouring pixels. Two popular texture feature extraction methods are the *Haralick texture features* and the *Local Binary Pattern (LBP)* [45].

The Grey-Level Co-occurrence Matrix (GLCM), also known as Haralick matrix, [46], is used to quantify the texture by calculating the spatial relationship between pixels in an input image. The GLCM reveals spatial relationships between image pixels. To create the GLCM, the input image is first converted to grey-scale, as grey-scale values effectively highlight pixel intensities. Two thresholds are established. The first threshold determines which pixels with the same value ($|\text{pixel1} - \text{pixel2}| \leq$ threshold) are considered neighbours. It is use to determine the maximal absolute difference between two pixels in order to convert them as neighbors to each others. The second threshold dictates the permissible direction of pixel connections. For every pixel in the image, the neighboring pixels are identified at a specific distance and angle, determined by the thresholds. This establishes a set of neighbouring pixels. Subsequently, the co-occurrence matrix is computed by tallying the instances of similar neighbouring pixel pairs from the preceding step. The feature generated by the GLCM can describe the correlation between each image's pixels, energy, entropy, and homogeneity. In remote sensing, Haralick's matrix can be used to determine the type of soil from spectral imagery [46]. The GLCM has been successfully used in several pattern recognition cases, such as breast cancer identification, [47], and in image recognition [48]. From the GLCM one can computed the Run-Length Matrix (RLM) [49], which summarises the number of consecutive pixels with the same value ($|\text{pixel1} - \text{pixel2}| \leq$ threshold) in one of the following directions: vertical, horizontal and obliques. The same two thresholds used in the GLCM are also utilized for this one. In the image converted to grey-scale, for each row and column, the pixels with the same value and direction are counted.

While Harlick's matrix describes global features, LPB is used to represent the local texture of a grey-scale image. The conversion to a grey-scale image helps to discover the intensity of each pixel of the input image. Usually, a threshold is defined, and each pixel inside the circular range is compared within the central pixel. The pixels inside the circular range are compared to the central pixel and marked by a 1 in the case that the value of the pixel is greater than the central pixel, and by 0 bit otherwise. This process is called binary notation. The binary values are then concatenated together and converted in decimal form. The converted values became a local pattern of the central pixel. The same process will be repeated by keeping each different image pixel as central pixel. The local binary pattern of an example

image is shown in the Figure 4. LBP is used due to the computational simplicity of the process. For example, the authors of [50] have used the algorithm to build a robust face recognition algorithm. In [51] the local binary patterns are used to extract features from RS imagery. Both article IV and article V of the present study utilize Local Binary Patterns (LBP) obtained (CHM) and (DEM) rasters (refer to Section 4.2). These rasters offer information of the textures.

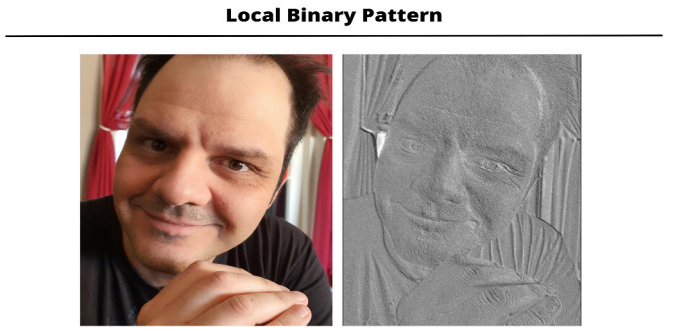

Figure 4. An example of the Local Binary Pattern transformation described above.

Textures can also be identified by analyzing the spatial and frequency characteristics of the input data. To achieve this, tools like filters [52] and wavelets [40] are invaluable. Filters, for instance, such as *Gabor filters* [52], serve to reveal distinctive content in specific directions. These filters excel at capturing features with the same orientations and scales. Gabor filters can even be employed as convolutional kernels [53] to unveil features. As the number of filters applied to an image increases, so does the potential for discovering a greater variety of features. On the other hand, Gabor wavelets [40] are employed to simultaneously capture both frequency and spatial information. Filters are especially effective at emphasizing specific patterns, whereas wavelets excel in revealing the scale and orientation decomposition of an image, capturing spatial frequency and information. The Figure 5 shows an example of Gabor filter application. All feature extraction techniques described in this section are summarized in the Table 4.

2.1.2 Feature selection

The goal of feature selection is to tackle the problem associated with the dimensionality of the data [32]. Feature selection is a process where the original feature space is reduced for revealing a smaller number of relevant features. It helps to increase the accuracy in the training and evaluation of models.
Feature selection can help to better understand the nature of the features. It tries to simplify the training of the model by reducing the model complexity and hence,

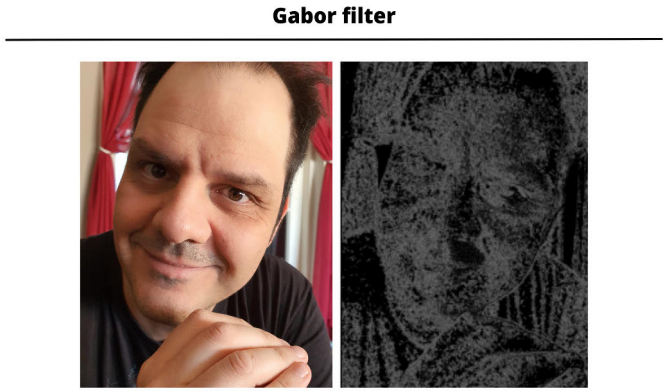

Figure 5. Utilization of Gabor filters. The left image presents the original input. In the right image, Gabor filters are combined with the Canny edge detection algorithm. In this scenario, both algorithms collaborate to amplify the edges and contours within the input image.

hopefully, increase the generalization ability of the model. Feature selection is a combinatorial problem where the task is to select k optimal features out of n features, where $k \leq n$. This leads total of $\Sigma_{k=1}^{n} \binom{n}{k}$ combinations.

Greedy forward selection approach is a feature selection algorithm where a subset of features is selected from the original dataset. It starts with an empty features set and during each iteration of the algorithm, a new feature is added. The added feature is the one that produce best performance when combined with the already selected features. A common performance metric is the model accuracy maximization. The process is repeated until all the features in the original dataset have been selected or when the model does not improve its performance. Article IV and VI uses SFS to reduce the number of input dimensions. In order to find the most relevant features, SFS train all the features one by one and it select the one that maximize the classification accuracy. During the next iteration, the algorithm re-train the selected input from previous iteration with all remaining available inputs. The process is repeated until the input with the best performance is selected. Figure 6 shows an example of greedy forward selection conducted in these articles. In addition [54] uses the greedy forward selection algorithm to select features from a Multi-frequency Single Aperture Radar (SAR) Polarimetric dataset.

To address the combinatorial challenges posed by feature selection techniques, a practical candidate solution could be offered. The *Genetic Algorithms (GA)* [36] use the theoretical backbone of Darwin's theory of natural selection. GA exploits ran-

Table 4. Summary of feature extraction techniques.

Technique		Description
PCA	Pricipal Component analysis	Dimensionality reduction technique
ICA	Independent Component analysis	Components are statistically independent
GLCM	Grey-Level Co-occurrence Matrix	Spatial relationships between pixels
RLM	Run-Length Matrix	Representation of consecutive pixels value
LPB	Local Binary Pattern	Comparison between neighboring pixels
Gabor kernels	Gabor wavelets	Comparison between pixels' spatial frequency

dom search, data mutation and data combination to produce high-quality feature sets. The high-quality features are collected by continuous modification of the feature set. During each iteration of the algorithm, a set of parents is picked (the selection stage). At this point, the data inside each pair of parents is associated together, producing a set of new children (crossover stage). The children solution is randomly mutated (mutation stage), and a fitness score of the new created candidate solution is calculated. If the new candidate solution of data produces acceptable accuracy, then the algorithm makes it a new parent, and the process is repeated until the newly produced candidate solutions do not significantly increase the accuracy of the selected features. These steps are summarized in Table 5. The feature selection techniques analysed in this section are summarized in Table 6.

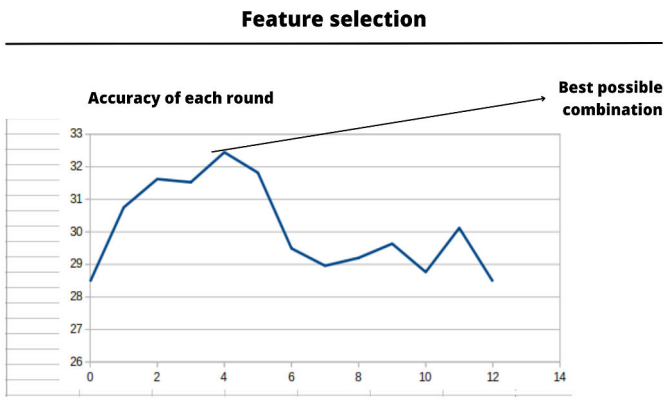

Figure 6. An example of feature selection. Remote Sensing data sources are analyzed to attain the highest possible accuracy using a greedy forward selection algorithm. The image displays the accuracies, obtained using Stratified K-Fold cross-validation, during each iteration of the algorithm. The X-axis represents the number of rounds, while the Y-axis displays the corresponding accuracy. In this specific case, the optimal combination is identified at round number 4. The resulting selected features are then utilized for further analysis. The level of accuracy drops after iteration number 4 because then adding a new feature to the feature set decreases the importance of the whole set.

In the opposite case, the *greedy backward selection*, the feature set is initialized with all the features in the original dataset. In each iteration of the algorithm, the feature that produces the worst performance in terms of accuracy or loss is removed, and the process is repeated until performance improvement stops [55].

The forward and backward feature selection methods can also be used *together*. This process involves recursive iterations of both forward and backward selections until the optimal subset of features has been found. However, this approach is time-consuming and not necessarily the most efficient procedure for feature selection. In the study referenced by [56], the authors employed a forward-backward algorithm to select features from a dataset containing around 500000 samples of Single Nucleotide Polymorphisms (SNPs). Additionally, the computational complexity of the greedy forward selection is lower than that of the backward version of the algorithm, as mentioned in [57]. It is worth of mentioning that both forward and backward selections do not guarantee the finding of an optimal feature set.

Feature selection can be conducted using permutations [58], [59]. The *Permutation* method is used to find all possible relationships between each feature. During each iteration, the algorithm permutes the feature, and calculates the performance of the model trained with all features. Permutation consists in a re-arrangement of the feature dimension, in order to explore different combination, consequently enhancing the performance of the model. A feature is considered important when, after the permutation, the accuracy of the model increases, vice-versa, a feature is considered not important if, after the permutation, the accuracy of the model does not change and it should be ignored. The permutation guarantees that all the relationships between features are exploited. It helps to eliminate hidden relationships in the original features. In addition, the permutation of the features does not require any extra computation. However, if the feature variables are correlated together, permutation can output inaccurate results.

Table 5. The three main stage of a GA algorithm.

Stage of GA algorithm	
Selection	Selection of random data
Crossover	Combination of two or more selected data
Mutation	Application of random changes to the data

2.1.3 Some classical PR methods

While the main focus of the present thesis is the deep learning methods, we shortly discuss some of the most popular CPR methods. This material acts as a background

Table 6. Summary of feature selection techniques used in this present study.

Technique	Description
Greedy forward selection	Features are added one per time until the best combination is found
Greedy backward selection	Features are removed one per time until the best combination is found
Greedy forward-backward selection	A mixture of forward and backward selection techniques
Feature permutation	The relationships between features are exploited
Genetic algorithm	It permutes candidates solutions to find the optimal ones

to the design of deep learning methods which will be the main methods used in article I to VI.

Classical pattern recognition offers unsupervised and supervised algorithms to conduct studies in classification or regression. An unsupervised learning algorithm involves the analysis of the data without labels. Most of the *unsupervised* approaches are based on clustering, where the data with similar characteristics are grouped together, and the data with dissimilar characteristics is ported in different clusters. One of the most popular and classical clustering methods is *K-means clustering* [60]. In this method, the algorithm begins with randomly initialized centroids, and in each iteration, data points are assigned to the cluster with the closest centroid. The distance between data points and centroids is typically calculated using some metrics like Euclidean distance. Additionally, the centroids are updated through the use of means or medians. The process of cluster assignment and centroid updating continues until either the algorithm has iterated a predetermined number of times or the cluster assignments does not change anymore.

The Self Organizing Maps (SOM) [61] algorithm, detects efficiently topographical feature relations from the input. SOM is able to map high-dimensional inputs to a normally 4 or 2 dimensional discrete lattice of units so that inputs that are close in the feature space tend to be represented with nearby in the map space.

The Multilayer Perceptron (MLP), is a supervised pattern recognition neural network. The model maps sets of input data into the proper output. It is a universal approximator [62] due to it continuously approximates a wide range of functions. These functions can exploit the linear or non linear relationship of the data (both input and output), only if the number of given unit and the activation function are correct [63]. A MLP differs from a simple Perceptron network because it is a combination of multiple processing units. Perceptron networks are largely used in binary classification. MLP can have multiple layers between its input and its output layers. These layers are called hidden layers. There are not any rule about how many hidden layers a MLP and the number of neurons in each layer should have. The hidden layer has a minimum of one neuron or unit. A good practice is to test the MLP with differ-

ent numbers of layers and neurons and evaluate the one with the best performance. The neurons are stacked together in order to accumulate more knowledge and accuracy. MLP can use several different activation functions, like Sigmoid, Softmax, Relu, etc., weights are continuously updated layer by layer, and consequently, the prediction accuracy rises. The data should always be normalized to make the data in the same range.

The training of MLP is based on the back-propagation method which works directly with weights updating momentum in a recursively manner. When a weight is updated the error from this update is tested and the error is also computed backward. In MLP each hidden-layer output is fed to the next hidden layer. The back-propagation helps the algorithm to lower the prediction error. In the output layer of the MLP the classification decision is prompted.

Self-Organizing Maps (SOM) differ from traditional pattern recognition algorithms like K-means. In K-means, the nodes operate independently, whereas in SOM, the nodes are closely interrelated [64].

In classical pattern recognition, one of the simplest supervised classification techniques is the *k-Nearest Neighbors (KNN)* algorithm [65]. In KNN, 'k' represents the number of neighbours the algorithm considers when making predictions. This value is a hyper-parameter that significantly impacts the final outcomes of the algorithm. While 'k' can be an arbitrary number, the goal is to adjust it to minimize prediction errors. A special case of KNN is the nearest neighbour (NN), where only the closest sample is determined. KNN supports various distance metrics, including Euclidean, Manhattan, Minkowski, and Hamming distances [66]. This diversity enables the algorithm to be adaptable to different data characteristics and problem domains. NN is employed when dealing with non-linear features and there is a limited number of data points in the training set.

Support Vector Machine (SVM) [67], [68], [69] is a *supervised* pattern recognition algorithm that can be used for classification or regression tasks. The objective of SVM is to find a boundary hyperplane that clearly separates data points. The number of hyperlines is proportional to the number of classes that the SVM algorithm has to separate. The classes are mapped into a space, and SVM calculates a division line that maximizes the distance between each class. The data points on one side of the line will represent one class, the member points on the other side of the line will represent another class, etc.

Decision trees are a family of supervised pattern recognition algorithms. They can learn the hierarchical structure of the input data. Each tree data structure consists

of nodes and branches. Branches are directed connected from a higher level node (parent) to a lower level node (child), and for each node there is exactly one parent. Each tree has a special node called root which does not own a parents. A forest has several root. Every node of the tree stores a feature, and each node defines a unique decision path from the root node that the tree should follow. Decision trees learn by a brute force approach. The algorithm studies and learns recursively each sample in order to build the feature hierarchy. The feature hierarchy determines the importance of each sample. This recursion occurs until all the data points are outside the leaves of the tree.

An example of decision trees is *Classification And Regression Trees (CART)*. The CART can be used for classification and regression tasks. CART provides each node of the tree with a predictor value and a target variable. The algorithm uses the Gini impurity measure [70] to evaluate how much random data are classified wrongly according to the true class distribution of the dataset. However, the Gini impurity measure is not the only possible chaise to adopt for the CART decision tree e.g. mean square error can be used as well.

When multiple random trees are combined, a Random Forest (RF) [71] is created. RF uses bootstrap algorithms [72], such as bagging and boosting, to build the feature hierarchy. Bagging uses sets of trees trained on randomly sampled data. By collecting results from different trees, RF may improve the model's performance for unseen data. Boosting considers a random subset of features contained in every node of the tree, reducing overfitting. In general, RF uses the outcome of multiple decision trees to make predictions. Each decision tree has its classification output. The decision layer of the random forest calculates how many times a class is predicted in each decision tree. Finally, the class with more prediction is the final predicted class.

2.2 Deep Learning

Deep learning is inspired by biological learning [73]. Like classical machine learning approaches, deep learning models can be supervised, or unsupervised or a mixture of both.

The difference between classical machine learning and deep learning is in the feature generation and extraction. In deep learning, feature extraction and detection are tackled directly into the network, and deep learning features are learned by presenting the data to the model. However, this requires much more data than classical machine learning in order to train and evaluate the model.

Object localization and classification are one of the most common deep learning

operations. The architecture used is the *Convolutional Neural Network (CNN)*. CNN has shown a great level of accuracy in discovering features from images. These features are used to classify and localize objects from images. Moreover, CNN can also be used to segment images into regions. Figure 7 shows examples of CNN capability. Images are not the only input that CNN can deal with. Audio classification, time series analysis, and signal data analysis are other tasks that CNN can tackle.

Convolutional Neural Networks contribute to raising the level of accuracy in the maritime environments field. Real-time classification and ship detection are essential components for modern autonomous ships. To guarantee reliable detection and classification of objects, CNN [74], [75] RNN (outside the scope of this thesis) [76] and cGAN (outside the scope of this thesis) [77] are beneficial to enhance the image quality under various weather and lighting conditions. These methodologies can improve the visual quality of images, but at the same time, some of the image features are eventually lost, and the reliability of the deep learning model is reduced.

The anatomy of a CNN can be summarized as follows: there is an input layer and an output layer, and in between, there are one or multiple hidden layers. Hidden layers are composed of a convolution layer and the convolution activator. In addition, a convolution layer can be followed by a batch normalization layer and a pooling layer. A batch normalization layer is used to normalize the data during each computation of the layers CNN. The batch normalization layers increase the training and evaluation speed. The task of the pooling layer is to combine features from a previous layer with the new features from the new layer, Figure 8 The combination can occur using the maximization technique, extracting the max candidate from the feature map, or by averaging, and extracting the average of the feature sets.

At high level, a network comprises two main sections: the feature extraction block and the classification section. In the feature extraction block, features are examined and extracted through a sequence of layers, including convolutional layers, pooling layers, and dropout layers. These layers collectively identify and capture important patterns and characteristics within the input data. Moving to the classification section, a flatten layer is employed to transform the output from the previous layers into a one-dimensional vector. This vector is then passed through fully connected layers that enable the network to learn complex relationships and correlations between the features. Finally, the Softmax layer is used to determine the predicted class, providing probabilities for each class label based on the learned features and relationships. These feature maps are followed by a decision layer that defines the output of the network, which uses fully connected layers to make final decision about classification or object localization. The fully connected layers should have a sufficient amount of neurons to make the final decision. Traditionally, there are multiple ways how

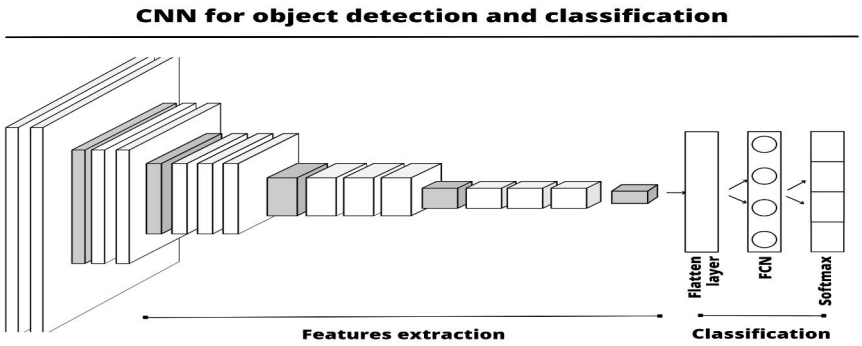

Figure 7. A classical CNN architecture used for classification purposes.

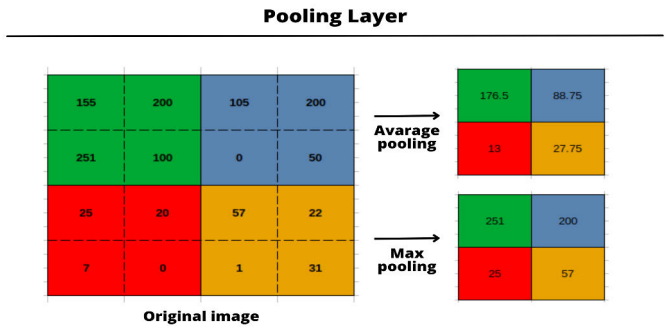

Figure 8. Operation of the Pooling layers which are used to reduce the dimensionality of the feature. The feature reduction occurs with average pooling and max pooling.

a CNN can be applied to pattern recognition problems. These include 1) Selective Search (SS), 2) Sliding Window approach, 3) Region Based approach, and 4) Semantic Segmentation.

In the traditional *Selective Search (SS)* [78], the CNN searches and puts together pixels with the same features (such as colours, textures), and then constructs an initial candidate region. This process is called the bottom-up approach. On the contrary, in the top-down approach, the generated regions are attached together in order to generate an initial hierarchy of candidate regions. During each iteration of the algorithm, the quality of the candidate region is evaluated, and a score is produced. If the score achieves a certain grade of accuracy, the candidate region becomes a *Region Of Interest (ROI)*. All ROIs are evaluated, followed by a creation of a hierarchy of ROIs that are used for further prediction or classification. Moreover, the ROI can also be

compared with the others ROI which are coming from pre-trained model in order to gain more fine-grained prediction.

SS [79] can achieve excellent results, but it is computationally expensive to conduct. Sliding window [80] is another approach that CNN uses to extract ROIs. The working principle of the sliding windows approach is to scan the whole input feature using a window. During every window pass, a process similar to the selective search is conducted. The windows became scored regions of interest. The ones with better score assume the value of objects of interest. In order to achieve better performances and fit objects of different dimensions, the input is re-scaled and windows of different dimensions are applied. The sliding window approach assures that all the parts of the input are covered.

In the region based approaches, traditionally, there are two types of CNN detectors: two-stage detectors and one-stage detectors. In the first family of detectors, the *R-CNN* [81] and the *Faster R-CNN* [82] use selective search to determine object candidates as the first stage of detection. These regions of proposal are sent to the second stage detectors, where a sub-layer classifies the proposed objects, and a second one is responsible of collecting the bounding box coordinates from the proposed region. The R-FCN [83] performs the detection with similarity to the R-CNN family of network. It does not use two different layers for the final classification and bounding box regression, but it uses a fully connected network architecture.

The Fast R-CNN uses traditional methods like SS to discover RPN. In Fast R-CNN [84], the features are extracted by a pre-trained CNN. The Faster R-CNN addresses computational inefficiencies of Fast R-CNN. It is faster because it shares the feature extraction and Region of Proposal Network (RPN) operations. These RPNs are generated within shared convolutional layers. Sharing this process improves the computational speed of the model. Articles I, II and III use states that Faster R-CNN is a solid choice for object detection in maritime environment.

Moreover, in Faster R-CNN, the features are extracted directly from the input image instead of each RPN. The introduction of the ROI pooling that creates fixed size feature maps of consistent size. Both Fast R-CNN and Faster R-CNN share Fully connected layers that exploit the ROIs and generate the object classification, and location passed to a non-maximum-suppression [85] that removes duplicates of bounding boxes. When Faster R-CNN identifies multiple overlapping bounding boxes for a single object, it select the best detection while discarding the others.

Single Stage Detector (SSD) [86] uses a one stage detector to perform object detections and classification. SSD scores the region of interest directly, generating only a

limited number of boxes. In the same time SSD classify the object. Since SSD uses only one stage for detection or classification, it is faster compared to other two-stage detector neural networks. Articles I, II, and III uses SSD to evaluate object detection maritime environment.

EfficientDet [87] is a high-performance and inexpensive computational resource Neural Network. Its backbone EfficientNet, optimizes the model parameters in order to get the best possible accuracy during the training and prediction phase. The models inputs are scaled in width, depth, to achieve better performance in the feature extractions. Article II uses EfficientNet, and it achieves the best average precision with large objects.

Semantic segmentation models [88] are based on *auto-encoders*. Semantic Segmentation is used to give for each pixel of an image with a specific label. Therefore, it classifies each pixel of the input image into provided classes. An example of semantic segmentation can be found in Figure 9.

Auto-encoders [89] are unsupervised learning neural networks that efficiently learn from un-labelled data. Auto-encoder can discover relationships with the data without the usage of the labels. They are mainly formed by an encoder layer, of a deep learning algorithm where the data is reduced in a features map descriptor. Similar to the Unet [90], described in the next paragraph, this reduction operation is executed by blocks of the convolution layers and max-pooling layer. The feature map descriptor is then transferred to a bottleneck layer in charge of understanding the context of the feature descriptor. This layer preserves only all the crucial information. At this point, the noise is filtered as well. Finally, the decoder layer scales up the filtered features, incorporating all the information discovered by the previous layers. Auto-encoders are used mostly in the de-noise operation or in image generators.

A popular semantic segmentation network is the *Unet*, which uses two connected models, an encoder model and a decoder model [90]. It is useful in remote sensing [91]. The Unet is composed of an encoder layer, typically structured with sets of convolution layers and max-pooling layers. These layers aim to capture spatial information of the input image. At this stage, the input dimension is reduced as well. After the encoder layer, a bottleneck composed of multiple convolutional layers captures high-level features from the image. The decoder layer uses sets of up-sample layers and convolution layers to increase the spatial dimension of the features. At this stage, the Unet refines the discovered features. A skip connection section of the network is focused on rebuilding the fine-grained spatial details from the high-level features discovered by the bottleneck. In the end, the output layer produces a segmentation map of the input.

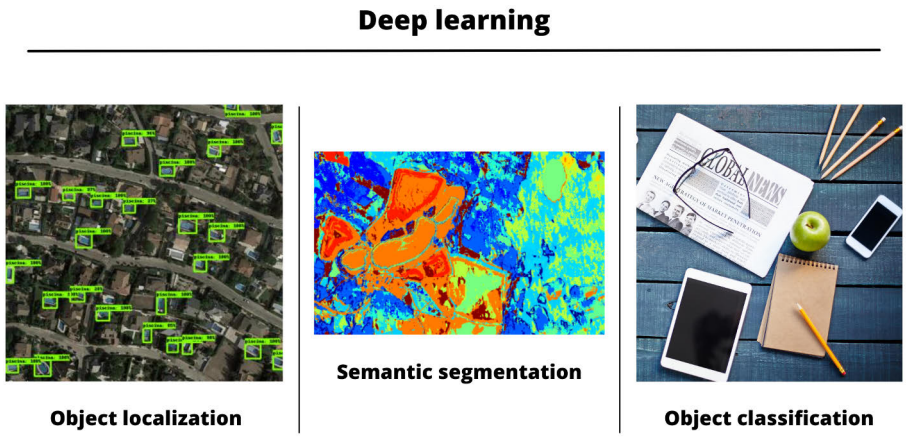

Figure 9. Example of deep learning techniques. In the leftmost image, the CNN performs object detection and localization. The program detects and localizes swimming pools in Madrid. In the second image, there is a pixel-wise classification (semantic segmentation) of Keminmaa region. In the third image, the CNN is in charge of classifying objects.

SegNet [92] is a *Fully Convolutional Neural Network (FCN)* architecture for semantic segmentation. The FCN uses fewer parameters than a traditional convolution layer. The FCN layers utilize a small kernel size to make the feature analysis. The FCN is called fully connected because each input neuron is connected to the output neuron. SegNet uses a similar approach to Unet. The network has an encoder model followed by a decoder model. The task of the encoder model is to discover and map features from the input image. During this process, the input is down-sampled so the FCN layer can use the interconnected neurons and a small kernel to reduce feature mapping uncertainty. Moreover, the decoder branch uses the feature discovered by the encoder to map each pixel provided by the input. SegNet uses the advantage of FCN combined with pooling layers to achieve a concise borderline between objects. During the final pixel map with classes, the input is up-sampled. The up-sampling process also uses the max-pooling layer to avoid over-computation [93].

Generative Adversarial Networks (GAN) [92] has been designed to operate in the cases where it is plausible to de-noise the input data by an auto-encoder and compress the output. The method uses two networks, one against the other one, to create similar input to the ones given. The first network takes random data as input and has the task of generating data similar to the input. The second network works as a discriminator. It takes as input the data generated by the first network and the real data.

If the two inputs disagree, the first network generates more data, and the process is repeated until the discriminator is not able to distinguish the difference between the created data and the original one. During every iteration of the GAN the first network improves the internal knowledge of what is expected to render. The process is called adversarial training because on the first network tries to be better than the second one.

The knowledge provided by trained deep learning models can work also in different domains due to the ability of the deep learning models to generalize. For example, a network trained with a large dataset, such as Microsoft COCO or ImageNet, can be used as a feature extractor for other networks. This process is called *transfer learning*, and it helps to achieve higher model performance than training a model from zero.

In addition, transfer learning techniques can make possible to use the same model in different domains. Within small datasets or small amounts of data, classical pattern recognition achieves better performance than deep learning algorithms. In other hand, transfer learning has several advantages. Transfer learning can also be used to train models from similar domains, saving plenty of computation time and training data. The process works also with domains that share similarities as well.

Another advantage is that transfer learning uses the generalization capabilities of a previously trained model to solve a totally new problem. The general idea behind transfer learning is that the knowledge gained in terms of the weights accumulated during the training with an abundant amount of data will be transferred to another model with a small amount of data. Further, transfer learning can be used as a feature extractor, may be followed by feature selection, where the goal is to obtain a subset of the original features. The process of creating a subset of the original feature is important, especially when it is costly or difficult for the pattern recognition system to extract some of the features by itself. In addition, transfer learning offers the possibility of training a model. Some layers of the pre-trained model can be trained in order to get better accuracy. In some cases, layers in charge of feature extraction can be set as non-trainable due to their already good capabilities to extract features.

Transfer learning can be applied in various areas of pattern recognition, such as natural language processing, computer vision, autonomous vehicles and the healthcare sector [94]. For example, due to the scarce amount of large ship datasets, transfer learning is one efficient way to address the data problem by using the knowledge generated by a general purpose dataset in one specif maritime domain.

2.3 Data augmentation

Both supervised, and unsupervised learning methodologies demand a substantial amount of data. The objective behind data augmentation is to increase the amounts of samples into the dataset through transformative techniques applied to the original data. By using techniques such as random rotation, random flipping, and random zoom, it is possible to enhance existing features. Random flipping and random rotation are used in articles IV, V, and VI.

Random flipping can be conducted in the horizontal and or vertical axes. The flipping mechanism consists of changing the side of the image content. Translation and rotation techniques consist of changing the content of the image in different positions from the original ones. With rotation, the content of the image is rotated in a particular random angle. Thanks to this operation, it is possible to see the content of the image from different positions and angles. It can be very helpful to generalize the deep learning model. For example, in the case of real-time object detection, it is important to detect an object promptly in every possible position. The detection becomes simpler if the same object is trained in different positions. With the scaling algorithm, the features in the image are scaled with different amount in order to make the content of the images bigger or smaller.

Adding noise to the image is another technique that helps the ML algorithm to improve the generalization capabilities. An example technique is so called salt and pepper, where the algorithm randomly adds white or black spots to the image. This technique helps to reduce the blurriness of the image. Brightness and contrast adjustment are used to add or remove light effects from an image. Saturation is used to add or remove the purity of the colours in an images. Colour augmentation changes pixels' colour. These data augmentation techniques are summarized in the Table 7. The authors of [95] uses *Generative Adversarial Network (GAN)* to transfer weather condition from one image to another; this helps to increase the amount of data. GAN techniques are helpful in maritime environments because weather conditions vary often.

2.4 Model building and performance evaluation

Machine learning provides different types of models that can be adopted to solve different types of practical problems. Depending on the type of problem that the model should tackle, linear or non-linear models are adopted. A non-linear model is used when there is a non-linear relationship between the dependent and independent variables. Machine learning model building includes determining the model class and its hyper-parameters for training the model.

Table 7. Some of data augmentation techniques. The techniques highlighted in bold are used in articles IV, V, and VI.

Technique	Description	Type
Flipping	Move the axes of the feature	geometrical
Translation	Move the position of the feature content	geometrical
Scaling	The feature's content is produced in different scale	geometrical
Rotation	The feature are rotated in random angles	geometrical
Salt and pepper	Add noise to the feature	noise
Brightness	Add or remove brightness from feature	noise
Contrast	Add or remove contrast from feature	noise
Color	Change color space from the feature	noise
Saturation	Add or remove saturation from image	noise

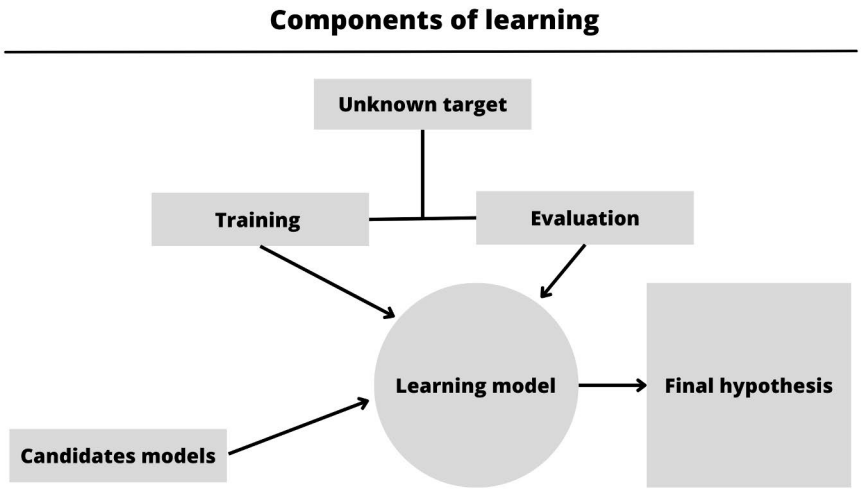

Figure 10. Components of the learning process.

A typical approach to solve a ML problem starts with the formulation of the research question. The process typically starts with a training phase. Evaluation is used to see how well the model is performing. Various candidate models are then commonly employed to address the research question. When a suitable model has been identified, its performance is evaluated with unseen data, leading to the formulation of the final hypothesis. These components of the learning process are summarized in Figure 10.

Learning algorithms

Supervised learning algorithms can extrapolate patterns from specific instances to make predictions about future outcomes. *Unsupervised learning*, on the other hand, focuses on identifying common characteristics within data groups without prior examples. *Reinforcement learning* takes inspiration from the idea of learning through experience, where algorithms learn based on a reward system shaped by positive and negative outcomes. These diverse learning paradigms are summarized in Table 8. It should be noted that the scope of machine learning extends beyond the tasks of pattern recognition. Explanatory modeling utilizes models to shed light on anomalies or phenomena, while predictive modeling yields predictions from input samples.

In *unsupervised learning*, the labels are not provided to the algorithms. In this case, the model is going to learn by itself. These kind of algorithm can interpret the data and create a data hierarchy.

In reinforcement learning, a set of actions is presented to the model. Every action has consequences; good actions will give a good reward to the model, and bad ones will give bad reward. The algorithm learns from a good rewards.

Model training can be done by several different methods, including rote learning, supervised or unsupervised learning and a mix of both. *Rote learning* [96] is a machine learning algorithm where each sample is memorized, and the algorithm will predict the learning input if it matches exactly. If the predicted sample does not match exactly the learned sample, the model will miss-classify the output. The algorithm memorizes new features as they are, storing their characteristics. Due to their nature, rote learning algorithms need plenty of space because every new feature encountered is stored. Algorithms of this class do not have generalization capabilities due to the pattern matching [97]. In addition, the rote learning algorithms do not understand the relationships between features. In order to avoid misunderstanding the relationships between the features, supervised and unsupervised learning algorithms can build a feature relationship hierarchy.

Data types

The data types that the supervised and unsupervised models can handle are manifold. *Categorical data* involves segmenting information into distinct categories, each sample is assigned a new label. Alternatively, *continuous data* assigns numerical values to each sample, accommodating a wide range of possibilities. Data representation extends to objects like digital fingerprints, retinal scans, or handwriting samples.

Moreover, data can capture event registrations, from gesture recognition to the steering directions of vehicles. Even images can be augmented with data, e.g. specific regions can hold labelled objects. In such cases, the label encapsulates pixel values, guiding the model's discovery process.

Table 8. Different types of learning.

Types of learning	
Supervised	Uses labels to train the network
Unsupervised	Only the data is given
Semi-supervised	Mix between labelled and un-labelled data
Reinforcement	Feedback system

ML models have a common denominator. They normally work with large numbers of parameters. There are two main sets of parameters. The parameters of the first type are continuously updated during training. They are typically weights in the model. The second type of parameters is called hyper-parameters, that define the trend of the model in terms of performance.

Hyper-parameter tuning is the search for ideal parameters to get an acceptable performance grade. These parameters are not trainable. A set of hyper-parameters might help a model to enhance the performance, but other settings of the hyper-parameters (resulting in different models) can reduce performance. The set of the parameters varies as well. For example, in the case of image classification conducted with CNN, hyper-parameters fix the number of units of each convolution layer, the learning rate and the optimizer momentum. In RF, the hyper-parameters are the number of leaves or the estimator's number. Hyper-parameter tuning can be executed manually, but more efficient methods exist, such as grid search or random search algorithms [98]. These techniques operate akin to brute force algorithms, systematically testing various combinations of hyper-parameters. The configuration that yields the best performance emerges as the chosen one.

The parameters and hyper-parameters should always be chosen carefully. A poorly conducted selection of parameters can result in overfitting, where the model performs well with the training data but cannot produce accurate prediction with data that was not previously seen during the performance evaluation phases. This phenomenon happens when the model has too much complexity or too many of parameters. When overfitting happens, the algorithm memorizes all the data without understanding the characteristics that help the model generalization. On the contrary, underfitting occurs when the model has a too small set of parameters, and it is impossible to capture all the necessary information needed to understand the underlying hierarchy of the data. A model suffering from underfitting produces poor results both in training and

testing.

Performance evaluation

The evaluation stage estimates the performance of a model. Typically, this involves dividing the dataset into a training set and a test set.The model learns from the training set and then evaluates and makes predictions on the test set. It is crucial to note that the model does not learn from the test set during the training phase; the test set is exclusively reserved for evaluation purposes. Estimating the model's performance can be conducted with a more specific methodology. There are two common methodologies for evaluating the model's performance: *cross-validation* and *bootstraping* [99]. The most common cross-validation techniques are *k*-fold, stratified *k*-fold, and leave one out [99]. Bootstraping algorithms estimate the performance of a dataset by re-sampling the data into subsets. For both cross-validation and bootstrap, the process of randomize and re-sample the dataset help to judge the performance of the model and it estimates how good the model performs.

K-fold cross-validation is used to re-sample the dataset in *k* sub-samples of equal size. The data is randomized in each sub-sample and divided into a training set and a test set. For each fold, the model is trained and evaluated (giving a scope for each evaluation round), and at the end of the *k* iterations, the model is evaluated, calculating the average of the *k* scores.

Moreover, nested *N*-fold cross-validation can adjust the performance of machine learning models and, at the same time, it can optimize the hyper-parameters of the model. It comprises an outer N-fold cross-validation, which divides the dataset into folds, training sets, and test sets. The task of the outer loop is to access the model's performance on a specific fold. At this point, for any iteration of the outer loop, an inner loop with *N*-fold cross-validation has the purpose of tuning hyper-parameters. The advantage of the nested cross-validation is robust estimation of the model capabilities by discovering the best trade-off between the model's performance with unseen data and the tuning of hyper-parameters.

In an unbalanced dataset, some of the classes have many data points, and others have a few. In this case, a special *K*-fold is used. In the *Stratified k-fold*, each subsample of data is produced, preserving the percentage of the class samples. Within Stratified *k*-fold, classes are balanced, and each fold contains samples from all the classes. The authors of [100] propose a statistical approach to select the best possible *v*alue. They state that the choice of *k* parameter should generally keep a low dataset variance. The *k* values are deducted using an algorithm called Complete KCV.

When the choice of k in k-fold cross-validation doesn't provide sufficient confidence in model performance assessment, it is possible to resort to a more exhaustive technique. This involves setting k to the total number of samples in the dataset. This approach, known as *Leave-One-Out Cross-Validation (LOOCV)*, aims to thoroughly validate the model's performance by randomizing the dataset in such a way that each sample takes a turn as the test set. Its strength is the ability to evaluate model performance on every individual sample. This method is particularly beneficial when dealing with a small dataset where obtaining meaningful results from other cross-validation techniques might be challenging due to limited data availability.

Performance metrics

The output of cross-validation should be interpreted with further analysis. Common metrics to evaluate the performance of predictive models include the *Mean Absolute Deviation* (MAD) [101], *Mean Absolute Error* (MAE) [102], *Mean Square Error* (MSE) [103], ROC [104], and *Confusion matrix* [105].

MAD calculates the absolute difference between the actual and predicted value, taking the average from it. The MAD values should low as possible, meaning that the values of the predictions are close to the values of the actual data. MAE measures the difference between the predicted and actual classes but for regression operation. MSE is used in regression tasks to measure the prediction accuracy. The difference between MAE and MSE is that MSE weights better the errors but it is more sensitive to outliers.

Classification generates different types of outcomes to evaluate the performance of the model. These performance metrics compare the predicted class and the actual ground truth. They are summarized in the Table 9. Let us assume that the actual value can be true and false, and the predicted value can be positive or negative. A *True Positive* (TP) case is when the classifier predicts a true class with positive ground truth. In *True Negative* (TN), the predicted class assumes a false value, and the ground truth value is negative. A *False Positive* (FP) case belongs to the case when the predicted class is true, and the ground truth value is negative. A *False Negative* (FN) case occurs when the predicted value is false, and the ground truth is positive.

ROC [106] is a plot that illustrates the performance of a solver for a binary classification problem. However, it can also be used in multi-label classification as one class versus the other classes. The ROC estimates how many times a prediction belongs to the correct class. In binary classification, Sigmoid function gives a value between zero and one where the number expresses how confident the model is for one class or the other. A manually defined threshold determines if the prediction belongs to

Table 9. Measures for evaluating the performance of a classifier.

Model evaluation metrics			
Value		Predicted value	True value
True Positives	TP	true	positive
True Negatives	TN	false	negative
False Positives	FP	true	negative
False Negatives	FN	false	positive

the first or the second class. It works as a discriminator between the two classes. Given different thresholds, the trade-off of the curve can be analysed. The ROC curve may change its shape within the area inside it. The obtained area expresses the performance of the classifier. Moreover, the *Area Under the Curve (AUC)* [107] of ROC measures the separability. Separability measures how much the model is able to distinguish between classes correctly. For example, in binary classification, it tells the user how many time the model predicts class 0 when the real class is 0, and contrary it tells how many times the model predicts class 1 when the real class was 1. In general, the higher the AUC curve runs, the better the model predicts.

It is possible to determine the accuracy of the model and its performance by the means of hypothesis testing. In statistics, null significance of an hypothesis means the hypothesis is without a significant effect or relationship with other hypotheses. This hypothesis can generate so called Type I and Type II errors. These errors are raised when the prediction of the data is incorrect. Type error I occurs when the null hypothesis tested is correct. Moreover, it is possible to reduce the probability of Type I errors. However, it may increase the probability of raising Type II errors when the null hypothesis is false. The Type II error is also called false negative, and it occurs when the assumption of the hypothesis is true, but in reality is false.

Confusion matrices are valuable methods for checking the performance of a classifier. It identifies how well the model performs classification in all classes. Binary classification divides the predicted values based on their true label. In multi-label classification, each row shows the true value of class instances. Each column shows the predicted value. In each cell generated by the combination between rows and columns, the confusion matrix displays the count of each combination between actual and predicted values. With confusion matrices, it is possible to derive other performance indices of the classifier, such as precision, recall and F1 score. The confusion matrix provides not only the accuracy of the model but also the frequency of predictions divided by classes.

The *precision* Equation 1 is used to evaluate the ratio between the TP over predic-

tions. A high precision score indicates that the model can make accurate precision by minimizing FP.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

The *recall* or sensitivity Equation 2 evaluates the true positive predictions over all the actual positive cases included in the dataset. The meaning of high recall value is that the model can properly detect true positive cases.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

The *specificity* Equation 3 measures the ability of the model to correctly identify negative cases among all true negatives case, therefore, it is focused on the minimization of the false positive cases.

$$Specificity = \frac{TN}{FP + TN} \tag{3}$$

The *F1-score* Equation 4 is the harmonic mean between precision and recall. The F1 score shows the balance between precision and recall.

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

In object localization, the *Intersection Over Union* (IOU), represented by equation 5, assesses the model's accuracy in localizing objects within a given input. This metric quantifies the model's performance by computing the ratio of the intersection area between the actual labeled box (typically the manually generated ground truth bounding box aiding the algorithm in object identification) and the model's predicted box, divided by the union area of these boxes. The IOU ranges between 0 and 1, where a value of 0 indicates no overlap, while a value of 1 signifies perfect alignment between the boxes.

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{5}$$

Article I, II, and III uses *Average Precision (AP)* and *Mean Average Precision (mAP)* to compare the performances between detectors.

$$\text{Average Precision} = \sum_{n} (R_n - R_{n-1}) P_n \tag{6}$$

In (6), R_n and P_n are *recall* and *precision* at the *n* IOU threshold.

The average precision summarized in Equation 6 It measures the area under the precision-recall curve. It differ from the F1-score (4) which measures the trade-off between precision and recall across all level of thresholds.

$$\text{Mean Average Precision} = \frac{1}{|C|} \sum_{i=1}^{|C|} \text{AP}_i \tag{7}$$

The mAP Equation in 7 calculate the AP across all the classes *i* of the dataset.

3 Maritime studies

Vehicles in the maritime environment expect a significant level of safety during navigation, docks operations and moor operation. The vessel's crew can not eliminate all the possible risks related to sea navigation. Therefore, modern vessels can count on multiple sensors to augment the level of security and safety on board the ship and around the vessel's environment. AI can increase the level of security by constantly reading and elaborating information about the environment around the vessel. Some of the naval operations can be automatized as well, thanks to the reliability of AI. In addition, while sensors can improve security in the maritime environment, each sensor type has its advantages and disadvantages. One way to increase the usability of the variable data is to adopt *sensor fusion* techniques [108]. The fusion techniques can raise the understanding of the external environment in time and space, to make proper *Situational Awareness* [109][110] (SA).

SA helps to increase the safety of high-risk industrial sectors and consequently to reduce human errors. It implies knowing what is going on around the environment and the implications of a change on it. SA is a concept that is around multiple disciplines with elevated levels of volatility, complexity, and uncertainty, such as autonomous vehicles, aviation, and maritime environments. Information technologies and AI introduce powerful, intelligent systems to tackle the complexity of SA. To exploit the environmental complexity, three stages of SA need to be handled. These are 1) the perception of the environment, 2) the understanding of the situation and 3) the prediction of future outcomes. This thesis focuses on the *perception of the environment* (article I, II), and the *understanding of the situation* (article III).

In the maritime environment, the *Automatic Identification System (AIS)* [111], allows a ship to track the positions of other ships. AIS assists the ship crew in avoiding incidents and also enables maritime surveillance. It keeps track of the position, speed, and direction of the vessel, and is completely automated, and it does not rely on human interactions. In addition, the system can be interfaced with other vessel systems, such as the radar, in order to increase the SA and safety of the ship.

This chapter considers the main concepts and parts of a maritime Situational Awareness system. The first section contains the different technical solutions for getting

perceptional information from the navigation environment, while the second section considers the pattern recognition problem related to the understanding of the situation. Furthermore, the third section describes how predictive models work.

3.1 Perception of the environment

The perception of the environment is a challenging process in developing maritime autonomous devices. There are many different perception sensors that can be used in the maritime environment. Sensors such as RGB cameras, Forward-looking Infrared (FLIR) cameras [112], radar [113], and Light Detection and Ranging (LiDAR) [114] equipment are placed into the vessel in order to sense, track and detect objects around the vessel. Article I and II utilize RGB camera streams for vessel data collection, while articles IV and VI leverage LiDAR data to construct *Canopy Height Models (CHM)* and *Digital Elevation Models (DEM)* rasters.

The main components of an automatic navigation system include commonly visual or camera sensors. The images that cameras provide are challenging due to dynamic background, absence of static points and distance of objects [115]. In addition, weather conditions such as fog or rain can increase the difficulty of the visual sensors to properly detect objects. Lighting conditions, geo-spatial location, motion of sensors and the difficulty to distinguish between sky and sea make the automation of the perception process challenging.

RGB cameras are widely used in autonomous vehicles with the purpose of capturing information from the external environment. RGB cameras sense visible light spectrum (400-700nm) that is converted into electric signals. The combination of the primary light spectrum colours renders images that replicate the human vision. RGB cameras work properly if the viewer has a proper amount of light. However, the light conditions differ from day to night. RGB camera specifications are summarized at Table 10.

Table 10. Typical RGB camera specifications.

Resolution	Frames per second	Pros	Cons
720 pixels 1080 pixels 2k (2040×1080) 4k (3840×2160)	Depends on camera model	cheap good resolution	Depends on lighting visibility

Infrared (IR) cameras [116] use the infrared region of the light spectrum to capture images with scarce lighting. However, IR cameras are not useful when the light

Table 11. Typical IR camera specifications.

Resolution	Frames per second	Pros	Cons
160×120 320×240 384×288 640×480 1024×768	9 to 60 fps	vision in the dark no sun light req.	weather consitions low resolution high price

condition is equal to zero. The specification of the IR cameras are summarized in the Table 11.
FLIR cameras increase the visible spectrum light between 1000nm and 14000nm. FLIR cameras rely on the principle that the higher the temperature of an object is, the more radiation will be emitted. Each pixel of the captured image is assigned a distinct colour based on the intensity of the radiation that it emits, for creating a temperature map of the object. In the maritime environment, FLIR cameras are used because they can detect the differences in heat between the lower temperature of the water and the heat generated by inside vessels' machinery. FLIR cameras can detect objects from wider distances because they need less light than their IR counterpart. Moreover, FLIR cameras can see through the fog until a certain distance, improving the quality of vision in the maritime environment. FLIR camera specifications are in Table 12.

Table 12. Typical FLIR cameras specification.

Resolution	Frames per second	Pros	Cons
320×240 640×480 640×512	from 9 to 1000	works in low light conditions detects temperature's of the objects	high price weather dependence limited resolution

In the maritime environment, *radars* are used to detect, track, and locate objects. Radars send electromagnetic signals, and read back the echoes generated of the objects. Radars send an electromagnetic pulse in some direction. When the electromagnetic pulse hits an object, a slice of the signal is reflected back. Radars can detect objects within a greater distance than cameras. The working principle of radar is simple. The elapsed time between the sending and receiving of the signal is converted into distance and direction. The electronic pulse that the radar uses is unaffected by weather or light conditions. In the maritime environment, radar is effective when visual navigation is restricted by fog or heavy rain. Radar helps to determine the current ship position and the position and cruise of other vessels. They allow a safe passage between obstacles. The pros and cons of Radar system are summarized in

the Table 13.

Table 13. A summary of the advantages and disadvantages of Radar systems.

Advantages	Disadvantages
ability to penetrate fog and cloud	Long operative time
gives position of objects	Multiples false reading
ability to penetrate insulators	Difficulty to distinguish multiple objects
cover wide geographical area	Does not show colors of objects
determine speed of objects	Can not distinguish type of objects

Light Detection and Ranging (LiDAR), works with the same principles as radar, which measures the distance to a target. It produces a 3D points cloud representation of the environment. However, instead of sending and receiving electromagnetic pulses, it uses light. Light pulses are sent from the device. When the object is traversed by the light, it produces light reflection and the receiver part of the LiDAR transforms the time elapsed between the sending and the reflection of the object in distance and speed. Typical LiDAR specifications are summarized in the Table 14. LiDAR has multiples usages. It is widely used in autonomous vessels. It can create a map of the ship's surroundings, enhancing the security of port operation or navigation in situations of scarce visibility and high maritime traffic as well.

In addition to the above sensors, vessels rely on *Automatic Identification System (AIS)* that is used as a localization system. The AIS is a broadcast transponder, and it is used to send the current vessel location, in GPS coordinates, current course and the current vessel speed to a satellite. At the same time, the AIS system shows the positions of all other vessels within a range. The transmission and retirement of the vessel data are conducted at regular intervals in order to keep the system updated. The system is helpful to avoid incidents and collisions. Together with radars, AIS is used as real-time sensors to raise awareness in the maritime environment. AIS can not be switched off, even when the vessel is at anchor.

Table 14. Typical LiDAR specifications.

Rage	Range (m)	Angle	Points	Pros	Cons
Long	35-100m	90°	5M	All weather performance High resolution Accuracy	High costs
Mid	90-200m	45°	5M		
Ultra	200-400m	22°	2M		Limited field of view

3.2 Understanding the situation

The understanding of the situation is a typical pattern recognition problem. When multiple sensors are used to understand the situation, the SA model uses all the sensors' information together. This process is called *sensor fusion*.

Sensor fusion can achieve a great level of accuracy in feature extraction from different types of input data, and it can be conducted at the input level of the neural network. In *early fusion*, [117] the input data from sensors is concatenated before feeding it into ML or DL model. The advantage of early fusion is that the data is seen as a unique stream, and less data processing is needed. This process may reduce the amount of errors in comparison to the use of single sensors. However, with the early fusion of data, it is difficult to find a common ground to relate each data point from different sensors.

Middle fusion [118] combines the data from different sources at an intermediate stage of processing after the initial feature extraction and before the final decision. This methodology is flexible, allowing the fusion at different stages of the model. Middle fusion allows a flexible process of feature extraction because it is able to isolate and collect only the proper features from each sensor, removing the not consistent ones. However, middle fusion can be heavy in computation because the process of extracting and combining features from different sensors is hard.

In the *late fusion* [119], the data from each sensor is passed to the model independently, and the results are fused in the final decision layer of the algorithm. The technique is used when the data from different inputs has different dimensionality, and it is difficult to find a common ground to keep the data in the same dimension. The advantage of late fusion is that the data are handled independently until the decision layer. Late fusion offers the advantage of flexibility because there is more freedom in the choices of the algorithm used to extract features from each single sensor. It also avoids the problems of combining fusion data together.

Digitalization and data collection have helped developers and researchers to develop powerful deep-learning techniques for the maritime environment. When the data is collected from sensors, and the perception stage is over, the understanding of the situation phase takes place. The data provided by the sensors described in Section 3.1 is used to precept the external environment. Once the data has been collected, a process called feature selection, see section 2.1.2 can be applied. Feature selection uses techniques such as greedy forward selection, greedy backwards forward selection, or genetic algorithms to select the data with the most significance in terms of accuracy. In fact, feature selection reduces the number of input variables, dis-

charging the data that contributes to reducing the accuracy, and it helps to reduce the computational cost of the model. Moreover, the collected data is used by DL or ML algorithms to create an efficient and accurate object detection and classification. In [120], a DL proposed to detect targeted objects (classification) using multiples different detectors. The approach selects the detector that produces best accuracy in the classification process, and then provides the localization of the object [121] using the same previously selected network [120]. Articles IV and VI use late fusion to achieve the best possible performance in pixel-wise classification. Article V studies different levels of fusion (late and early), comparing these methods with uni-modal architectures.

3.3 Predictive SA model

Prediction of future outcomes needs to create a model that is able to predict what would be SA after a certain time ahead. For example, the model should be able to predict the vessel's position against other vessels in 5, 7 or 10 minutes ahead of time. The decision after the prediction process should be considered in different scenarios as well. Articles I, II, and III employ Faster R-CNN, R-FCN, EfficientDet, and SSD to investigate detection performance across varying object sizes. In other scenarios, for example, the time series prediction task can be conducted by *The Long Short Term Memory network (LSTM)* [122]. LSTM is very helpful in time series analysis due to the ability to memorize information over a long time range. Special information gates handle the information flow. In order to train the model, LSTM uses techniques such as back-propagation. The back-propagation, described in Chapter 2, helps to update the weights of the network over the time series. Linear regression, in particular auto-regression, may be used to study time-series as well. The working principle of auto-regression is that the model uses past values to predict future outcomes. Auto-regression works well when the mean and the variance of the input data do not change over time.

3.4 Existing maritime datasets

In maritime applications, large amounts of data are needed for the SA to solve the detection problem with high accuracy. Due to the scarce amount of large ship datasets, one efficient way to address the data problem is the usage of transfer learning. Transfer learning [94] uses a trained network from a generalized dataset, MSCOCO [123], PASCAL VOC [124] and ImageNet [125], in a specific target task such as ship detection. In the maritime environment, there are some datasets that are valuable for building ML and DL models for improving SA.

In article I, and II Faster R-CNN, R-FCN, EfficientDet, and SSD models are pre-

Table 15. Types of general dataset with vessel annotations.

Dataset	Annotations	Vessels	Type of images	Categories
COCO	2500000	3146	RGB, Gray-scale	8
ImageNet	1034908	1071	RGB, Gray-scale, CMYK	15
Open Image	16000000	1000	Various	8
Pascal VOC	27450	353	RGB	1

trained using the MSCOCO dataset, which offers a large amount of features, rendering the process of feature generalization. Article III uses trained model with PASCAL-VOC and MSCOCO thanks to their abilities to locate object with high confidence score. Furthermore, in article III, the efficiency of employing various feature extractors within the detectors is analyzed.

In a maritime environment, one of the crucial operations is data collection. High-performance object detection and classification algorithms rely on large-scale and, multi-scenario datasets [121]. MSCOCO, ImageNet, Open Image and Pascal VOC, contain ship annotations:
The MSCOCO dataset contains RGB and gray-scale images. There are eight types of vessels in the dataset: 1) fishing boat, 2) cargo ship, 3) sailboat, 4) speed boat, 5) kayaks, 6) canoes, 7) raft, and 8) other types of watercraft. The ImageNet contains images from different colour spaces, such as RGB, gray-scale, and CMYK (Cian, Magenta, Yellow and Black). Vessels are inherited in two macro-categories: vessel and watercraft. They are summarized as follow: amphibious vehicles, narges, canoes, catamarans, cruise ship, fishing boat, gondolas, kayaks, lifeboat, motorboat, sailboat, steamship, submarines, tankers, watercraft. The OpenImage dataset contains various types of colour space images. Due to the continuous update to the dataset, it is difficult to say exactly how many vessel categories the dataset contains. It may include: cruise ship, kayak, lifeboat, motorboat, sailboat, speedboat, submarine, watercraft. The Pascal VOC dataset contains only RGB images in one category (boat).

In the maritime environment, there are few specialized datasets for ship detection. They are summarized in Table 16. The Singapore maritime dataset [126] contains 17450 annotated ships divided into six categories. The images are recorded during the day and night. The RGB images resolution is 1920x1080. The dataset also contains NIR images. Images are captures inshore and offshore. The ship categories are: 1) cargo ship, 2) tankers, 3) container ship, 4) passenger ship, 5) ferries, 6) tugboat,

7) pilot boat 8) pleasure boats.

The SeaShip dataset [127] consist of 31455 annotated RGB images with six different types of ship. The image resolution is 1920x1080. The categories are 1) ore carrier (5126 images), 2) bulk cargo carrier (5067 images), 3) general cargo ship (5342 images), 4) container ship (3657 images), 5) fishing boat (5652 images), 6) passenger ship (3171) [127]. In addition the SeaShip dataset contains also 3440 images.

The MCShip [128] contains 14709 labelled images with six military ships (7953 images) and seven civilian ships (8942 images). The images are obtained using a web crawler, and they have a minimal resolution of 500x500 pixels. In the military types of annotations, the dataset contains: aircraft carrier (906 images), auxiliary ship (926 images), landing ship (340 images), destroyer (4355 images), submarines (1175 images), missile boat (251 images). The civilian ship category contains: container ship (1208 images), fishing boat (1660 images), passenger ship (1219 images), sailboat (2444 images), speedboat (1087 images), tugboat (611 images), support ship (713 images).

The "ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations" [129] provides a maritime-specific dataset for inshore and offshore vessels. This dataset takes into consideration nine ship categories, and it is also recorded in different weather conditions, such as light and heavy rain, fog, and sun. The dataset contains 9880 images with a resolution of 1920x720 pixels. It contains nine ship types and two objects. The ship categories are: boat, cargo ship, cruise ship, ferry, military ship, misc boat, passenger ship, motorboat, and sailboat. Moreover, there are annotations for other objects such as floater, sea-mark.

Table 16. List of maritime datasets.

Name	Total images	Annotations	Ship types
Seaship dataset	31455	40077	6
Singapore Maritime dataset	17450	192980	6
MCShip dataset	14709	26529	13
Aboship	9880	41967	9

3.5 Existing Maritime SA system

Due to the development of new technologies in AI, the demand for the development of autonomous ships increased. Intelligent computer vision system supports the automation of ship navigation. Autonomous navigation helps to improve the SA of the vessel environment. In [130] the authors reported that 75 – 96 % of the collision is

due to human errors.

ML algorithms, such as SVM, can prevent abnormal ship movements [131]. The data provided by the boat AIS system can be used to predict the behaviour of each vessel's trajectories [132]. In addition, the speed signal and course signal from AIS system are used as input for AI algorithms to predict ship movements. Bayesian analysis offers advantages in pattern recognition in the maritime environment. Bayesian statistics handle incomplete data. It detects unusual activities in the AIS data [133]. In addition, the Bayesian analysis improves the accuracy of vessel tracking relaying only on AIS data. A discrete level of accuracy in tracking vessels is obtained using Ornstein Uhlenbeck stochastic process [134] applied to AIS data. The Ornstein Uhlenbec algorithm efficiently detects vessels in unusual course or speeds in heavy traffic conditions. AI algorithms require plenty of computational power to produce results with a reliable degree of accuracy. Tracking vessels is computationally expensive, which can be saved by detecting objects. The motion of a vessel is constructed over time, and it is not necessary to detect motion at each clock tick [135].

The authors of [115] state that horizon detection is one of the challenging tasks to do to provide a reliable SA of the maritime environment. In [136] the authors perform the horizon detection by using correlations between frames. To detect the horizon, they use two images, one as the base image and a new one that is tested against it. The correlation generates a peak, resulting in the separation between the sea and the horizon. Horizon detection can also be achieved as the maximum statistical colour distribution between a region classified as sea and the region classified as sky [137]. Hough transform [138] is effective in this task because it can detect a separation line between the sea and the sky. Therefore, the sea portion of the frame is further analyzed to detect objects.

One technique that is used to detect objects is called background subtraction [139] [140]. The background subtraction aims to detect background information from a frame. It then subtracts the background from the frame. Background subtraction techniques are challenging in maritime environments because the background is not static. Moreover, objects such as foam waves can be detected as foreground objects. The maritime background subtraction becomes more accurate when using FLIR cameras. These types of cameras can read the temperature of the water, creating a uniform space in the frame [139]. There are multiple background subtraction algorithms used in maritime environments; the authors of [140] illustrate the performance of 23 algorithms. The algorithms are divided into methods that use basic statistical distance techniques, Gaussian distribution, colours and texture, and machine learning approaches. The results from [140] show that background subtraction is a demanding task, and it is difficult to tackle in a maritime environment. The low accuracy is

due to the presence of colour variations, waves, and the ghosting effect of the camera.

At the data level, traditional machine vision approaches use background subtractions and foreground detection to extract features from a given input to detect objects. AdaBoost [141], HAAR cascade [142], and SVM [143] use sliding windows to collect visual descriptors of the image. These methods determine the basic characteristics of the proposed object, such as colours and shapes. Moreover, due to the particular continuous shape and orientation of vessels, the separation between ships and the background is a tedious operation, resulting in a reduction in the accuracy of ship detection because some background areas may be classified as a vessel.

Vessel localization and tracking are challenging problems due to the continuous changing of the environment. Rapid movement of objects is another demanding situation for real-time object detection and classification. Operations such as detection and classification must rely on multiple input sensors to achieve proper tracking and classification of other objects.

The authors of [144] proposed a multi-sensor fusion approach coordinated by a probabilistic data association method to tackle the inaccurate level of detection and classification in maritime environments. The object region proposal is generated by LiDAR, RGB camera, Infrared camera and radar. The data provided by each of the above sensors is fused, and then a Convolutional Neural Network is used to classify objects inside these regions proposed. In [144] the sensor fusion approach, in maritime environments reached the accuracy of 96.6%.

The author of [145] studied the sensor fusion approach to increase the level of SA in offshore working environments. The aim was to avoid dangerous situations during offshore operations. The data were provided by the Offshore Simulator Centre (OSC) and originated from different sources, such as audio, video, and biometric data. However, the author presented only fused data for a visualization approach. The study visualized in real-time the data provided by sensors. For all this data, a predictive model should still be implemented. The model should predict the SA in different ranges of time in a manner that would guarantee safety improvements.

In [146], a sensor fusion approach to identify and isolate dangerous zones during the placement operation of offshore hardware. The system is installed into the operational helmet of the operator, and it starts to vibrate, providing feedback when a dangerous situation is very close to the worker. Fusing the various sensors from different hardware installed offshore and creating a predictive model able to follow the work activities should increase the level of security in this type of operation. The authors of [147] provided a vessel detection pipeline using Faster-RCNN [148]. The

pipeline is divided into three major steps. In the first one, the authors use a region proposal layer to generate possible ship candidates using detectors such as LPB and HOG [149]. In the second step, the proposed objects are classified, and in the end, the pipeline verifies the detected region. The region evaluation is usually conducted by SoftMax, placed in the final layer of the CNN. The activator produces a probability distribution that minimizes the cross entropy between classes. However, the authors of [147] state that SVM achieves better results in object classification. The authors handled the vessel localization using HOG [149] instead of traditional bounding box regression adopted by R-CNN [150] or Faster RCNN.

In the Transfer Learning for Maritime Vessel Detection using Deep Neural Networks [151], the authors use transfer learning with Faster-RCNN, R-FCN, and SSD. The model is pre-trained using the Microsoft COCO dataset as the base domain. The transfer learning process is applied to a specific maritime domain dataset, the Sea-Ship dataset, explained in section 3.4. The authors of [152] state that the maritime environment needs to increase the level of efficiency in the classification and detection of vessel. However, a traditional neural network approach can result in a performance below the average due to the lack of specific data. Therefore, in order to achieve the desired level of the efficiency, transfer learning should be adopted. It is helpful to initialize all the parameters of a CNN and apply it to a smaller dataset. The authors of the article number 2 used eight pre-trained models using the ImageNet dataset. These models are: 1) VGG16 [153], 2) Xception [154], 3) Resnet50 [155] 4) DenseNet121 [156], 5) MobileNet [157], 6) EfficientNetB0 [158], 7) InceptionResnet [159], and 8) Inception [160]. The best model i.e. the one that produced the best accuracy, was the EfficientNetB0 with 94.88% accuracy.

4 Satellite and Areal Imagery

DL and ML algorithms are powerful in remote sensing (RS) applications. They have shown high performance in tasks such as land soil classification [161], land cover classification [162], semantic segmentation [163] and vessel detection [164]. Remote sensing utilizes different satellites and areal measurement systems such as Synthetic Aperture Radar (SAR), Airborne Laser Scanning (ALS), and optical satellites.

Furthermore, peatland semantic segmentation, as highlighted in [165], plays a crucial role in ecosystem understanding and finds widespread applications in agricultural field planning and urban management. Article V of the present study suggests its utility in land cover classification, as it assigns labels to individual pixels in images based on predefined classes.

4.1 Remote Sensing

ML has an important role in RS data processing. Traditional ML algorithms, such as SVM [166] and logistic regression [167], can classify different types of crops from RS data. They show advantages and disadvantages in terms of accuracy depending on the input data used. Therefore, fusing different satellite image sources is a solution to improve the land cover classification accuracy; early, middle, and late fusion are applicable.

Early fusion is conducted at the input level of the network. It can directly concatenate raw data from different sources. Data concatenation in the input level of the networks helps to use RS input with common spatial resolution. In middle fusion, input sources are fused at the intermediate level of the network. This fusion may be used when the different remote sensing sources have the same spatial resolution. Before the fusion process, each input's information is extracted as a stand-alone feature. Then, they are combined to obtain a more exhaustive representation of combined features as a single entity. Before the decision layers, some convolutional layers can analyse the detail of the obtained features. This fusion methodology is adopted in article V. Article V conducts a comparison between early and late fusion techniques in remote sensing. Additionally, it explores the differences between these

fusion approaches using all available bands of each input, as well as fusion using only the selected best bands.

In the late fusion, each of the inputs is treated as a single entity. The RS data is combined at the final decision layer of the entire network. It helps to isolate the feature from each single input. Late fusion is used in the SFS process in article IV and VI. Each of these fusion approaches, early, middle, and late, were already described in Section 3.2.

The authors of [168] showed the difference in the performance in the three types of fusion architectures of maritime vessel detection using colour and infrared cameras. These fusion architectures gained more performance in accuracy than non-fused architectures. In [169], the authors developed a sensor fusion for water and wet areas of the ground classification, integrating SAR images, optical images, and LiDAR data. In addition, they investigate the differences between the classification of individual sets of inputs and the fusion of these three inputs. In soil type classification, errors can occur due to the wind, especially in water and ice classification. Therefore, errors can be mitigated with the usage of SAR sensors operating within the same angle.

CNN achieves an acceptable level of accuracy in many remote sensing applications [170][171][172], as they can extract complex information from raster. Moreover, CNN is also used in sensor fusion. It exploits features from different remote sensing sources. The authors of [173] used a combination of SAR images and LiDAR data to classify rural and urban areas. CNN and sensor fusion were combined. This enhanced the quality of the raster by transforming and combining low-resolution raster which rendered the more efficient collection of relevant information. The fusion is used to combine and fuse pixels from different sources, transforming the original input into super-sampled high-resolution image [174]. CNN has been used to classify crop types from different satellite sources, gaining better accuracy than Multi-Layer Perception (MLP) [175].

4.2 Remote sensing data sources

In this section, only the most common RS data sources are reviewed, having also contributed to this thesis. Satellites are built for different purposes, and they identify earth's features from space, such as electromagnetic fields, radiation, weather conditions, and maritime surveillance. Satellites have spectral, spatial, temporal, polarization and optical sensors [176]. These sensors can read and translate in human-readable content the light emitted by the Earth. Spectral sensors are able to measure the reflected light from our planet. From their orbital position, spectral sensors can detect the chemical composition of objects, revealing characteristics of materials

such as the soil type composition.

The authors of [177] simplified the approach used to read the spectrum from the spectral sensor. In particular, they used arrays of photo-detector sensors where each sensor is in charge to read a particular wavelength range. Moreover, they used a predictive model to rebuild the final spectral image. In their study spatial sensors measure the size of the features. The resolution of the image refers to the size of the area contained in one pixel provided by the satellite image.

The authors of [178] investigated the relationship between the size of images and the information that it contains. The temporal sensors measure the elapsed time between the sensing acts of the satellites. In the [179], temporal analysis was used to tackle the challenges that imply monitoring the changes in the Earth's environment.

Polarization is defined as the skewness between the vibration direction relative to the spreading direction of an electromagnetic wave [180]. It measures the wave vibration in relation to the direction of the propagation of the light. Objects from the Earth spread a polarized signal within some directions. Therefore these signals collect information about this movement. The polarization sensors read the electromagnetic radiations emitted by natural Earth resources such as snow, rocks, clouds and oceans. The authors of [180] state that bidirectional polarization (polarization in two directions) is caused by: 1) solar electromagnetic radiation that irradiates the Earth and 2) electromagnetic radiation emitted by an object on Earth, and bi-directional polarization can enhance the information provided by the polarization sensors.

Aperture Radar is a sensor from the optical sensor family that produces and sends wave-light from space to a location. The SAR sensor calculates the quantity of energy that is reflected back from the Earth to the sensor. The resolution of the image provided by SAR sensors is directly related to the length of the antenna in which the SAR sensors are mounted. A particular optical sensor array is the *Multispectral Instrument* (MSI) [181] that allows collecting data from different bands of the electromagnetic spectrum. Some of the most used bands are summarized in the Table 17.

Spectral sensors, spatial sensors, temporal sensors and polarization sensors are mounted as sensor arrays into satellites. Sentinel-1 uses C-band SAR to detect features from the Earth [182]. These bands are summarized in Table 17. The main purpose of Sentinel-1 is the surveillance of the maritime environment, land, ice and forestry. The SAR module has a spatial pixel resolution of 5 meters. The Sentinel-1 constel-

[1] ESA: https://sentinel.esa.int/web/sentinel/missions/sentinel-1/instrument-payload/resolution-swath Accessed 26th November 2023.

Table 17. Typical Sentinel 1 (SAR) bands with their applications. In this table, for each band reported, a typical usage is described. source: Europen Space Agency (ESA).[1]

Band	Frequency	Wavelength	Application
Ka	27–40 GHz	1.1–0.8 cm	Airport surveillance
K	18–27 GHz	1.7–1.1 cm	Water absorption
Ku	12–18 GHz	2.4–1.7 cm	Satellite altimetry
X	8–12 GHz	3.8–2.4 cm	High resolution monitoring
C	4–8 GHz	7.5–3.8 cm	Ice, oceans and maritime navigation
S	2–4 GHz	15–7.5 cm	Agriculture monitoring
L	1–2 GHz	30–15 cm	Biomass and vegetation mapping
P	0.3–1 GHz	100–30 cm	Biomass and vegetation mapping

lation orbits in the same direction as the sun, with a cycle length of 175 days. Their operational mode is summarized in the Table 18. Articles IV, V, and VI use Sentinel-1 rasters.

Sentinel-2 is a constellation of two satellites. They orbit in the same direction as the sun. The aim of Sentinel-2 is land monitoring. In order to monitor the Earth, it mounts an MSI system composed of 13 bands, where four bands have 10 m spatial pixel resolution, six bands have 20 m spatial pixel resolution, and three bands have 60 m spatial pixel resolution. The MSI collects data row by row using the locomotion of the satellite. The applications of Sentinel-2 bands are summarized in the Table 19. Moreover, Sentinel-2 raster were used in article IV, V, and VI.

Table 18. Operational mode for Sentinel-1 satellites. Source: ESA.[2]

Operational mode for Sentinel-1		
SM	Stripmap mode	5m $\times$ 5m
IWS	Interferometric Wide Swath	5m $\times$ 20m
EWS	Extra Wide Swath	25m $\times$ 80m
WM	Wave mode	20km $\times$ 20km - 20m $\times$ 5m

In order to capture as much information as possible, the Sentinel-2 bands are combined together. Natural colour images are obtained by aggregating bands B4, B3, and band B2. Combination of bands B8, B4, and B3 creates a map of the vegetation status because band B8 is good for capturing the light reflected by the chlorophyll

[2] ESA: https://sentinel.esa.int/web/sentinel/missions/sentinel-1/instrument-payload Accessed 26th of November 2023

[3] ESA: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/instrument-payload/resolution-and-swath Accessed 26th November 2023.

Table 19. Bands and their typical usage for Sentinel-2. Each band can work as single feature extractor. Combining multiples bands together, more features can be extracted. Source: ESA.[3]

Band	Wavelength	Application
B1	443 nm	Coastal and Aerosol analysis
B2	490 nm	Blue band
B3	560 nm	Green band
B4	665 nm	Red band
B5	705 nm	Visible and Near Infrared (VNIR)
B6	740 nm	Visible and Near Infrared (VNIR)
B7	783 nm	Visible and Near Infrared (VNIR)
B8	842 nm	Visible and Near Infrared (VNIR)
B8a	865 nm	Visible and Near Infrared (VNIR)
B9	940 nm	Short Wave Infrared (SWIR)
B10	1375 nm	Short Wave Infrared (SWIR)
B11	1610 nm	Short Wave Infrared (SWIR)
B12	2189 nm	Short Wave Infrared (SWIR)

contained in the leafs of trees. The raster produced by combining the short wave infrared bands (B12, B8A and B4) describe the vegetation density in green scale. Bands B11, B8 and B2 are used to enhance the agriculture in the analysed areas. The bands B12, B11 and B2 are used to find geological features, and coastal studies are possible by combining B4, B3, and B1.

More complex imagery is possible by performing arithmetic operations for bands [183]. The vegetation index is obtained by (B8 - B4) / (B8 + B4), where dense colours show the good quality of the canopy. The moisture index is used to check the humidity of the terrain and it is obtained by (B8A - B11) / (B8A + B11). The band combinations are summarised in the Table 20.

Table 20. Typical band combinations Sentinel-2. Source ESA.[4]

Sentinel-2 MSI combination	
RGB image	B4 B3 and B2
Vegetation status	B8 B4 and B3
Vegetation status	B12 B8A and B4
Agriculture	B11 B8 and B2
Geology	B12 B11 and B2
Bathymetric	B4 B3 and B1
Vegetation index	(B8 - B4) / (B8 + B4)
Moisture index	(B8A - B11) / (B8A + B11)

The scope of the Radarsat satellites constellation (there are two twin satellites, Radarsat-1 and Radarsat-2) is to provide scientific information about forestry, agriculture, water and ice analysis. The main hardware that Radarsat uses is SAR sensors. It uses a C-band described in Table 17 with a frequency of 5.3 GHz. Radarsat is able to acquire images with single or dual polarization with different orientations, vertical and horizontal. The pixel spatial resolution of the image is between a range of 5 and 20 m for high-resolution images. The medium-resolution images offer a spatial pixel resolution between 20 and 500m.

Terrasar-X [184] is a German satellite that collects SAR images with very precise geographical information. The scope of these satellites is the understanding of the Earth's surface. It mounts an X band 17 in order to collect data. It uses image acquisition techniques such as strip-map, spotlight and scanSAR. Spotlight applications are mainly used in surveillance and intelligence applications. Moreover, the scanSAR is used as a surveillance device in maritime environments. The data collected from the twin satellites TerraSAR and TanDEM, imply the production of DEM images. The pixel spatial resolution has a range between 1 m and 16 m[5].

Since 1970, Landsat has been monitoring Earth's resources and environment. Currently, Landsat constellations are composed of nine satellites. However, some satellites, such as Landsat 7, have dismissed due to their extensive periods of utilization. Currently, only Landsat 8 and 9 are operative. Landsat satellites are designed to monitor the Earth's environment for extensive periods of time. Nowadays, the Landsat program has collected more than 4 million images, and this number is going to increase because Landsat 9 has the capacity to record more than 1400 images per day [6]. The images are acquired by fluctuating sensors back and forth continuously. At the end of fluctuation, sensors are calibrated in order to get the best possible information.

The Operational Land Imager 2[7] and the Thermal Infrared Sensor 2[8] are two innovative instruments that operate in Landsat 9. The Operating Land Imager 2, composed of a modern set of telescopes and photo-sensitives detectors, is able to capture high-resolution images with a range of 185km. The Thermal Infrared Sensor 2 is the

[4] ESA: https://www.eoportal.org/satellite-missions/copernicus-sentinel-2 Accessed 26th November 2023.

[5] ESA: https://earth.esa.int/eogateway/missions/terrasar-x-and-tandem-x#instruments-section Accessed 12th December 2023

[6] NASA: https://landsat.gsfc.nasa.gov/satellites/landsat-9/ Accessed 12th December 2023

[7] NASA: https://landsat.gsfc.nasa.gov/satellites/landsat-9/landsat-9-instruments/ Accessed 12th December 2023

[8] NASA: https://landsat.gsfc.nasa.gov/satellites/landsat-9/landsat-9-instruments/ Accessed 12th December 2023

improved version of the one mounted on Landsat 8. The new version removes the calibration problem of the previous version. It also mitigates artefacts produced by the previous model when sensors are exposed to intense light [185]. The authors of [185] state that calibration errors are reduced by 8%. Both Landsat 8 and 9 mount 11 bands. They are summarized in Table 21.

Table 21. Typical Landsat bands. Source: National Aerospace Agency (NASA).[9]

Typical Landsat 9 and 8 bands usage			
Bands	**Wavelength (µm)**	**Resolution**	**Applications**
1	0.433–0.453	30 m	Dust and smoke
2	0.450–0.515	30 m	Blue
3	0.525–0.600	30 m	Green
4	0.630–0.680	30 m	Red
5	0.845–0.885	30 m	NIR
6	1.560–1.660	30 m	SWIR
7	2.100–2.300	30 m	SWIR
8	0.500–0.680	15 m	Panchromatic band
9	1.360–1.390	30 m	Brights band
10	10.6-11.2	100 m	TIR
11	11.5-12.5	100 m	TIR

The Landsat 9 bands cover various ranges of the light spectrum. In particular, Band 1 captures a particular range of blue and violet, and it is mainly used to collect dust and smoke information. Moreover, it is used also to check the quality of the air and water. Bands 2, 3, and 4 are used mainly to produce RGB images. Band number 5 is in charge of analysing the near-infrared wave-lights. It is also used to check the vegetation status because the leaves of vegetation reflect the wave-lights back into the sky. Bands 6 and 7 are combined to produce SWIR, which measures the level of land's moisture. The Band 8 is used to collect black-and-white images of the Earth. The band produces accurate and sharp black-and-white images. Band 9 is unique in its gender because it can capture a very tiny portion of the wave-lights, and it is used to study clouds. The combined bands 10 and 11 produce the thermal infrared TIR that measures the temperature of the air [186].

Satellites offer plenty of source information about the Earth from space. However, satellites are not the only source of data for remote sensing. Areal photography is conducted on the Earth and it produces RGB images, LiDAR cloud map points and derivatives such as Digital Elevation Model (DEM) and Canopies High Model

[9] NASA: https://landsat.gsfc.nasa.gov/satellites/landsat-9/landsat-9-bands Accessed 26th November 2023.

(CHM). LiDAR, analysed in Section 3.1, is able to measure the distance between the sensor and the surface of the Earth. The quality of DEM and CHM is determined by the number of points per minute that a LiDAR is able to emit. For more information, see Table 14. DEM is derived by extracting from the LiDAR clouds the points with the lowest ground level, and the CHM are derived from the upper points. In addition, satellites such as TerraSAR-X and Radarsat-1 may produce DEM thanks to their interferometric sensors. DEM shows how the physical surface appears, providing a visual representation of the terrain elevation. The CHM, on the other hand, shows the status of the vegetation, providing their elevation information.

National Forest Inventory [187] (NFI) is the result of previous works, and it can be used as remote sensing data source. National Forest Inventory had a central role in the articles IV, V and VI included in this thesis. The NFI, provides periodic information about forests that cover all Finland, including de-forested zones. The raw data was obtained by collecting real in-site observations, and the NFI maps were created predicting from these in-site observations.

In the RS sector, there are multiple different data sources, such as data provided by micro and mini-satellites. However, the description of those is skipped because they are out of the scope of the present thesis.

5 Contribution of this thesis

In this section, the scientific contributions of our publications included in this thesis are reviewed. The publications are divided into two research topics. The first research topic deals with the maritime environment, and the second one concerns remote sensing. Three of these publications, articles I-III, deal with the the maritime environment. Three articles IV, V, and VI are related to the remote sensing. For each article, we provide a summary, description of methods and datasets, and the contribution of the author of this thesis.

5.1 Article I: Comparing CNN-based object detectors on two novel maritime datasets

Summary

This paper evaluates the performances of the three main CNN-based detection models including SSD, Fatser R-CNN, and R-FCN in the maritime environment. We investigate the impact of different feature extractors and vessel size on the performance of these models. In addition, the models are compared in terms of running time and accuracy. They are trained on large-scale MS-COCO dataset as a general dataset. The experiments are carried out using two real datasets which have been collected in Finland.

Methods and data

We selected one-stage detector (SSD) and two stage detectors (Faster R-CNN and R-FCN) for vessel detection. The detectors were used with different feature extractors. For instance, SSD was evaluated with MobileNet-v1 [188], MobileNet-v2 [189] and Inception-v2 [190]. Detectors Faster R-CNN and R-FCN were evaluated using different feature extractors NasNet [191], ResNet50 [192], ResNet101 [193] and Inception-resnet-v2 [193], see Chapter 2 for description of detection methods. Two real marine datasets were obtained in Finland to represent different weather and light conditions. The first dataset, (Dataset1), is a preliminary version of the dataset that will be described in the article 2. It is composed of 4800 frames captured from 135 videos recorded from June to July 2018 between Turku and Ruissalo archipelago

Table 22. Summary of the average precision (%) for Dataset1.

Object size	Detector	Feature extractor	AP
Small objects	Faster R-CNN	NasNet	34.28
Medium objects	Faster R-CNN	ResNet101	55.65
Large Objects	Faster R-CNN	ResNet101	74.00
Al objects	Faster R-CNN	ResNet101	57.05

in South-West Finland. The frames have a resolution of 1920×720 pixels. To enhance the dependability of the article's findings, 400 frames were chosen randomly and then they were manually annotated. The total number of annotated vessels was 850. In addition, the frames included scenes with different lighting conditions and weather varieties.

The second dataset, Dataset2, includes 1750 images from Turku archipelago. The dataset contains vessels (passenger vessels, motorboats, sailboats, and docked vessels) with different types of surrounding landscapes. The dataset contains RGB images and IR images. The size of the images is 1200×400 pixels. The total number of annotated objects was 9137.

Results and contribution

To evaluate the performance based on different object sizes, we computed the Average Precision (AP), metric for small, medium, and large vessels in both datasets see Table 22 and 23. In maritime, the object can be categorised based on the size of bounding box that represents the distance between the object and camera. Small objects are transformed into distant objects, medium objects become objects at a moderate distance, and large objects are considered as close objects. For small objects the area is less than 32^2 pixels, medium objects where the area is in between 32^2 pixels and 96^2 pixels and large objects have an area greater than 96^2 pixels. Area is measured as the number of pixels in each bounding box. For small vessels, Faster R-CNN (NasNet as feature extractor) got the best accuracy in both datasets. Faster R-CNN with ResNet101 can have high accuracy for medium (55.6%), large (74.0%) and all objects (57.0%) compared to other methods. SSD (MobileNet-v1) got the best accuracy for small objects in Dataset 2. For medium objects, Dataset 1 scored 55.6% with Faster R-CNN (ResNet101) and Dataset 2 scored 24.6% with SSD (MobileNet-v2). For large objects, Dataset1 got the best score (74.0%) with Faster R-CNN (ResNet101) and Dataset 2 (86.4%) with R-FCN (ResNet101).
To compare the proposed models in term of the running time, Faster R-CNN with inception V2 is the fastest. However, its performance is lower than Faster R-CNN with ResNet101. As a conclusion, this article contributes to investigating how the

Table 23. Summary of the average precision (%) for Dataset2.

Object size	Detector	Feature extractor	AP
Small objects	SSD	MobileNet-v1	16.43
Medium objects	SSD	MobileNet-v2	24.66
Large objects	R-FCN	ResNet101	86.46
All objects	Faster R-CNN	ResNet101	19.60

observed object distance affects the recognition and detection accuracy. Without the usage of a specialized dataset, it is not possible to obtain high recognition rates.

Author's contribution

The author of this thesis implemented the Python code in charge of preparing the dataset and creating the annotations manually for Dataset1. Moreover, he also helped in the evaluation process and in the interpretations of the results. He has also took part in the process of writing the article, especially the related work section.

5.2 Article II: Aboships—an Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations

Summary

Maritime vessel detection is crucial for autonomous maritime vehicles or maritime surveillance. Satellites can provide real-time data for the maritime environment, but their use is challenging because satellites can not provide valid information about the visible portion of the vessel, occlusion, and scale variations of the vessels. To address these problems, it is necessary to create a maritime dataset from waterborne images. Waterborne images can be helpful in relevant tasks such as maritime surveillance, monitoring of illegal fishing, and military operations.

In this article, a new maritime dataset, Aboships was created and evaluated by different CNN-based detectors. The dataset is specialized for performing vessel detection and classification. It contains 9880 images which are manually annotated into vessel instances (including nine types of vessels), seamarks and miscellaneous floaters. Totally, the Aboships dataset includes 41967 annotations.

Methods and data

In order to improve the annotation process and increase the consistency of the dataset

labels, the authors use the Channel and Spatial Reliability Tracking (CSRT) [194][195] as a common tracker algorithm. The CSRT algorithm tracks objects in videos or in a series of frames. It uses a set of correlation filters derived from the target objects to track the objects frame by frame. In addition, the correlation filters are updated in each frame in order to estimate the changes in dimension or appearance of the objects. In this article, if the CSRT tracker loses objects after a certain amount of frames, these faulty frames are labelled again. For performing object detection and classification, we evaluated four state-of-the-art CNNs which were used in article I with the same feature extractors. Here, we also tested EfficientNet D1 [87]. In addition, the performance of all detectors was evaluated based on the object's size.

The primary source of the dataset was a set of 135 videos recorded with a resolution of 1,920 $\times$ 720 pixels. The camera was installed in a vessel and it recorded videos for 13 days (26 June 2018–8 July 2018), between Turku and Ruissalo. The recorded videos contain a variety of weather conditions. A LiDAR system was adopted as well, with the primary purpose of calculating the distance between the vessel and all the other objects around. When the LiDAR detects objects far away, or the detection is out of range, the frames were removed from annotation.

Results and contribution

The chosen accuracy evaluation methods are the Intersection Over Union (IOU), and AP, see Chapter 2. The average precision is calculated for objects of different dimensions in the same way as in Article 1. The experimental results (Table 24) show that Faster R-CNN with InceptionResNet V2 as a feature extractor got 35.18% total AP for all objects. For small objects, Faster R-CNN with InceptionResNet V2 also got the best performance. For medium objects, SSD with ResNet101 got 31.18% AP. EfficientDet achieved the maximum AP (55.48%) for large objects. The first contribution of this article was the creation of a precise and open-source annotated maritime dataset. The second contribution relies on the measurement of the performance of four well-known state-of-the-art CNN methods with several feature extractors.

Author's contribution

The author of this thesis supervised and helped for creating the annotations. Moreover, the author wrote the Python code for the dataset evaluation algorithm and took part at DL model development. All the article's authors contributed to analyses of the evaluation results. The author of this thesis actively contributed to the writing of the article, specifically focusing on the creation of the tables. Particular attention was dedicated to crafting the tables, detailing various object classes and vessel counts,

Table 24. Summary of the average precision (%) for Aboships dataset.

Object size	Detector	Feature extractor	AP
Small objects	Faster R-CNN	Inception ResNet V2	23.16
Medium objects	SSD	ResNet101 V1 FPN	31.18
Large Objects	EfficientDet	EfficientNet D1	55.48
All objects	Faster R-CNN	Inception ResNet V2	35.18

requiring meticulous care and precision.

5.3 Article III: Transfer Learning for Maritime Vessel detection using Deep Neural Networks

Summary

The aim of this work was to study how transfer learning could improve the performance of vessel detections in maritime environments. For this purpose, we use a domain and none-domain dataset for pre-training. None-domains dataset is a generic COCO dataset. The domain dataset is an open-source marine dataset such as SeaShip (see section 3.4).

Object detection is a challenging operation in the maritime environment due to various weather conditions, lighting conditions, and vessel oscillations due to waves. CNN model requires a large amount of data in order to achieve acceptable results in terms of accuracy. In most cases, there is not sufficient data to train the models properly. The way to avoid the scarcity of the data problem is to use transfer learning. Transfer learning has the ability to transfer the knowledge contained in the weights of a network to another network. In this way, it is possible to use the power of deep learning in multiple fields.

Methods and data

The same deep CNNs as in article II were selected for this article as well. They respectively are Faster R-CNN, R-FCN and SSD. The Faster-RCNN is a two-stage detector. The first stage acts as a Region Proposal Network. It determines the potential regions of interest from the input, which go to the second stage. Here, a Fast R-CNN network performs the double task of classifying the type of object and producing bounding boxes. R-FCN, avoids two stage operation and operates only with Convolutional layers. The results are generated directly from the feature maps, which are all shared across the network without using the region of proposals. Then, the feature maps are scored, and the ones with the best scores are used to classify the

objects and to generate the bounding boxes. The third detector was the SSD. It is a one-stage network where classification and bounding box creation occur in the same stage. The production of bounding boxes and the classification of objects occur at the same time in the same network. These three detectors were trained with the TensorFlow Object detections API, which is an open-source framework for object detection.

The proposed CNNs were used in two different experiments to investigate the effects of transfer learning. In the first one, the initial weights were provided by the COCO dataset, and then the model was re-trained with the Dataset 2 (see article I). In the second experiment, the models were trained on the SeaShip which was described in Chapter 3. SeaShip provided authentic, real-world vessel operation for inshore and offshore ships.

Results and contribution

Each tested CNN detector produced the bounding boxes and then the corresponding classes. The performance measures used to evaluate the bounding boxes were IOU, AP, and Recall, see Chapter 2.

We calculated AP for all possible IOU thresholds and sizes of the objects (small, medium, and large). The mean average Precision (mAP) was used to calculate the accuracy over all classes. The Average Recall (AR) measures two times the area under the Recall. AR was calculated with different numbers of objects. The full results are reported in the Table 1, 2, and 3 of the article III. The results shows that the highest accuracy is achieved by the Faster R-CNN with ResNet101 as feature extractors, with a mAP of 38.4% when the model was pretrained with COCO dataset. However, we got 39.1 % when we used SeaShip for transfer learning from the same model. In summary, the article contributes to evaluating transfer learning in the specific maritime domain and a general dataset for vessel detection. Moreover, we evaluated the performance of different CNN based detectors based on the different objects' size.

Author's contribution

The author of this thesis created the python code in charge to process the data according the requirement of CNN models, took part in the DL model development and conducted the experiments. The author also conducted the transfer learning process, and evaluations of the results. He also participated in writing the article. In addition, the author participated in writing the article, with a particular focus on the literature review and Section 3, which delves into the state-of-the-art deep CNNs.

5.4 Article IV: Multistream Convolutional Neural Network Fusion for Pixel-wise Classification of Peatland

Summary

This article proposes a CNN in order to fuse multiple input sources with different spatial resolutions for peatland site (location) type classification. The proposed fusion architecture is of the late fusion type as three separate streams are concatenated in it to generate the final pixel-wise classification. The data are acquired by optical and radar satellite remote sensing, airborne laser scanning data and multi-source forest inventory GIS datasets. Based on our data sources, we are dealing with high-dimensional class-imbalanced data for solving pixel-wise classification of peatlands. To reduce the data dimension and find an optimal subset of inputs, we first applied the sequential feature selection method. Then, we proposed a window-based pixel classification approach based on the selected inputs. This approach can extract the spatial information around each training sample in a defined window region and produce a pixel-wise classification map. Experiments were carried out for ecological classification of peatlands in Finland.

Methods and data

We collected the data from four different remote sensing sources. (1) The Synthetic Aperture Radar (SAR) data contains raster from Sentinel-1. In addition, two types of Sentinel-1 rasters are used in this work: intensity images and coherences. The coherence images are polarized, concatenating vertical and horizontal looks. The two-way polarization reduces the noise from a single polarization. The Sentinel 1 images were acquired between May and September of 2017. TerraSAR-X raster is used in the dataset as well. These rasters cover a period of time between May and June. Radarsat-2 were acquired in 5 days from the beginning of May to August.

(2) The optical satellite images are from Sentinel-2. The images cover a period of time between May and September, between 2018 and 2020. Both SAR data and optical satellites share the same spatial pixel resolution (10 m). The spatial pixel resolution identifies how many meters a pixel of the raster is in the real world. For this peculiarity, the sets of raster are integrated into the same input stream. SAR images and optical satellites data are part of the first stream of the proposed CNN fusion architecture.

(3) The open source Airbone Laser Scanning (ALS) data acquired with Light Detection And Ranging (LiDAR) was provided by the National Land Survey of Finland

(NLS) from approximate flight altitudes of 1800–2500 m in varies of flight campaigns conducted in 2008 – 2019. The ALS data is used to derive both the Digital Elevation Model (DEM) and the Canopy Height Model (CHM). Both DEM and CHM are converted into a spatial pixel resolution of 2 m. The second input stream of our CNN model consists of DEM and CHM.

(4) Multi-Source National Forest Inventory (MS-NFI) of Finland produces information of forests in the form of thematic maps of forest variables, see Chapter 4 for more details. The current resolution of the MS-NFI map grid is 16 m.

The study area is called Keminmaa which belongs to the Southern Aapa mire zone. Experiments were carried out for two different versions of the datasets, identified as version 1 (V1) and version 2 (V2). Dataset V1 has 2065 data points divided into 39 classes. In the dataset V2 dataset, the data points and site type information are the same as in V1 but it was divided into *drained* and *undrained* datasets in order to remove miss-classification between the pristine areas and the areas which have been artificially drained for forestry.

As to methodology, the *Greedy Forward Selection* (GFS) algorithm conducts the process of selecting the best possible combination of input data for the training phase (input selection). Due to the highly imbalanced dataset used in this article, the GFS uses a subset of features from the original feature set. The process starts with an empty set of features. During each iteration of the algorithm, each unselected feature is evaluated with stratified k-fold cross-validation (described in Chapter 2). The new feature is validated against all the others already previously selected. The one that achieved the highest performance gain will be included in the current feature selection. For both input selection and classification phases, we proposed a multi-stream CNN fusion architecture (Figure 11). The architecture consists of three separate streams according to the different spatial resolution of inputs. The leading idea of the three-stream architecture is to preserve all possible information of the different resolution inputs, instead of interpolating the inputs to the same resolution while losing potentially useful information. Each stream is comprised of two convolution layers, two pooling layers, and one fully connected layers. This architecture was found as a good compromise between the model complexity, number of training samples and classes in our problem.

We also the used 5-fold Stratified Cross Validation (SCV) technique to divide the dataset into training and validation datasets. SCV is one of the standard methods to evaluate classifier's generalization accuracy. Compared to the standard CV, SCV ensures that each fold of the dataset has the same distribution of the classes in each fold to address the class imbalance problem. We also applied data augmentation to generate additional training data by random rotation, vertical and horizontal flips. Each

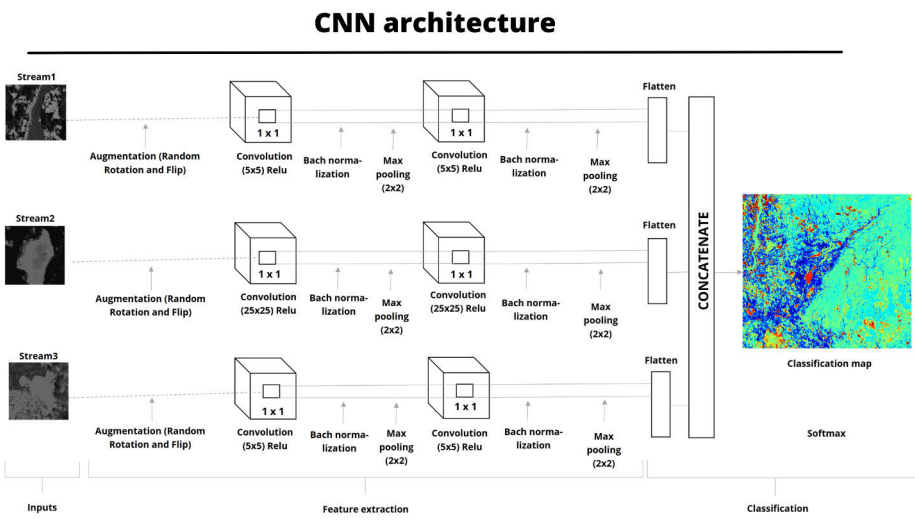

Figure 11. The CNN architecture used in the Article 4.

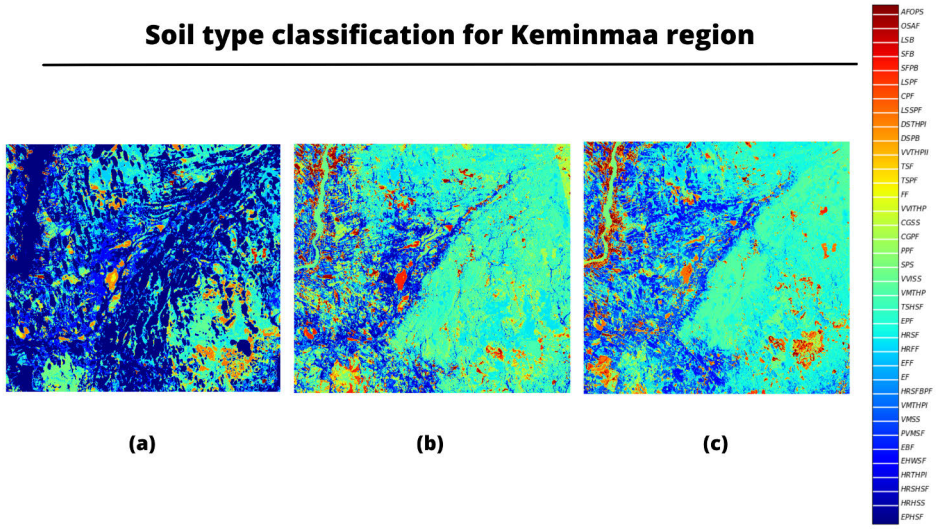

Figure 12. Keminmaa final classification maps. (a) the Keminmaa predicted peatland obtained with Dataset V1. (b) the Keminmaa peatland with undrained part of the peatland, and (c) the undrained counterpart.

pixel is classified based on the trained CNN model with the selected inputs. For this task, we proposed a sliding window approach. that refers to a rectangle region with a defined width and height that moves over the image. In fact, each pixel is labelled one-by-one, with the same amount of surrounding pixels as a spatial pattern to help to classify it. The pixel is always at the centre of the window. The classification starts from the upper left pixel. Then the features of the whole window are extracted separately for each stream. After that the class predictions from each individual stream are combined. This is done by first concatenating the features, using aggregation. After classifying the upper left pixel, the window is moved into the next pixel. When the window arrives at the last pixel in the first row, it moves down one pixel and the same process starts again from the left most pixel. This process iterates until the window has moved over all pixels.

Results and contribution

The overall classification accuracy is 32.4%, 33.6% and 31.8% for Keminmaa V1. Keminmaa V2 (drained) and Keminmaa V2 (undrained) (33.6%). The results are summarized on Table 25. The best accuracy is obtained for *drained* of Keminmaa V2. The resulting peatland maps are shown in Figure 12 for two versions of dataset. The main contribution of this article is applying a CNN for peatland analysis when combining three different types of inputs with multi-resolution. The proposed methodology can overcome the problems in this application such as class imbalance, computational and time complexities.

Table 25. The classification performance metrics for Keminmaa region.

Performance type	Keminmaa V1	Keminmaa V2 (drained)	Keminmaa V2 (undrained)
Classification accuracy	32.4	33.6	31.8
Average user's accuracy	25.8	15.1	16.2
Average producer's accuracy	21.0	12.9	18.8
Kappa value	29.2	21.1	27.6

Author's contribution

The author of this thesis took part to the methodology design of the entire project. He was in charge of creating the Python code for methodologies. In particular, he created the data pre-processing algorithm, the feature extraction algorithm, and the window-based pixel-wise classification map. The feature extraction process and the pixel-wise classification approach where implemented as parallel process in order to speed up the entire process. In addition, he participated in the writing of the methodology Section.

5.5 Article V: CNN-based Boreal Peatland Fertility Classification from Sentinel-1 and Sentinel-2 Imagery

Summary

This article is an extension of article IV treated in section 5.4 adding the following contributions. 1) We use open-source satellite data such as Sentinel-1 (Sentinel-1) and Sentinel-2 (Sentinel-2) because based on our previous results, they may provide sufficient information for peatland classification. The open-source satellite data greatly adds chances for using the results in further studies and operational work such as producing land cover maps periodically. 2) Unlike in article IV where peatland site types were reclassified into higher hierarchical level fertility classes, we train the CNN directly to five class fertility level classes and two land-use classes both on forestry drained and undrained lands. 3) We also investigate how the fusion level effects the results when we combine these two input data. Early and late fusion architectures are considered. We also compare the performance of the fusion architectures to the uni-modal architecture which uses only one input data type for the peatland fertility level classification. 4) Two scenarios are proposed for fusing Sentinel-1 and Sentinel-2 data. The first scenario combines all features (bands) of Sentinel-1 and Sentinel-2 images by using the two proposed architectures. The second scenario combines only the best band of each raster dataset of Sentinel-1 (2 bands) and Sentinel-2 (10 bands) in two early and late fusion levels. The study area is Keminmaa such as in article IV. The study area is divided into drained and undrained subareas, which are analysed individually to reduce the classification error.

Methods and data

The data used in this article is composed of SAR and Optical images. The SAR images contain two Sentinel-1 data: 1) Intensity images and 2) coherence images. The intensity images from year 2017 were acquired one per month from May to September. Intensity images refer to images built within a multi-look system and re-projected, which are referred to with *repeated overpass* images (RO) in this work. The *coherence images* are captured with the single-look methodology, and they acquire the radar reflection. The first set of coherences (12 images) were captured using the Sentinel-1A, and they are temporally consecutive pairs. The images were acquired between May and September 2017. In addition, a set of other 12 images is included as reference images. The total number of the coherence images was 24. Sentinel-2 images are used as optical data and they are from May to September from 2018 to 2020. For Sentinel-2, all the 10/20m bands were used. We refer to them as *long-term baseline* (LB) coherence images.

Two CNN fusion architectures are proposed which combine the input from SAR images and Optical images. The first architecture (Figure 13) combines the two inputs by stacking them one by one, before feeding them to the CNN (Early fusion). The anatomy of the first CNN is as follow: after the input and the data-dimensionality combination, a convolution layer followed by a max pooling layer is located. The final classification uses a Fully Connected layer. The second architecture uses a late fusion network (Figure 14), where each input has its own independent stream. The output of each stream is concatenated to a fully connected layer and the type of land is classified within a softmax activation function. Moreover, to study the variation in the accuracy of the two architectures described above, results for uni-modal CNN architecture are reported as well.

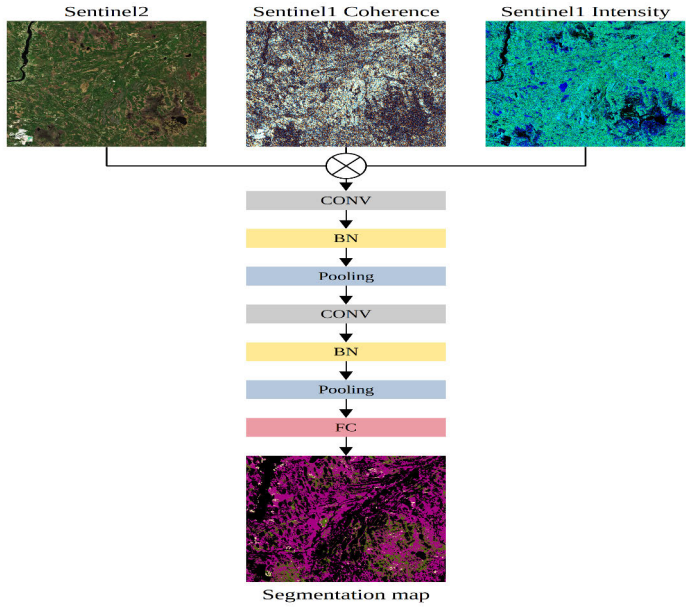

Figure 13. The early fusion architecture used in Article V.

Adam [196] algorithm is used to optimize the learning weights generated by each iteration of the CNN. The categorical cross-entropy [197] is used to determine to which class the sample belongs. To avoid overfitting of the model a dropout layer is used. In addition, random rotation horizontal and vertical directions flips are implemented to generate data augmentation. The accuracy of the model was determined by the Stratified K-Fold cross-validation method. It ensures that all the classes of the training sets are in each fold.

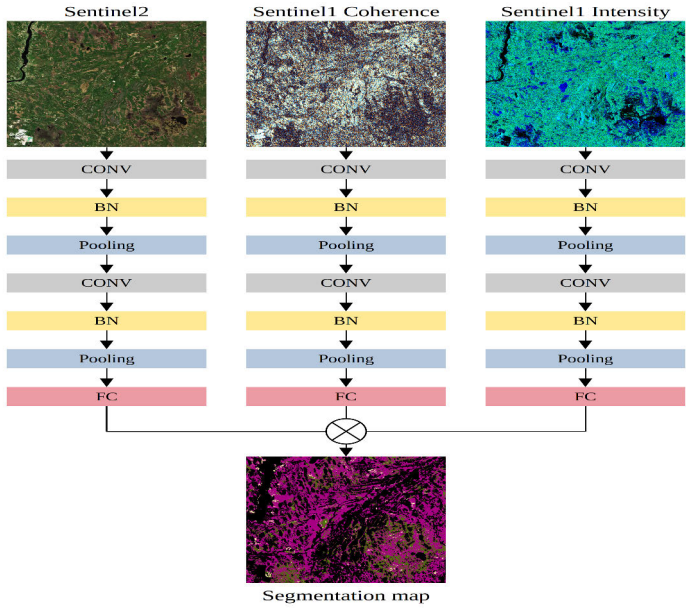

Figure 14. The late fusion architecture used in Article V.

Results and contribution

The performance of the model was evaluated using four different data input. Each of these included several different bands as told in section 4.2. In particular, 1) Intensity data included 3 bands, 2) Coherence RO included 2 bands, 3) Coherence LB included 2 bands, and 3) Sentinel-2 included 10 bands. These bands cary different amounts of information as seen in Table 26. The Table gives a summary of the results of the accuracy giving the best bands of each data input. Here *VV* stands for Vertical look, and *VH* stands for horizontal look. In Sentinel-2 B2 represent the blue band. To see which input data and band has more information on peatland level we determined the performance of CNNs for each single bands. This was achieved by training the best band of each input as a single entity and training the same data with all bands.

The performance of the CNNs was evaluated using four different datasets 1) Intensity data, 2) coherences RO are called *VV*, 3) coherences LB in Table 26 are called *VH*, and Sentinel 2 all bands. These datasets were available for two different types of soils grounds (drained, undrained). The best overall accuracy is obtained by the raster coherence LB acquired on 20.07.2017 with 38.85%. In the not fused (uni-modal) CNN architecture, Sentinel-2 got the best accuracy both for drained (38.95%) and undrained (42.49%). Due to the fact that Sentinel-2 has more bands than Sentinel-1,

Table 26. The classification accuracy from all the possible inputs.

Type of data	Drained			Undrained		
	Acquisition date	**Band**	**Accuracy**	**Acquisition date**	**Band**	**Accuracy**
Intensity data	26.09.2017	VH	32.96	26.09.2017	VV	35.98
Coherences RO	20.07.2017	VH	37.81	30.09.2017	VV	35.12
Coherences LB	20.07.2017	VH	38.85	01.08.2017	VV	34.70
Sentinel2	18.07.2018	B2	38.85	18.07.2018	B3	42.49

the accuracy is better.

Table 26 presents the best input combinations selected from all possible sets. This table highlights the raster with the highest accuracy from each input set. It includes both the top-performing raster and its best band. This method helped to eliminate inputs that did not yield high accuracy. Furthermore, Table summarizes the accuracy achieved by the best input across all bands identified in the previous table. The evaluation contains both uni-modal CNNs and multi-modal fusion CNNs.

Table 27 summarized the accuracy obtained by evaluating both uni-modal and multi-modal CNNs using only the best bands obtained from Table 26. In the fused (multi-modal) CNN architectures section, the highest accuracy was obtained from the late fusion (both drained and undrained). These late fusion results mean that the two inputs have different information that is combined on the classification layer. The accuracy of best bands reaches 50.36% when the input is Sentinel-1 Intensity and Sentinel-2. Comparing this result with the same input of uni-modal, the accuracy rises by about 5%. The undrained accuracy is 56.73% when the inputs are coherence (RO + LB) and Sentinel-2.

In the early fusion, the accuracy obtained with the best band is 51.66%, and the same combination has the best-undrained accuracy with 49.64%. With all bands together, the best-drained accuracy is for Sentinel-1 intensity and Sentinel-1 coherence (42.43%). For the undrained Section, the accuracy arrives at 48.93% using Sentinel-1 intensity + Sentinel-1 coherence (RO + LB) + Sentinel-2. The results are summarized on Table 27 and Table 28.

Author's contribution

The author of this thesis actively participated in the development of the methodologies' of the publication. In addition, he was in charge of programming, which was done using Python. The author was also involved in the discussion and the practical part of the CNN fusion design because it was a crucial aspect of the design of the CNN. The author also participated in the brainstorming of research questions and results discussions. In addition, he participated during the writing process of the results and discussion Section and the experiment design Section.

Table 27. The classification accuracy of the proposed fusion architectures with best bands.

Architecture	Input	Accuracy with best bands (%)	
		Drained	Undrained
Uni-modal	S1 intensity	35.66	31.91
Uni-modal	S1 coherence (RO + LB)	44.48	32.14
Uni-modal	Sentinel 2	47.88	45.95
Early fusion	S1 intensity + Sentinel 2	47.05	43.26
Early fusion	S1 intensity + S1 coherence (RO + LB)	47.23	45.39
Early fusion	S1 coherence (RO + LB) + Sentinel 2	51.66	49.64
Early fusion	S1 intensity + S1 coherence (RO + LB) + Sentinel 2	45.22	48.93
Late fusion	S1 intensity + Sentinel 2	50.36	44.36
Late fusion	S1 intensity + S1 coherence (RO + LB)	45.22	45.39
Late fusion	S1 coherence (RO + LB) + Sentinel 2	49.44	56.73
Late fusion	S1 intensity + S1 coherence (RO + LB) + Sentinel 2	45.58	43.26

Table 28. The classification accuracy of the proposed fusion architectures with all bands.

Architecture	Input	Accuracy with all bands (%)	
		Drained	Undrained
Uni-modal	S1 intensity	30.88	35.91
Uni-modal	S1 coherence (RO + LB)	34.19	35.09
Uni-modal	Sentinel 2	37.13	41.65
Early fusion	S1 intensity + Sentinel 2	40.80	32.62
Early fusion	S1 intensity + S1 coherence (RO + LB)	42.43	46.80
Early fusion	S1 coherence (RO + LB) + Sentinel 2	46.69	46.09
Early fusion	S1 intensity + S1 coherence (RO + LB) + Sentinel 2	37.13	47.51
Late fusion	S1 intensity + Sentinel 2	33.08	32.62
Late fusion	S1 intensity + S1 coherence (RO + LB)	43.17	41.13
Late fusion	S1 coherence (RO + LB) + Sentinel 2	50.18	43.97
Late fusion	S1 intensity + S1 coherence (RO + LB) + Sentinel 2	43.75	51.06

5.6 Article VI: Peatland Pixel-level Classification via Multispectral, Multiresolution and Multisensor data using Convolutional Neural Network

Summary

This article proposed a CNN model for creating a high-resolution land cover segmentation map based on multi-source geospatial datasets. The segmentation map is obtained using a late CNN fusion architecture that combines different resources, including optical and SAR satellite imagery, DEM, CHM, and NFI. The quality of peatland classification is evaluated by the accuracy precision recall and F1-score.
The three study areas (Keminmaa, Southern Ostrobothnia, and Eastern Finland) differ in vegetation types and peatland vegetation. Keminmaa is dominated by fertile site types. Easter Finland has plenty of poor nutrient types, and the Southern Ostrobothnia is dominated by abandoned agricultural fields. In addition, for each region the nutrient level, peatland, drained and undrained, are produced as well.
The peatland was derived into two different categories: drained and undrained. The difference between the undrained portion and the drained portion of the peatland was already described in the two previous articles IV and V.

Methods and data

The CNN architecture is the same as in Article 4 (Figure 15). A variety of Synthetic Aperture Radar (SAR) images is adopted in the present work. In particular, Keminmaa uses TerraSarX and Radarsat 2 images that were not available for the other two pilots. All three pilots use rasters from Sentinel1. Moreover, the Sentinel1 vertical looks and the horizontal looks rasters are polarized in order to keep as much information as possible. The Optical and SAR data are the first stream of our CNN. The ALS data is used as an input of the second stream of CNN. The DEM and CHM have 2 m and 1m spatial pixel resolutions, respectively.
In order to solve the problem of non-matching pixels in the raster, the CHM 1 m was re-sampled to 2 m. These derivations are present in all three pilots. The third stream of the proposed CNN uses NFI data as an input. The approaches used in this article can be summarized into: input data preparation, input selection and pixel-wise classification.
In the end, the nutrient peatland is derived, reducing the peatland map to their upper hierarchy fertility level, OSAF (Organic site agricultural fields), AFOPS (Abandoned agricultural fields), and Negative (mineral site).

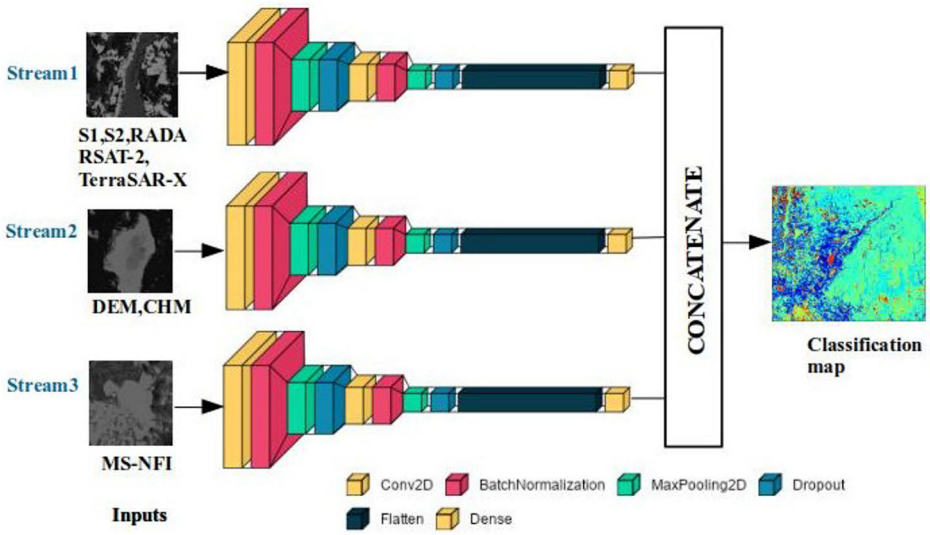

Figure 15. The CNN fusion architecture used in the Article VI.

Results and contribution

The classification metrics for different peatland site types are reported in Table 29. For each region, the highest accuracy is obtained by the undrained portion of the peatland classification. The classification metrics obtained of fertility levels are summarized in Table 30. The best total classification accuracy for undrained nutrient types is obtained for Keminmaa (55%). For the drained dataset, the Southern Ostrobothnia obtained the best accuracy (50.7%). Moreover, precision and F1-score have similar results which differ from the other metrics.

Table 29. Metrics for peatland site type classification using a data fusion CNN Neural Network.

	Keminmaa		**S. Ostrobothnia**		**E. Finland**	
	Undrained	Drained	Undrained	Drained	Undrained	Drained
Accuracy	**33.6**	31.8	**32.8**	32.5	**31.8**	29.6
Precision	13.9	11.9	13.9	12.9	9.6	7.8
Recall	13.9	12.9	14.7	13.7	10.6	7.9
F1 score	10.2	15.5	13.1	12.1	6.2	6.6

The classification map in Figure 16 shows the peatland of the Ostrobothnia, Figure 17 and 18 show the peatland classification map for the Eastern Finland and Keminmaa, respectively. We also created the confusion matrices based on site type and fertility level classes.
One of the main contributions of the article is an efficient sliding window-based data

Table 30. Metrics for fertility level classification using data fusion CNN.

	Keminmaa		S. Ostrobothnia		E. Finland	
	Undrained	Drained	Undrained	Drained	Undrained	Drained
Accuracy	**55.0**	48.1	**52.8**	50.7	**47.2**	36.6
Precision	44.2	40.3	39.0	43.3	32.7	28.7
Recall	35.3	42.3	39.2	45.7	35.3	21.1
F1	40.7	35.7	36.3	43.1	29.6	21.9

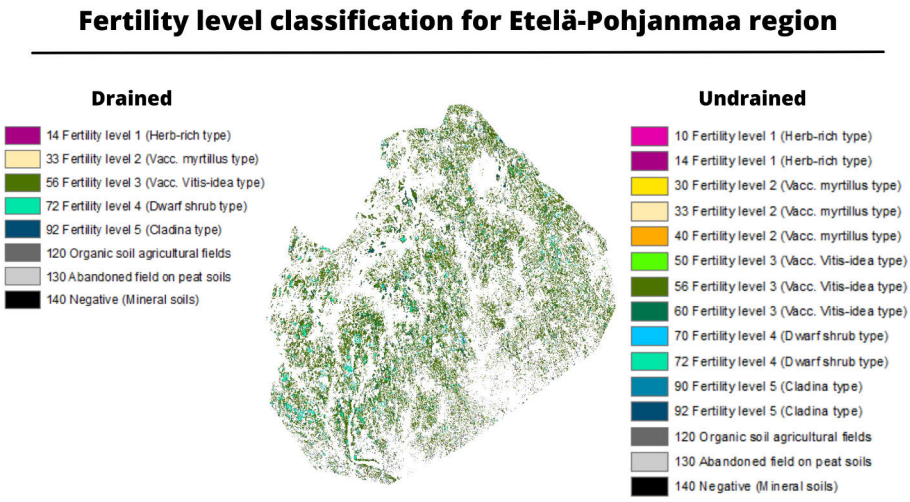

Figure 16. Southern Ostrobothnia Peatland. Drained and undrained section of the Peat are combined. This map shows the level of Fertility Level for undrained part of the peatland and drained ones.

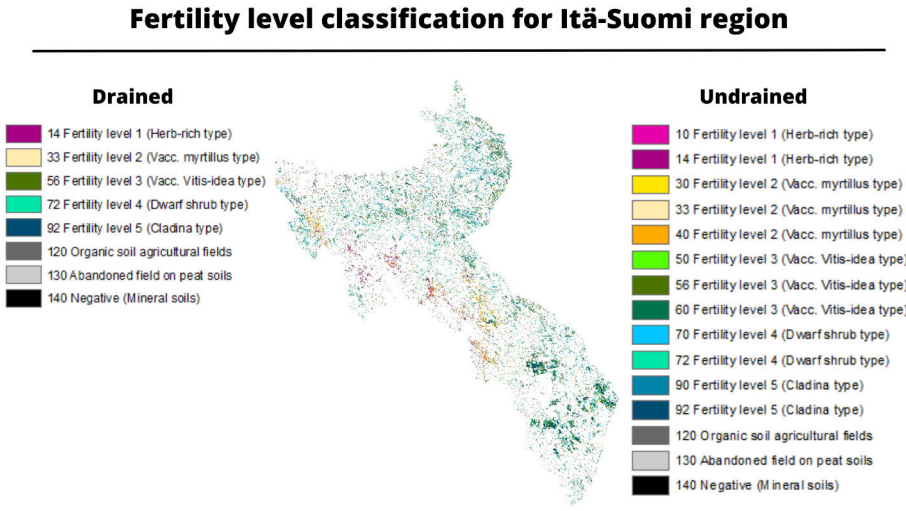

Figure 17. Eastern Finland Peatland. Drained and undrained section of the Peat are combined.

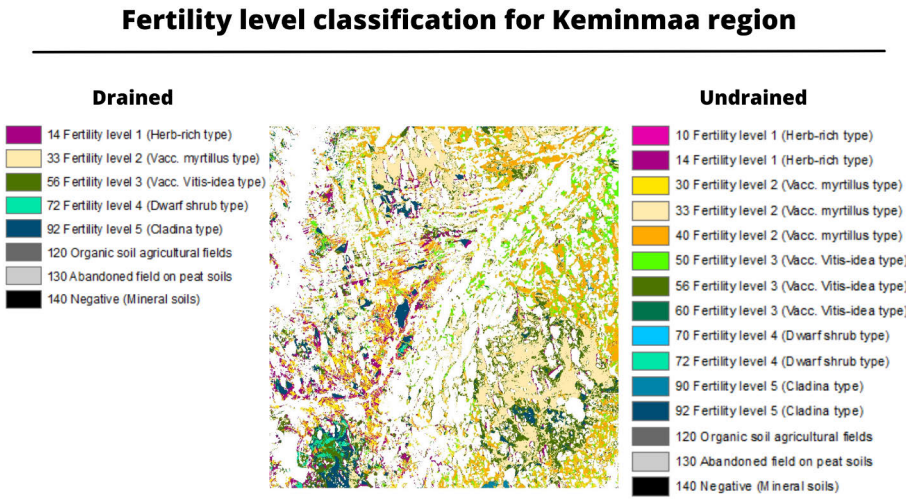

Figure 18. Keminmaa region peatland. Drained and undrained section of the Peat are combined.

fusion CNN architecture able to deal with different spatial resolution data and few number of labeled data. The article contributes to realising an efficient sensor fusion CNN which combines different input remote sensing data. The late fusion helps to avoid the problem of geo-referencing the pixel in the wrong place. This means that every input of the CNN points to the same pixel when using rasters of different spatial resolutions. The feature extraction helps to improve the training process before creating the maps. The feature extraction keeps only the important features of input dataset, removing the unimportant ones. Using a sliding windows approach, each part of the prediction area is predicted based on the neighboring information in order to reduce time and computing complexities.

Author's contribution

The author of this thesis was in charge of the design of the methodologies of the entire work. The design, evaluation and Python code have similarities with the previous Article 4. The Python code created has four main Sections. 1) Dataset creation; 2) Input selection; 3) Pixel-wise peatland classification for site type; 4) Derivation of Peatland for fertility level. In this article, the Python code for the 3) was able to produce undrained and drained peatland at the same time. In Article 4, for the same operation, there are two separate runs needed. The author participated in the evaluation of the results. He participates in writing the Section regarding the methodologies of the article.

6 Conclusion

6.1 Summary of the thesis

The introduction in Chapter 1 started by giving a brief introduction about the importance of the data and how information can be derived from it. It also discussed the potential of AI to make predictions in maritime environments and in remote sensing. The two case scenarios of this thesis were described as well.

In Chapter 2, a theoretical background of the theis was reviewed. The Chapter started with a discussion about pattern recognition, which can be either classical or more modern deep learning based. In the classical pattern recognition section, the process of feature extraction, feature selection and classical pattern recognition approaches were described. The main components of DL were reported in the chapter. The DL section also examines some of the CNN detectors. The chapter concluded with model building and performance evaluation. Classical pattern recognition is based on the manual definition of features extracted from the data, and classification is based on the acquired features set, whereas in DL, the process of feature extraction is conducted automatically by the DL model. A description of the differences between supervised, unsupervised, and semi-supervised pattern recognition was given as well. Chapter 2 dealt with the sensors fusion topic. Three disciplines of pattern recognition were described: 1) Classical approach, 2) DL, and 3) Sensor fusion.

Chapter 3 covered topics related to the maritime environment. It started with the Situational Awareness modelling (SA). The SA components are divided into perception of the environment, understanding of the situation, and predictive SA model. The Chapter included a description of potential hardware devices used to perceive the environment. Situational understanding deals with how the data provided by sensors can be used to understand the external environment and how sensor fusion can be used to improve performance in this stage. The predictive SA model section used the scenarios of the previous step and tries to predict the future outcome to a certain time ahead. The chapter continued with a maritime literature review, and by describing the maritime datasets used in this thesis.

Chapter 4 explored how artificial intelligence, especially machine learning, can be

applied in remote sensing. The section also described principal remote sensing data sources with their main properties. Satellites are not the only sources for remote sensing. For instance, from Airborne Laser Scanning, DEM and CHM can be easily derived, showing the digital elevation model of the land and the canopy high model, respectively. The role of sensors and data fusion used to enhance the level of accuracy of a CNN architecture was described.

Chapter 5 presented the articles included in this thesis. For each article summary, methods and data, results, contribution of the article and author's contributions to the work were included.

6.2 Discussion and outcome

AI-driven systems are becoming common tools in remote sensing and in maritime environment. Pattern recognition identifies patterns and regularities in the data. The data are initially analysed, and the features with high grades of importance are selected. Supervised or unsupervised models may be used to produce results. The results are evaluated. Data availability is increasing constantly. Powerful DL models are adopted to recognise objects, locate objects or make a segmentation out of them. The goal of pattern recognition, DL, is to generate a predictive model with a high grade of generalisation.

Domain-specific datasets increase the performance of models in specific domains, such as maritime environments. CNN models such as Faster-RCNN, RFCN, and SSD have the advantage of being reliable in detecting objects in maritime environments, and they automate the feature extraction process. The performance of these models can be further improved with transfer learning that transfers knowledge accumulated with a general-purpose dataset to a specific domain. Powerful CNN architectures can be trained with this data. Apart from transfer learning, in remote sensing, fusing multiple input sources into a CNN architecture enhances the model's predictive capabilities. In remote sensing, sensor fusion architectures are implemented to generate pixel-wise peatland classification which describes the characteristics of the analysed region. The performance of the model can be evaluated using cross-validation. The stratified K-Fold is a cross-validation algorithm that creates a randomised training and validation dataset, Its use greatly improves the generalisation of the model and prevents erroneous overfitting or underfitting.

To summarize the finding of this thesis in articles 1, 2, and 3 indicates that Faster-RCNN is very reliable in object classification and localization in the maritime environment. Especially in article number 3, transfer learning improved the prediction performance. In a maritime environment, it is crucial to distinguish between objects'

dimensions. The mAP (mean Average Precision) shows that the size of the objects heavily affects the recognition. The dataset created in article number 2 increased the performance in real-time maritime object detection in inshore and offshore environments. Faster RCNN was the best performing detector. However, for small or far away objects, EfficientDet achieved better results. A maritime-specific domain dataset with accurate manual annotations was created as well. The dataset contains 11 types of objects, and it was validated using the Channel and Spatial Reliability Tracker (CSRT) [198] in order to maximize the dataset's accuracy. The maritime dataset was tested with different detectors and feature extractors. The average precision was calculated according to the dimension and distance of the detected objects.

Articles 4 and 5 proposed a DL architecture applied to heterogeneous multi-resolution RS data. In RS, available data is high-dimensional and imbalanced. Feature selection is applied to avoid overfitting and to capture the relevant features. Articles 4 and 5 show that drawing a window around every data point, and this point is allocated in the centre of the window, helps to avoid overfitting. Moreover, the adoption of the Sequential Forward Selection (SFS) gradually includes inputs that maximize the accuracy in the features array. Every iteration uses five folds of Stratified K-Fold to generate random training and testing split. The task of each fold is to address the imbalanced class problem. The late fused CNN architecture adopted in these articles shows that the performance of the recognition increased for land cover classification.

List of References

[1] Jinzhu Lu, Lijuan Tan, and Huanyu Jiang. Review on convolutional neural network (cnn) applied to plant leaf disease classification. *Agriculture*, 11(8):707, 2021.

[2] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11), 2021.

[3] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Overview of supervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, pages 9–41, 2009.

[4] Zoubin Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer, 2003.

[5] Aaron E Maxwell, Timothy A Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: An applied review. *International journal of remote sensing*, 39 (9):2784–2817, 2018.

[6] B Mselmi, Zouhaier Ben Rabah, Imed Riadh Farah, and Basel Solaiman. Multi-resolution and multi-spectral analysis for satellite images classification with fuzzy spatial relationships. In *International Image Processing, Applications and Systems Conference*, pages 1–6. IEEE, 2014.

[7] Sicheng Zhu, Bang An, and Furong Huang. Understanding the generalization benefit of model invariance from a data perspective. *Advances in Neural Information Processing Systems*, 34: 4328–4341, 2021.

[8] FG Dias, JF Neves, VP da Conceição, and VJAS Lobo. Maritime situational awareness, the singular approach of a dual-use navy. *Scientific Bulletin" Mircea cel Batran" Naval Academy*, 21(1):1–14, 2018.

[9] Tolga Ahmet Gülcan and Kadir Emrah Erginer. National and international maritime situational awareness model examples and the effects of north stream pipelines sabotage. *International Journal of Critical Infrastructure Protection*, 42:100624, 2023.

[10] Qiang Zhu, Ke Ma, Zhong Wang, and Peibei Shi. Yolov7-csaw for maritime target detection. *Frontiers in neurorobotics*, 17, 2023.

[11] Richard Lindsay. Peatland classification. 2016.

[12] Valerie Thomas, Paul Treitz, Dennis Jelinski, John Miller, Peter Lafleur, and J Harry McCaughey. Image classification of a northern peatland complex using spectral and plant community data. *Remote Sensing of Environment*, 84(1):83–99, 2003.

[13] E Brown, M Aitkenhead, R Wright, and IH Aalders. Mapping and classification of peatland on the isle of lewis using landsat etm+. *Scottish Geographical Journal*, 123(3):173–192, 2007.

[14] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021.

[15] Wilfried Elmenreich. An introduction to sensor fusion. *Vienna University of Technology, Austria*, 502:1–28, 2002.

[16] Giuseppe Scarpa, Massimiliano Gargiulo, Antonio Mazza, and Raffaele Gaetano. A cnn-based fusion method for feature extraction from sentinel data. *Remote Sensing*, 10(2):236, 2018.

[17] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[18] Oliver Kramer and Oliver Kramer. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*, pages 13–23, 2013.

[19] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Abubakar Malah Umar, Okafor Uchenwa Linus, Humaira Arshad, Abdullahi Aminu Kazaure, Usman Gana, and Muhammad Ubale Kiru. Comprehensive review of artificial neural network applications to pattern recognition. *IEEE access*, 7:158820–158846, 2019.

[20] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

[21] Raghunathan Rengaswamy and Venkat Venkatasubramanian. A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engineering Applications of Artificial Intelligence*, 8(1):35–51, 1995. ISSN 0952-1976. doi: https://doi.org/10.1016/0952-1976(94)00058-U.

[22] GS Cox and G De Jager. A survey of point pattern matching techniques and a new approach to point pattern recognition. In *Proceedings of the 1992 South African Symposium on Communications and Signal Processing*, pages 243–248. IEEE, 1992.

[23] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[24] Tingting Cheng and Xianquan Zhan. Pattern recognition for predictive, preventive, and personalized medicine in cancer. *EPMA Journal*, 8:51–60, 2017.

[25] Antonio Moreda-Pineiro, Andrew Fisher, and Steve J Hill. The classification of tea according to region of origin using pattern recognition techniques and trace metal data. *Journal of Food Composition and analysis*, 16(2):195–211, 2003.

[26] J. Daugman. Face and gesture recognition: overview. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):675–676, 1997. doi: 10.1109/34.598225.

[27] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015.

[28] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.

[29] Christoph Schröer, Felix Kruse, and Jorge Marx Gómez. A systematic literature review on applying crisp-dm process model. *Procedia Computer Science*, 181:526–534, 2021.

[30] Gunter Ritter and María Teresa Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18(6):525–539, 1997. ISSN 0167-8655. doi: https://doi.org/10.1016/S0167-8655(97)00049-4.

[31] *Mahalanobis Distance*, pages 325–326. Springer New York, New York, NY, 2008. ISBN 978-0-387-32833-1. doi: 10.1007/978-0-387-32833-1_240.

[32] Kostantinos Koutroumbas Sergio Theodoridis. Pattern recognition. 2017. ISBN 978-1-59749-272-0.

[33] A Ferraz, E Esposito, RE Bruns, and N Durán. The use of principal component analysis (pca) for pattern recognition in eucalyptus grandis wood biodegradation experiments. *World Journal of Microbiology and Biotechnology*, 14:487–490, 1998.

[34] Joao Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa. Ensemble approaches for regression: A survey. *Acm computing surveys (csur)*, 45(1):1–40, 2012.

[35] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[36] Annu Lambora, Kunal Gupta, and Kriti Chopra. Genetic algorithm-a literature review. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 380–384. IEEE, 2019.

[37] Tae-Hwy Lee, Aman Ullah, and Ran Wang. Bootstrap aggregating and random forest. *Macroeconomic forecasting in the era of big data: Theory and practice*, pages 389–429, 2020.

[38] Robert E Schapire. The boosting approach to machine learning: An overview. *Nonlinear estimation and classification*, pages 149–171, 2003.

[39] Bo-Kyeong Kim, Hwaran Lee, Jihyeon Roh, and Soo-Young Lee. Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pages 427–434, 2015.

[40] Linlin Shen and Li Bai. A review on gabor wavelets for face recognition. *Pattern analysis and applications*, 9:273–292, 2006.

[41] Marco Bressan, David Guillamet, and Jordi Vitria. Using an ica representation of local color histograms for object recognition. *Pattern Recognition*, 36(3):691–701, 2003.

[42] David Mendlovic, Zeev Zalevsky, and Haldun M Ozaktas. Applications of the fractional fourier transform to optical pattern recognition. *Optical pattern recognition*, pages 89–125, 1998.

[43] Himanshu Gothwal, Silky Kedawat, Rajesh Kumar, et al. Cardiac arrhythmias detection in an ecg beat signal using fast fourier transform and artificial neural network. *Journal of Biomedical Science and Engineering*, 4(04):289, 2011.

[44] Hongxun zhang and De xu. Fusing color and texture features for background model. In Lipo Wang, Licheng Jiao, Guanming Shi, Xue Li, and Jing Liu, editors, *Fuzzy Systems and Knowledge Discovery*, pages 887–893, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-45917-0.

[45] Matti Pietikäinen. Local binary patterns. *Scholarpedia*, 5(3):9775, 2010.

[46] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314.

[47] Dian Candra Rini Novitasari, Ahmad Lubab, Asri Sawiji, and Ahmad Hanif Asyhar. Application of feature extraction for breast cancer using one order statistic, glcm, glrlm, and gldm. *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*, 4(4):115–120, 2019.

[48] Diah Ayu Larasati et al. Application of the k-nn method and glcm feature extraction in classifying formalin fish images. *Journal Of Research Computer Science*, 1(1):1–13, 2021.

[49] Mary M Galloway. Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2):172–179, 1975.

[50] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*, pages 469–481. Springer, 2004.

[51] T Vigneshl and KK Thyagharajan. Local binary pattern texture feature for satellite imagery classification. In *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, pages 1–6. IEEE, 2014.

[52] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2): 103–113, 1989.

[53] Yushi Chen, Lin Zhu, Pedram Ghamisi, Xiuping Jia, Guoyu Li, and Liang Tang. Hyperspectral images classification with gabor filtering and convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2355–2359, 2017.

[54] Cédric Lardeux, Pierre-Louis Frison, Céline Tison, Jean-Claude Souyris, Benoît Stoll, Bénédicte Fruneau, and Jean-Paul Rudant. Support vector machine for multifrequency sar polarimetric data classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(12):4143–4152, 2009.

[55] Dugal Harris and Adriaan Van Niekerk. Feature clustering and ranking for selecting stable features from high dimensional remotely sensed data. *International Journal of Remote Sensing*, 39(23):8934–8949, 2018.

[56] Ioannis Tsamardinos, Giorgos Borboudakis, Pavlos Katsogridakis, Polyvios Pratikakis, and Vassilis Christophides. A greedy feature selection algorithm for big data of high dimensionality. *Machine learning*, 108:149–202, 2019.

[57] Vipin Kumar and Sonajharia Minz. Feature selection: a literature review. *SmartCR*, 4(3):211–229, 2014.

[58] Jian-Bo Yang, Kai-Quan Shen, Chong-Jin Ong, and Xiao-Ping Li. Feature selection for mlp neural network: The use of random permutation of probabilistic outputs. *IEEE Transactions on Neural Networks*, 20(12):1911–1922, 2009.

[59] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[60] Huda Hamdan Ali and Lubna Emad Kadhum. K-means clustering algorithm applications in data mining and pattern recognition. *International Journal of Science and Research (IJSR)*, 6(8): 1577–1584, 2017.

[61] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[62] Taehwan Kim and Tülay Adalı. Approximation by fully complex multilayer perceptrons. *Neural computation*, 15(7):1641–1666, 2003.

[63] Pramod Gupta and Naresh K. Sinha. Neural networks for identification of nonlinear systems: An overview. In *Soft Computing and Intelligent Systems*, Academic Press Series in Engineering, pages 337–356. Academic Press, San Diego, 2000. ISBN 978-0-12-646490-0. doi: https://doi.org/10.1016/B978-012646490-0/50017-2.

[64] Siddharth Misra, Hao Li, and Jiabo He. Robust geomechanical characterization by analyzing the performance of shallow-learning regression methods using unsupervised clustering methods. In Siddharth Misra, Hao Li, and Jiabo He, editors, *Machine Learning for Subsurface Characterization*, pages 129–155. Gulf Professional Publishing, 2020. ISBN 978-0-12-817736-5. doi: https://doi.org/10.1016/B978-0-12-817736-5.00005-3.

[65] Daniel T Larose and Chantal D Larose. k-nearest neighbor algorithm. 2014.

[66] Zhu Fan, Jia-kun Xie, Zhong-yu Wang, Pei-Chen Liu, Shu-jun Qu, and Lei Huo. Image classification method based on improved knn algorithm. In *Journal of physics: Conference series*, volume 1930, page 012009. IOP Publishing, 2021.

[67] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[68] Shan Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[69] Derek A Pisner and David M Schnyer. Support vector machine. In *Machine learning*, pages 101–121. Elsevier, 2020.

[70] Barry De Ville. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6): 448–455, 2013.

[71] Xiaosheng Peng, Jinshu Li, Ganjun Wang, Yijiang Wu, Lee Li, Zhaohui Li, Ashfaque Ahmed Bhatti, Chengke Zhou, Donald M Hepburn, Alistair J Reid, et al. Random forest based optimal feature selection for partial discharge pattern recognition in hv cables. *IEEE Transactions on Power Delivery*, 34(4):1715–1724, 2019.

[72] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[73] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

[74] Yu Guo, Yuxu Lu, Ryan Wen Liu, Meifang Yang, and Kwok Tai Chui. Low-light image enhancement with regularized illumination optimization and deep noise suppression. *IEEE Access*, 8: 145297–145315, 2020. doi: 10.1109/ACCESS.2020.3015217.

[75] Lei Rao, Bin Zhang, and Jizhong Zhao. An energy-efficient accelerator for rain removal based on convolutional neural network. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(8):2957–2961, 2021. doi: 10.1109/TCSII.2021.3071455.

[76] Dongwei Ren, Wei Shang, Pengfei Zhu, Qinghua Hu, Deyu Meng, and Wangmeng Zuo. Single image deraining using bilateral recurrent network. *IEEE Transactions on Image Processing*, 29: 6852–6863, 2020. doi: 10.1109/TIP.2020.2994443.

[77] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):3943–3956, 2020. doi: 10.1109/TCSVT.2019.2920407.

[78] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104:154–171, 2013.

[79] Xiaoyuan Ji, Qiuyu Yan, Dong Huang, Bo Wu, Xiaojing Xu, Aibin Zhang, Guanglan Liao, Jianxin Zhou, and Menghuai Wu. Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition. *Journal of Materials Processing Technology*, 292:117064, 2021.

[80] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

[81] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[82] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2606–2615, 2017.

[83] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

[84] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[85] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[86] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[87] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[88] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8828–8838, 2020.

[89] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, and Andrea Mechelli. Autoencoders. In *Machine learning*, pages 193–208. Elsevier, 2020.

[90] Yu Weng, Tianbao Zhou, Yujie Li, and Xiaoyu Qiu. Nas-unet: Neural architecture search for medical image segmentation. *IEEE access*, 7:44247–44257, 2019.

[91] Luca Zelioli. Environmental damage assessment based on satellite imagery using machine learning. 2020.

[92] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[93] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

[94] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning, 2018.

[95] Xuelong Li, Kai Kou, and Bin Zhao. Weather gan: Multi-domain weather translation using generative adversarial networks. *arXiv preprint arXiv:2103.05422*, 2021.

[96] Hua Wang, Cuiqin Ma, and Lijuan Zhou. A brief review of machine learning and its application. In *2009 international conference on information engineering and computer science*, pages 1–4. IEEE, 2009.

[97] D. Partridge. Artificial intelligence. In *Reference Module in Neuroscience and Biobehavioral Psychology*. Elsevier, 2017. ISBN 978-0-12-809324-5. doi: https://doi.org/10.1016/B978-0-12-809324-5.02995-3.

[98] Zelda B Zabinsky et al. Random search algorithms. *Department of Industrial and Systems Engineering, University of Washington, USA*, 2009.

[99] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[100] Davide Anguita, Luca Ghelardoni, Alessandro Ghio, Luca Oneto, and Sandro Ridella. The'k'in k-fold cross validation. In *ESANN*, pages 441–446, 2012.

[101] Hiroshi Konno and Tomoyuki Koshizuka. Mean-absolute deviation model. *Iie Transactions*, 37 (10):893–900, 2005.

[102] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific model development discussions*, 7(1):1525–1534, 2014.

[103] David M Allen. Mean square error of prediction as a criterion for selecting variables. *Technometrics*, 13(3):469–475, 1971.

[104] Matthias Kohl. Performance measures in binary classification. *International Journal of Statistics in Medical Research*, 1(1):79, 2012.

[105] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de Las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

[106] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[107] Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2): 145–151, 2008.

[108] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. Multilevel sensor fusion with deep learning. *IEEE sensors letters*, 3(1):1–4, 2018.

[109] Jevon P Chan, Rose Norman, Kayvan Pazouki, and David Golightly. Autonomous maritime operations and the influence of situational awareness within maritime navigation. *WMU Journal of Maritime Affairs*, 21(2):121–140, 2022.

[110] Amit Sharma, Salman Nazir, and Jorgen Ernstsen. Situation awareness information requirements for maritime navigation: A goal directed task analysis. *Safety Science*, 120:745–752, 2019.

[111] Abbas Harati-Mokhtari, Alan Wall, Philip Brooks, and Jin Wang. Automatic identification system (ais): data reliability and human error implications. *the Journal of Navigation*, 60(3):373–389, 2007.

[112] Aparna Akula and Harish Kumar Sardana. Deep cnn-based feature extractor for target recognition in thermal images. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 2370–2375. IEEE, 2019.

[113] Zachary Baird, Michael K Mcdonald, Sreeraman Rajan, and Simon J Lee. A cnn-lstm network for augmenting target detection in real maritime wide area surveillance radar data. *IEEE Access*, 8:179281–179294, 2020.

[114] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5):741, 2020.

[115] Dilip K Prasad, C Krishna Prasath, Deepu Rajan, Lily Rachmawati, Eshan Rajabaly, and Chai Quek. Challenges in video based object detection in maritime scenario using computer vision. *arXiv preprint arXiv:1608.01079*, 2016.

[116] Jihao Shi, Yuanjiang Chang, Changhang Xu, Faisal Khan, Guoming Chen, and Chuangkun Li. Real-time leak detection using an infrared camera and faster r-cnn technique. *Computers & Chemical Engineering*, 135:106780, 2020.

[117] Shih-Cheng Huang, Anuj Pareek, Saeed Seyyedi, Imon Banerjee, and Matthew P Lungren. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *NPJ digital medicine*, 3(1):136, 2020.

[118] Yun Zhang. Understanding image fusion. *Photogramm. Eng. Remote Sens*, 70(6):657–661, 2004.

[119] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd international conference on information fusion (FUSION)*, pages 1–6. IEEE, 2020.

[120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

[121] Ruolan Zhang, Shaoxi Li, Guanfeng Ji, Xiuping Zhao, Jing Li, and Mingyang Pan. Survey on deep learning-based marine object detection. *Journal of Advanced Transportation*, 2021:1–18, 2021.

[122] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[123] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL `http://arxiv.org/abs/1405.0312`.

[124] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[125] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[126] Sung-Jun Lee, Myung-Il Roh, Hye-Won Lee, Ji-Sang Ha, and Il-Guk Woo. Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks. In *The 28th International Ocean and Polar Engineering Conference*. OnePetro, 2018.

[127] Zhenfeng Shao, Wenjing Wu, Zhongyuan Wang, Wan Du, and Chengyuan Li. Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE transactions on multimedia*, 20(10): 2593–2604, 2018.

[128] Y. Zheng and S. Zhang. Mcships: A large-scale ship dataset for detection and fine-grained categorization in the wild. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020. doi: 10.1109/ICME46284.2020.9102907.

[129] Bogdan Iancu, Valentin Soloviev, Luca Zelioli, and Johan Lilius. Aboships—an inshore and offshore maritime vessel detection dataset with precise annotations. *Remote Sensing*, 13(5):988, 2021.

[130] Dalei Qiao, Guangzhong Liu, Taizhi Lv, Wei Li, and Juan Zhang. Marine vision-based situational awareness using discriminative deep learning: A survey. *Journal of Marine Science and Engineering*, 9(4):397, 2021.

[131] Chamali Gamage, Randima Dinalankara, Jagath Samarabandu, and Akila Subasinghe. A comprehensive survey on the applications of machine learning techniques on maritime surveillance to detect abnormal maritime vessel behaviors. *WMU Journal of Maritime Affairs*, pages 1–31, 2023.

[132] Kutluyil Dogancay, Ziming Tu, and Gokhan Ibal. Research into vessel behaviour pattern recognition in the maritime domain: Past, present and future. *Digital Signal Processing*, 119:103191, 2021.

[133] Aaron Hunter. Belief modeling for maritime surveillance. In *2009 12th International Conference on Information Fusion*, pages 1926–1932. IEEE, 2009.

[134] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[135] Vladimir Avram, Uwe Glässer, and Hamed Yaghoubi Shahir. Anomaly detection in spatiotemporal data in the maritime domain. In *2012 IEEE International Conference on Intelligence and Security Informatics*, pages 147–149, 2012. doi: 10.1109/ISI.2012.6284274.

[136] Sergiy Fefilatyev, Dmitry Goldgof, Matthew Shreve, and Chad Lembke. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*, 54: 1–12, 2012.

[137] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5): 1049–1058, 2014.

[138] J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):690–698, 1987. doi: 10.1109/TPAMI.1987.4767964.

[139] Dilip K. Prasad, Deepu Rajan, Lily Rachmawati, Eshan Rajabally, and Chai Quek. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):1993–2016, 2017. doi: 10.1109/TITS.2016.2634580.

[140] Dilip K Prasad, Chandrashekar Krishna Prasath, Deepu Rajan, Lily Rachmawati, Eshan Rajabally, and Chai Quek. Object detection in a maritime environment: Performance evaluation of background subtraction methods. *IEEE Transactions on Intelligent Transportation Systems*, 20 (5):1787–1802, 2018.

[141] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[142] Sander Soo. Object detection using haar-cascade classifier. *Institute of Computer Science, University of Tartu*, 2(3):1–12, 2014.

[143] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[144] Mohammad-Hashem Haghbayan, Fahimeh Farahnakian, Jonne Poikonen, Markus Laurinen, Paavo Nevalainen, Juha Plosila, and Jukka Heikkonen. An efficient multi-sensor fusion approach for object detection in maritime environments. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2163–2170, 2018. doi: 10.1109/ITSC.2018.8569890.

[145] Filippo Sanfilippo. A multi-sensor fusion framework for improving situational awareness in demanding maritime training. *Reliability Engineering System Safety*, 161:12–24, 2017. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2016.12.015.

[146] F. Sanfilippo and K. Y. Pettersen. Xbee positioning system with embedded haptic feedback for dangerous offshore operations: A preliminary study. In *OCEANS 2015 - Genova*, pages 1–6, 2015. doi: 10.1109/OCEANS-Genova.2015.7271241.

[147] Fouad Bousetouane and Brendan Morris. Fast cnn surveillance pipeline for fine-grained vessel classification and detection in maritime scenarios. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 242–248. IEEE, 2016.

[148] Gaurav Verma, Aditya Gupta, Shobhit Bansal, and Himanshu Dhiman. Monitoring maritime traffic with ship detection via yolov4. In *2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)*, pages 1–7. IEEE, 2022.

[149] Muhammad Bilal and Muhammad Shehzad Hanif. Benchmark revision for hog-svm pedestrian detector through reinvigorated training and evaluation methodologies. *IEEE transactions on intelligent transportation systems*, 21(3):1277–1287, 2019.

[150] Sung Won Moon, Jiwon Lee, Jungsoo Lee, Dowon Nam, and Wonyoung Yoo. A comparative study on the maritime object detection performance of deep learning models. In *2020 international conference on information and communication technology convergence (ICTC)*, pages 1155–1157. IEEE, 2020.

[151] Huang Zhijun and SANG Qingbing. Ship detection based on improved r-fcn. *Journal of Frontiers of Computer Science & Technology*, 14(6):1045, 2020.

[152] Mostafa Hamdy Salem, Yujian Li, Zhaoying Liu, and Ahmed M AbdelTawab. A transfer learning and optimized cnn based maritime vessel classification system. *Applied Sciences*, 13(3):1912, 2023.

[153] Md Samiur Rahman Bhuiya, Nazmul Islam, Ayesha Siddiqua Drishty, Utsha Das Akash, Snigdha Suparna Saha, Amitabha Chakrabarty, and Shahriar Hossain. Surveillance in maritime

scenario using deep learning and swarm intelligence. In *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pages 569–574. IEEE, 2022.

[154] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[155] Qasem Abu Al-Haija and Adeola Adebanjo. Breast cancer diagnosis in histopathological images using resnet-50 convolutional neural network. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–7. IEEE, 2020.

[156] Yi Zhu and Shawn Newsam. Densenet for dense flow. In *2017 IEEE international conference on image processing (ICIP)*, pages 790–794. IEEE, 2017.

[157] Antonio-Javier Gallego, Antonio Pertusa, Pablo Gil, and Robert B Fisher. Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras. *Journal of Field Robotics*, 36(4):782–796, 2019.

[158] Mostafa Hamdy Salem, Yujian Li, and Zhaoying Liu. Transfer learning on efficientnet for maritime visible image classification. In *2022 7th International Conference on Signal and Image Processing (ICSIP)*, pages 514–520. IEEE, 2022.

[159] Sergey Voinov, Detmar Krause, and Egbert Schwarz. Towards automated vessel detection and type recognition from vhr optical satellite images. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 4823–4826. IEEE, 2018.

[160] Kun Liu, Shengtao Yu, and Sidong Liu. An improved inceptionv3 network for obscured ship classification in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4738–4747, 2020.

[161] Aaron E Maxwell, Timothy A Warner, Brian C Vanderbilt, Christopher A Ramezan, et al. Land cover classification and feature extraction from national agriculture imagery program (naip) orthoimagery: a review. *PE&RS, Photogrammetric Engineering & Remote Sensing*, 83(11):737–747, 2017.

[162] John Townshend, Christopher Justice, Wei Li, Charlotte Gurney, and Jim McManus. Global land cover classification by remote sensing: present capabilities and future possibilities. *Remote Sensing of Environment*, 35(2-3):243–255, 1991.

[163] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733*, 2021.

[164] Michael Bruno, Alexander Sutin, Kil Woo Chung, Alexander Sedunov, Nikolay Sedunov, Hady Salloum, Hans Graber, and Paul Mallas. Satellite imaging and passive acoustics in layered approach for small boat detection and classification. *Marine Technology Society Journal*, 45(3), 2011.

[165] Monique Poulin, Denis Careau, Line Rochefort, and André Desrochers. From satellite imagery to peatland vegetation diversity: how reliable are habitat maps? *Conservation ecology*, 6(2), 2002.

[166] Vikas Sharma, Diganta Baruah, Dibyajyoti Chutia, PLN Raju, and DK Bhattacharya. An assessment of support vector machine kernel parameters using remotely sensed satellite data. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1567–1570. IEEE, 2016.

[167] Hasnat Khurshid and Muhammad Faisal Khan. Segmentation and classification using logistic regression in remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(1):224–232, 2014.

[168] Fahimeh Farahnakian and Jukka Heikkonen. Deep learning based multi-modal fusion architectures for maritime vessel detection. *Remote Sensing*, 12(16):2509, 2020.

[169] Katherine Irwin, Danielle Beaulne, Alexander Braun, and Georgia Fotopoulos. Fusion of sar, optical imagery and airborne lidar for surface water detection. *Remote Sensing*, 9(9):890, 2017.

[170] Wei Zhang, Ping Tang, and Lijun Zhao. Remote sensing image scene classification using cnn-capsnet. *Remote Sensing*, 11(5):494, 2019.

[171] Wei Zhang, Ping Tang, and Lijun Zhao. Remote sensing image scene classification using cnn-capsnet. *Remote Sensing*, 11(5):494, 2019.

[172] Kavita Bhosle and Vijaya Musande. Evaluation of deep learning cnn model for land use land cover classification and crop identification using hyperspectral remote sensing images. *Journal of the Indian Society of Remote Sensing*, 47(11):1949–1958, 2019.

[173] Behnood Rasti and Pedram Ghamisi. Remote sensing image classification using subspace sensor fusion. *Information Fusion*, 64:121–130, 2020.

[174] D Valsesia. Enhancing satellite imagery with deep multi-temporal superresolution. 2019.

[175] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

[176] Ranganath R Navalgund, V Jayaraman, and PS Roy. Remote sensing applications: An overview. *current science*, pages 1747–1766, 2007.

[177] Kaylee D Hakkel, Maurangelo Petruzzella, Fang Ou, Anne van Klinken, Francesco Pagliano, Tianran Liu, Rene PJ van Veldhoven, and Andrea Fiore. Integrated near-infrared spectral sensing. *Nature communications*, 13(1):103, 2022.

[178] Peter M Atkinson and Paul J Curran. Choosing an appropriate spatial resolution for remote sensing investigations. *Photogrammetric engineering and remote sensing*, 63(12):1345–1351, 1997.

[179] Gong Jianya, Sui Haigang, Ma Guorui, and Zhou Qiming. A review of multi-temporal remote sensing data change detection algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B7):757–762, 2008.

[180] Lei Yan, Taixia Wu, and Xueqi Wang. Polarization remote sensing for land observation. In Lei-Ming Ma, Zhang Chang-Jiang, and Feng Zhang, editors, *Understanding of Atmospheric Systems with Efficient Numerical Methods for Observation and Prediction*, chapter 3. IntechOpen, Rijeka, 2018. doi: 10.5772/intechopen.79937. URL `https://doi.org/10.5772/intechopen.79937`.

[181] Vincent Cazaubiel, Vincent Chorvalli, and Christophe Miesch. The multispectral instrument of the sentinel2 program. In *International Conference on Space Optics—ICSO 2008*, volume 10566, pages 110–115. SPIE, 2017.

[182] András Gulácsi and Ferenc Kovács. Sentinel-1-imagery-based high-resolution water cover detection on wetlands, aided by google earth engine. *Remote Sensing*, 12(10):1614, 2020.

[183] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27, 2017.

[184] Stefan Buckreuss, Birgit Schättler, Thomas Fritz, Josef Mittermayer, Ralph Kahle, Edith Maurer, Johannes Böer, Markus Bachmann, Falk Mrowka, Egbert Schwarz, et al. Ten years of terrasar-x operations. *Remote Sensing*, 10(6):873, 2018.

[185] Matthew Montanaro, Joel McCorkel, June Tveekrem, John Stauder, Eric Mentzell, Allen Lunsford, Jason Hair, and Dennis Reuter. Landsat 9 thermal infrared sensor 2 (tirs-2) stray light mitigation and assessment. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–8, 2022. doi: 10.1109/TGRS.2022.3177312.

[186] Michael A Wulder, Thomas R Loveland, David P Roy, Christopher J Crawford, Jeffrey G Masek, Curtis E Woodcock, Richard G Allen, Martha C Anderson, Alan S Belward, Warren B Cohen, et al. Current status of landsat program, science, and applications. *Remote sensing of environment*, 225:127–147, 2019.

[187] Erkki Tomppo, Markus Haakana, Matti Katila, and Jouni Peräsaari. *Multi-source national forest inventory: Methods and applications*, volume 18. Springer Science & Business Media, 2008.

[188] Kalyani Kadam, Swati Ahirrao, Ketan Kotecha, and Sayan Sahu. Detection and localization of multiple image splicing using mobilenet v1. *IEEE Access*, 9:162499–162519, 2021.

[189] Siti Zulaikha Muhammad Zaki, Mohd Asyraf Zulkifley, Marzuraikah Mohd Stofa, Nor Azwan Mohammed Kamari, and Nur Ayuni Mohamed. Classification of tomato leaf diseases using mobilenet v2. *IAES International Journal of Artificial Intelligence*, 9(2):290, 2020.

[190] Derry Alamsyah and Muhammad Fachrurrozi. Faster r-cnn with inception v2 for fingertip detection in homogenous background image. In *Journal of Physics: Conference Series*, volume 1196, page 012017. IOP Publishing, 2019.

[191] Xu Qin and Zhilin Wang. Nasnet: A neuron attention stage-by-stage net for single image deraining. *arXiv preprint arXiv:1912.03151*, 2019.

[192] Brett Koonce and Brett Koonce. Resnet 50. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pages 63–72, 2021.

[193] Qi Zhang. A novel resnet101 model based on dense dilated convolution for image classification. *SN Applied Sciences*, 4:1–13, 2022.

[194] Khurshedjon Farkhodov, Suk-Hwan Lee, and Ki-Ryong Kwon. Object tracking using csrt tracker and rcnn. In *BIOIMAGING*, pages 209–212, 2020.

[195] Xurshedjon Farhodov, Oh-Heum Kwon, Kyung Won Kang, Suk-Hwan Lee, and Ki-Ryong Kwon. Faster rcnn detection based opencv csrt tracker using drone data. In *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, pages 1–3. IEEE, 2019.

[196] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[197] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. *arXiv preprint arXiv:2304.07288*, 2023.

[198] Yuanze Wang, Chenlu Liu, Sheng Li, Tong Wang, Weiyang Lin, and Xinghu Yu. A multi-target tracking algorithm for fast-moving workpieces based on event camera. In *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–5. IEEE, 2021.

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

Painosalama, Turku, Finland 2024

ISBN 978-951-29-9796-1 (PRINT)
ISBN 978-951-29-9797-8 (PDF)
ISSN 2736-9390 (PRINT)
ISSN 2736-9684 (ONLINE)