



**UiT** The Arctic University of Norway

Faculty of Science and Technology  
Department of Physics and Technology

## **Machine Learning for Classifying Marine Vegetation from Hyperspectral Drone Data in the Norwegian coast**

Silje B. S. Grue

FYS-3941 Master's thesis in Applied Physics and Mathematics 30 SP  
May 2022









# Abstract

Along the Norwegian coasts the presence of blue forests are the key marine habitats. Due to increased anthropogenic activity and climate change, the health and extent of the blue forests is threatened. However, no low-cost, reliable system for monitoring blue forests exists in Norway at this time. This thesis studied machine learning methods to classify marine vegetation from hyperspectral data acquired in Norway. The study area is situated by Larvik at Ølbergholmen. The dataset consists of 12 hyperspectral images with 173 spectral bands in the region 390 nm - 749 nm and corresponding labels of the different classes. This dataset was used to train and evaluate the machine learning methods. In addition, an independent dataset from a different site was used for robustness evaluation. Three machine learning methods were studied; Random Forest (RF), Support Vector Machines (SVM) and Convolutional Neural Network (CNN). The results indicate that the powerful CNN approach had the best performance during validation based on the computed statistical measures. However, when evaluated for robustness, RF performed the best. The computed confusion matrices for the validation and robustness studies revealed that the presence of a so-called turf algae caused difficulties in distinguishing between the classes, which is an important finding with regard to future research. This thesis has shown that machine learning can be used for monitoring blue forests and various marine vegetation species using hyperspectral drone imaging along the Norwegian coast.



# Acknowledgements

Throughout writing this thesis, I have received a great amount of support.

I would first like to thank my supervisor Katalin Blix, you have been the best supervisor I could have ever asked for. For the endless support, always being there for my questions and for pushing me to do my best. Thank you. I would also like to thank my co-supervisor, Martin Skjelvareid, for taking me on a field trip, discussing the methodology of the thesis and all your feedback. To the MASSIMAL project; thank you for the data and the exciting opportunity to write my masters thesis in collaboration with the project. Thanks for the fruitful conversation with Viktor R. Tóth at the Balaton Limnological Research Institute regarding the turf algae issues.

I would like to acknowledge my fellow students and the professors at UiT, who have made completing the degree during the Covid-19 pandemic possible.

I would like to thank my family, who have always been supportive and encouraged me to pursue a higher education. My friends, Marit, Marthe and Julie; thank you for the support and encouragement and for always being so understanding. My partner Thomas, thank you for always being so patient and supportive throughout all my years at the university, especially the last year when I worked a full time job while finishing my degree.

*Silje B. S. Grue,  
Tromsø, May 2022.*



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Structure of the thesis . . . . .	4
<b>2 Optical remote sensing of marine vegetation</b>	<b>5</b>
2.1 Electromagnetic radiation and blackbody radiation . . . . .	5
2.2 Light's pathway to the sensor . . . . .	8
2.3 Remote Sensing Reflectance and Water Leaving Radiance . .	11
2.4 Sun glint . . . . .	13
<b>3 Hyperspectral imaging</b>	<b>17</b>
<b>4 Blue forests</b>	<b>21</b>
<b>5 Algorithms</b>	<b>25</b>
5.1 Vegetation Indices . . . . .	25
5.2 Spectral Angle Mapping . . . . .	26
5.3 Machine Learning . . . . .	26
<b>6 Machine Learning for Marine Vegetation Mapping</b>	<b>27</b>
6.1 Machine Learning . . . . .	27
6.1.1 An introduction to Machine Learning . . . . .	27
6.1.2 Machine Learning and dimensionality reduction meth- ods used in this project . . . . .	33
6.2 Data acquisition . . . . .	41

6.2.1	Drone imaging . . . . .	41
6.2.2	The data . . . . .	44
6.3	Methodology . . . . .	51
6.3.1	Preprocessing . . . . .	52
6.3.2	Hyperparameter tuning . . . . .	53
6.3.3	Convolutional Neural Network . . . . .	54
6.3.4	Random Forest . . . . .	55
6.3.5	Support Vector Machine . . . . .	56
6.3.6	Statistical measures . . . . .	56
6.3.7	Algorithm application to images . . . . .	56
6.4	Results . . . . .	57
6.4.1	Dimensionality reduction . . . . .	57
6.4.2	Classification results . . . . .	60
6.4.3	Algorithms applied to images for visual presentation .	70
6.5	Discussion . . . . .	76
6.6	Conclusion . . . . .	80
<b>7</b>	<b>Appendix</b>	<b>83</b>
7.1	RF and SVM applied to additional image from the South side	83
7.2	Conferences and field work . . . . .	87
	<b>Bibliography</b>	<b>89</b>

# List of Figures

1.1	Illustration of Urchins and turf algae . . . . .	2
2.1	EM radiation - absorption and emittance . . . . .	7
2.2	Light's pathway to the sensor . . . . .	8
2.3	Absorption of EM radiation by the ocean . . . . .	9
2.4	Absorption spectrum of Chlorophyll-a, CDOM and water . . . . .	10
2.5	Change in reflectance spectrum with depth . . . . .	11
2.6	Turbid water vs. clear water . . . . .	13
2.7	Sun glint correction . . . . .	14
3.1	Hyperspectral data cube . . . . .	18
4.1	Different types of sea bottom vegetation . . . . .	21
4.2	Map of kelp forest along the Norwegian coast . . . . .	23
4.3	Reflectance spectra of three different types of sea grass . . . . .	24
6.1	Underfitting, Good fit and Overfitting . . . . .	29
6.2	Confusion matrix illustration . . . . .	30
6.3	Comparison of different ML systems . . . . .	32
6.4	Example of the architecture of a CNN . . . . .	33
6.5	Decision tree . . . . .	34
6.6	Support Vector Machine . . . . .	36
6.7	Convolution operation . . . . .	38
6.8	Field campaign . . . . .	42
6.9	Drone used in this project . . . . .	43
6.10	Example of underwater photo used for annotation . . . . .	45
6.11	Example of how the dataset is annotated . . . . .	45
6.12	Study site . . . . .	46
6.13	Transect map . . . . .	47
6.14	Images of the different classes . . . . .	49
6.15	Mean spectral response for the classes in the dataset . . . . .	50
6.16	Flowchart . . . . .	51
6.17	CNN architecture . . . . .	54
6.18	Study site - images . . . . .	57

6.19 Explained variance PCA . . . . .	58
6.20 Mean spectra and PCA vs wavelength . . . . .	59
6.21 Accuracy and loss for CNN for all classes . . . . .	61
6.22 Accuracy and loss for CNN for merged classes . . . . .	61
6.23 All classifiers applied to the image from the South, with all classes. . . . .	72
6.24 All classifiers applied to the image from the North, with all classes. . . . .	73
6.25 All classifiers applied to the image from the South, with merged classes. . . . .	74
6.26 All classifiers applied to the image from the North, with merged classes. . . . .	75
7.1 Map of where additional image is placed . . . . .	85
7.2 Established models applied on additional image . . . . .	86



# List of Tables

6.1	Classes in the dataset . . . . .	47
6.2	CNN architecture . . . . .	55
6.3	Statistics . . . . .	60
6.4	Confusion matrices validation, all classes . . . . .	64
6.5	Confusion matrices, image from the South, all classes . . . . .	65
6.6	Confusion matrices, image from the North, all classes . . . . .	66
6.7	Confusion matrices validation, merged classes . . . . .	67
6.8	Confusion matrices, image from the South, merged classes . . . . .	68
6.9	Confusion matrices, image from the North, merged classes . . . . .	69



# Abbreviations

<b>AA</b>	Average Accuracy
<b>CDOM</b>	Colored Dissolved Organic Matter
<b>CNN</b>	Convolutional Neural Network
<b>DL</b>	Deep Learning
<b>EA</b>	Expected Accuracy
<b>EM</b>	ElectroMagnetic
<b>FP</b>	False Positive
<b>FN</b>	False Negative
<b>HYPSO-1</b>	HYPer-spectral Satellite for Ocean observation-1
<b>MASSIMAL</b>	Mapping of Algae and Seagrass by using Spectral Imagining and Machine Learning
<b>ML</b>	Machine Learning
<b>NASA</b>	National Aeronautics and Space Administration
<b>NDVI</b>	Normalized Difference Vegetation Index
<b>NIR</b>	Near InfraRed
<b>NTNU</b>	Norwegian University of Science and Technology
<b>OA</b>	Overall Accuracy
<b>PACE</b>	Plankton, Aerosol, Cloud ocean Ecosystem
<b>PCA</b>	Principal Component Analysis
<b>PRISMA</b>	PRecurso IperSpettrale della Missione Applicativa
<b>RBF</b>	Radial Basis Function
<b>RF</b>	Random Forest
<b>RGB</b>	Red Green Blue
<b>SAM</b>	Spectral Angle Mapper
<b>SVM</b>	Support Vector Machine
<b>TP</b>	True Positive
<b>TN</b>	True Negative
<b>VI</b>	Vegetation Index
<b>VIS</b>	VISible

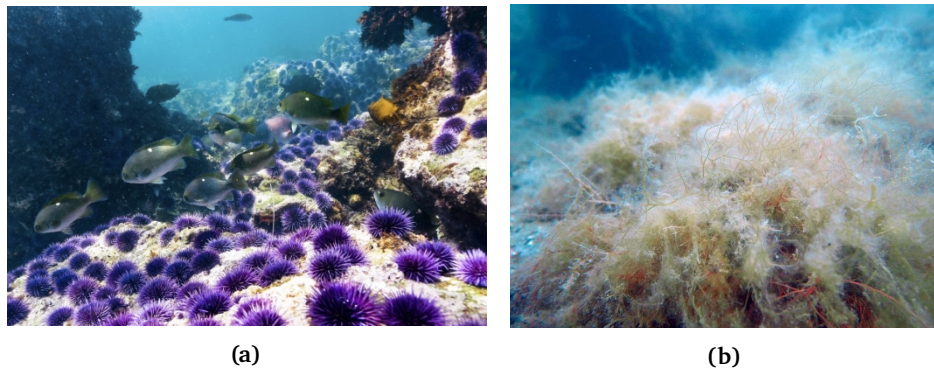




# Introduction

Coastal habitats are experiencing significant degradation [1], both globally and locally [2, 3]. Marine vegetation, known as blue forests, forms the basis of marine coastal ecosystems and have a great importance for climate and society [4–8]. As well as the green forests on land, blue forests also absorb a significant amount of the climate gas Carbon Dioxide (CO<sub>2</sub>) as they account for up to 70 % of the marine carbon storage. However, the absorbed CO<sub>2</sub> is released if the vegetation is demolished [7]. Blue forests absorb nutrients, which in turns reduces turbidity and bind sediments thus prevents coastal erosion. Blue forests offer recreational environments for thousands of species, including commercial seafood which many local communities depend on [6].

Blue forests along the Norwegian coasts have lately been receiving increased interest due to their importance [6]. However, blue forests face several existential threats in the Norwegian coast, such as destructive turf algae [9]. The so-called turf algae attaches itself to other marine vegetation, slowly suffocating it because the turf algae blocks the sunlight from reaching the underlying vegetation [10]. Large areas in the Southern part of the Norwegian coast are now covered by the damaging turf algae, and it is spreading further north [11]. Urchin overgrazing also threatens the existence of blue forests, because it causes large areas to appear as barren grounds as the urchins take over the seafloor. These barren grounds have limited productivity and potentially acts as hostile environments for the species that depend on the blue forests [2]. Figure 1.1 shows examples of how the local marine environments are effected by urchins overgrazing and turf algae. The blue forests are also sensitive to



**Figure 1.1:** a) Urchins have taken over the local marine environment (Credit: Jacey Can Wert [16]), (b) turf algae lays over the marine vegetation as a blanket, preventing the access of sunlight for the underlying vegetation (Credit: Hartvig Christie [17]).

environmental changes, as well as anthropogenic activity [2, 4, 6, 12].

The Norwegian Blue Forests Network suggests that the most important actions for preserving the valuable blue forests along the Norwegian coast is to "1) protect the blue forests we have; 2) restore the blue forests we have lost; or 3) replace the blue forests we have lost" [7]. In order to fulfill these three points, thorough mapping of the extent and the physiological state of the blue forests is needed. This has previously been done by different in-situ monitoring techniques [13]. In-situ mapping and monitoring of blue forests along the Norwegian coast is both resource-intensive and ineffective due to the large extent of the coast [5, 13]. Lately, remote sensing has been studied for monitoring of marine vegetation, as a more powerful and effective alternative to in-situ field campaigns. [3–5, 14, 15].

Hyperspectral and multispectral sensors onboard drones, air crafts and satellites collect valuable spectral information about the blue forests [5, 12, 14, 18–20]. Satellite imagery have the potential of providing continuous surveillance of the coastal regions in Norway. However, the high cloud coverage in the Arctic [21], the limited spatial and spectral resolution in freely available data is a limitation for the monitoring the blue forests. Using air crafts is an alternative to satellite imagery, but this is often a costly way of retrieving data [3, 22]. Drones offer a cost-effective on-demand and close-range method of collecting data of marine vegetation [23].

Marine vegetation mapping from hyperspectral drone data has been done using various Vegetation Indices (VI) [8]. Although VIs are widely applied for both marine vegetation and land vegetation due to their simple rule-based structure and good results [8, 24], they do have some limitations. As with other rule-based

methods, VIs are general methods that do not adapt to specific datasets [24,25]. In addition, VIs do not exploit available hyperspectral information, they only use a few predefined spectral bands [24]. This limits the VIs in the ability to classify different species as well as make them inaccurate in for some applications [24]. Machine Learning (ML) methods are capable of handling complex data by adapting the model to learn the patterns in the data and do not depend on predefined assumptions [25]. Lately, different ML methods have been studied for classifying marine vegetation, as an effective and powerful alternative additionally to in-situ campaigns [3–5]. These methods include for instance K-means clustering [5], Random Forest (RF) [3] and several other statistical and machine learning methods [5,23]. In the past years, neural networks have also been applied for classifying different hyperspectral satellite data as well as marine vegetation data with outstanding performance [26–29].

Hyperspectral imaging by using drones has, to the author's best knowledge, not yet been applied for mapping marine vegetation along the coast of Norway. The project Mapping of Algae and Seagrass by using Spectral Imaging and Machine Learning (MASSIMAL) is currently collecting hyperspectral data of the Norwegian coast by the use of drones. In order to use this unique data for marine vegetation mapping, algorithms need to be studied and established.

The objective of this thesis is to study various ML methods, including the Random Forest (RF), Support Vector Machines (SVM) and Convolutional Neural Network (CNN) for classifying marine vegetation from hyperspectral drone data acquired by the MASSIMAL project. CNN is suggested as a novel approach for classifying hyperspectral data, due to its ability to learn complex pattern with high accuracy and handle large amount of data [26,28]. This thesis has focused on a unique dataset acquired in Larvik, located in the south of Norway, close to Oslo. The three methods have been evaluated and compared by a thorough statistical analysis. The impact of turf algae in the classifications has been studied. The thesis also conducted a study of the robustness of the three methods by using an independent dataset. (Note, a feasibility of using the aforementioned methods in application has also been tested, although this was not the focus of this work. This was done by using the here established methodologies for predicting marine vegetation on data, where class labels are not available). The results of this work indicate that ML methods have the potential to be applied to hyperspectral drone data for monitoring blue forests along the coast of Norway.

## 1.1 Structure of the thesis

This thesis is structured in 6 chapters. Chapter 6 is the main part of the thesis.

**Chapter 2** outlines the fundamentals of optical remote sensing applied for marine vegetation, building a theoretical background needed for evaluation of the main objective of the thesis.

**Chapter 3** gives an introduction to hyperspectral imaging, its history, the process flow and its applications and challenges.

**Chapter 4** presents different types of marine vegetation, the importance of monitoring and mapping, and how it can be done by using hyperspectral remote sensing.

**Chapter 5** presents different algorithms that are, and can be used for underwater vegetation mapping.

**Chapter 6** is the main part of the thesis. The chapter first gives an introduction to machine learning and describes the methods used in the thesis. The dataset that was made available for this study is described, continued by the methodology. The results are presented together with a discussion and a conclusion of the thesis. Future work is also suggested.



# /2

## Optical remote sensing of marine vegetation

In this project, optical remote sensing is used for monitoring of marine vegetation. The sunlight is the energy source in optical remote sensing, where the sensors detect the reflected sunlight of a target. This chapter therefore gives an introduction to the physics of optical remote sensing, starting with describing ElectroMagnetic (EM) radiation and blackbody radiation in Sec. 2.1 which explains how the sunlight behaves. Sec. 2.2 describes how the sunlight propagates and how it is attenuated when it travels from the sun to the imaging sensor. Then, Sec. 2.3 introduces the remote sensing reflectance and the water leaving radiance. The last section, Sec. 2.4 describes what sun glint is, how it affects the measurements and what can be done in order to try to remove the influence of it.

### 2.1 Electromagnetic radiation and blackbody radiation

Optical remote sensing uses sunlight as the source of illumination. This section therefore provides background theory about ElectroMagnetic (EM) radiation and blackbody radiation, which describes how the sunlight behaves.

There are two models that can be used to describe EM radiation; the wave model and the particle model. The wave model describes the radiation as a travelling EM wave with a magnetic and electric field orthogonal to the direction of propagation. The magnetic and electric fields relation and behaviour are described by Maxwell's equations for vacuum, which can be written as

$$\nabla \cdot \mathbf{E} = 0, \quad (2.1) \quad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (2.2) \quad \nabla \times \mathbf{B} = \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}. \quad (2.4)$$

$\Delta$  is the vector differential operator,  $\mathbf{E}$  given in [V/m] is the electric field component,  $\mathbf{B}$  given in [T] is the magnetic flux density,  $\mu_0$  given in [H/m] and  $\epsilon_0$  given in [F/m] are the vacuum permeability and vacuum permittivity, respectively. By combining the equations and solving them leaves the wave equation for vacuum, given as

$$\mathbf{E} = \mathbf{E}_0 \exp(i(\mathbf{k} \cdot \mathbf{r} - \omega t)), \quad (2.5)$$

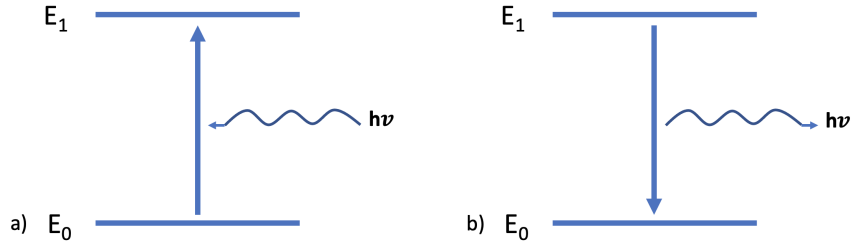
where  $\mathbf{E}_0$  is the complex wave amplitude,  $\mathbf{r}$  a position vector,  $\mathbf{k}$  is the complex wave number,  $\omega$  describes the angular frequency and  $t$  is the time [30].

The particle model describes EM radiation as a particle where its energy is given as

$$E = h\nu = \frac{hc}{\lambda} [J], \quad (2.6)$$

$h$  is the Planck's constant ( $h = 6.62607015 \cdot 10^{-34} \text{ JHz}^{-1}$ ),  $\nu$  is the frequency,  $\lambda$  is the wavelength and  $c$  is the speed of light ( $c = 3.0 \cdot 10^8 \text{ m/s}$ ). When an EM particle collides with another particle (atom, molecule, ion) it is either scattered or absorbed. To be absorbed, the energy of the EM particle needs to be  $\Delta E = E_n - E_0$ .  $E_n$  refers to energy levels which corresponds to the amount of energy the particle can take on. Energy level  $E_0$  is the ground state where the particle is at its lowest possible energy state. Energy level 1,  $E_1$  is the first excited state in the atom that electrons be lifted up to. Electrons can be lifted into excited states if the incoming energy matches the energy gap between the different energy states,  $\Delta E$ . For example, to be lifted up to the first excited state, energy level 1, the incoming energy needs to match the energy gap  $\Delta E = E_1 - E_0$ .

Figure 2.1 (a) shows an illustration of absorption, where an incoming photon matches the energy gap,  $h\nu = \Delta E = E_1 - E_0$ , which causes absorption and electrons to be lifted up to energy state  $E_1$ . If the energy of the incoming photon is higher than the binding energy of an atom then the photon can be absorbed and ionize the atom. If electrons are relaxed down to a lower energy state,



**Figure 2.1:** a) Absorption of incoming EM radiation that has an energy of  $\Delta E = E_1 - E_0 = h\nu$ . Electrons are lifted from energy level 0 (ground state) to energy level 1 (first emission state) and (b) emission of energy that corresponds to the same amount of energy between level 1 and 0;  $\Delta E = E_1 - E_0 = h\nu$ .

then emission occurs, as shown in Fig. 2.1. Doing so, the particle releases the same amount of energy that corresponds to the energy gap between the two levels, in this case  $\Delta E = E_1 - E_0$ . This phenomena is called emission. If the energy of the incoming photon is lower than the required energy for absorption, the particle is scattered away.

Different forms of marine vegetation consists of different substances, which each emit and absorb EM radiation at different wavelengths. Using the reflected and emitted EM radiation, it is possible to distinguish between different marine vegetation species [30, 31].

Figure 2.1(b) shows the opposite case, where electrons are relaxing down to a lower energy state. Doing so, the particle releases the same amount of energy that corresponds to the energy gap between the two levels, in this case  $\Delta E = E_1 - E_0$ . This phenomena is called emission.

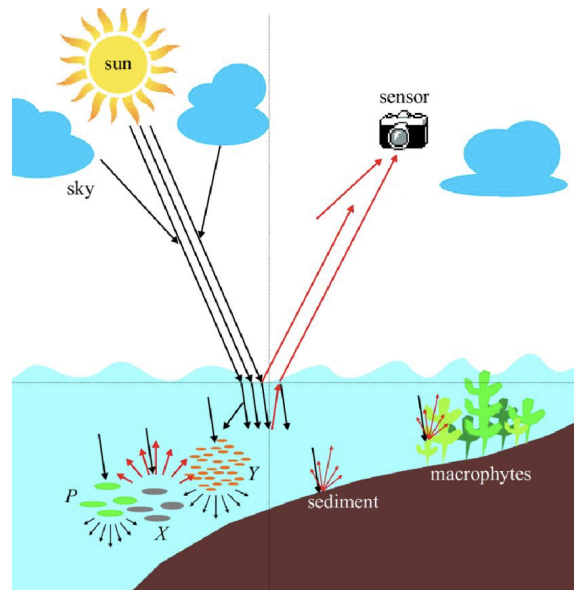
The Sun's radiation can be described by Planck's law which gives the spectral radiation for a blackbody. A blackbody is defined as a perfect absorber and emitter of energy. Planck's law is given by

$$B(\lambda, T) = \frac{2hc^5}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \quad [\text{Wsr}^{-1}\text{m}^{-2}\text{Hz}^{-1}], \quad (2.7)$$

where  $\lambda$  is the wavelength of the radiation,  $T$  is the temperature in Kelvin and  $k_B$  is Boltzmann's constant at  $1.380649 \cdot 10^{23} \text{ JK}^{-1}$ . Stefan Boltzmann's law is derived by integrating Planck's law with respect to wavelength and is given as

$$B(T) = \int_0^{\text{inf}} B_\lambda(T) d\lambda = \sigma T^4 \quad [\text{Wm}^{-2}], \quad (2.8)$$

where  $\sigma$  is a constant of  $5.67 \cdot \text{Wm}^{-2}\text{K}^{-4}$ .  $B(T)$  gives the total power radiated from a blackbody, and shows that the radiation is proportional with the tem-



**Figure 2.2:** Illustration of sunlight that is directly reflected at the ocean surface and diffuse scattering in the sky. Diffuse scattering of sunlight also occurs in the water column (scattering of for instance CDOM and suspended matter indicated with  $P$ ,  $X$  and  $Y$ ) which together with scattering from the sea bottom makes up the water leaving radiance. Figure reused from [32].

perature to the power of 4. It is also possible to derive Wien's displacement law from Planck's law by differentiating Planck's law with respect to wavelength, setting it equal to zero and solving it for zero. Wien's displacements law describes the maximum of the spectrum for a given temperature, and it can be expressed by

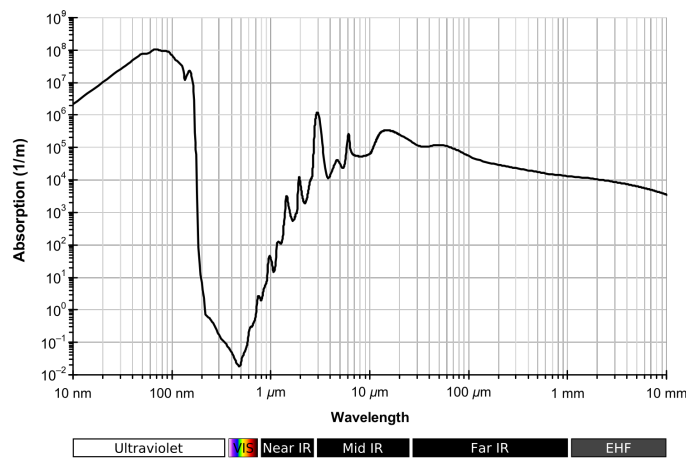
$$\lambda_m = \frac{a}{T} \quad [m], \quad (2.9)$$

where  $a$  is Wien's displacement constant at  $2898 \mu\text{m}$  [30].

This section provided a short overview of basic theory about EM radiation and blackbody radiation which is used to describe sunlight, the energy source in optical remote sensing. The next section will describe the path of the sunlight, from the sun to the imaging sensor.

## 2.2 Light's pathway to the sensor

The measured signal at the optical sensor originates for several different sources. Figure 2.2 shows an overview of how sunlight propagates before it reaches the imaging sensor. The sunlight first travels through the atmosphere before

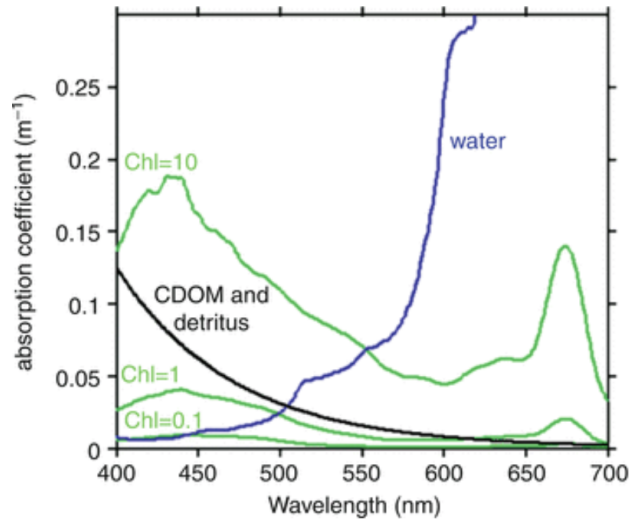


**Figure 2.3:** Absorption of EM radiation by the water. Figure reused from [34].

it reaches the ocean. On the journey through the atmosphere, some of the radiation is scattered and absorbed by the atmosphere. In the case of drone imaging close to the ocean surface, scattering from the atmosphere to the sensor is less relevant than for instance for satellites. Further, the radiation that reaches the ocean surface is either scattered or the surface, absorbed or transmitted through the surface. If the surface is smooth, it is assumed that the reflection of the sunlight is specular. However, for ocean surfaces, the surfaces often appear rough due to waves. The surface interactions then become more complicated, and the interaction is then governed by the Rayleigh criterion for rough surfaces, Snell's law and the Fresnel equations [31]. Reflection of rough surfaces in the direction of the imaging sensor generates glint (see Sec. 2.4) [33].

A portion of the sunlight will be absorbed by the ocean. Figure 2.3 shows the absorption spectrum of clear water. The figure shows that in the VISible (VIS) region of the spectrum (400 nm - 700 nm), the absorption of EM radiation is low for water. However, in the Near InfraRed (NIR) region and for larger wavelengths, as well as wavelengths shorter than 400 nm, the absorption is strong. This leaves the visible region of the EM spectrum as the part of the spectrum that has the possibility to propagate down through the ocean column, scatter off the sediments and macrophytes, propagate up through the ocean column and reach the imaging sensor [31].

Radiation that propagates through the water column decays as a function of the depth of the water. This occurs due to scattering and absorption in the water column. The presence of different substances at the water surface and in the water column affects the scattering and absorption of the radiation.

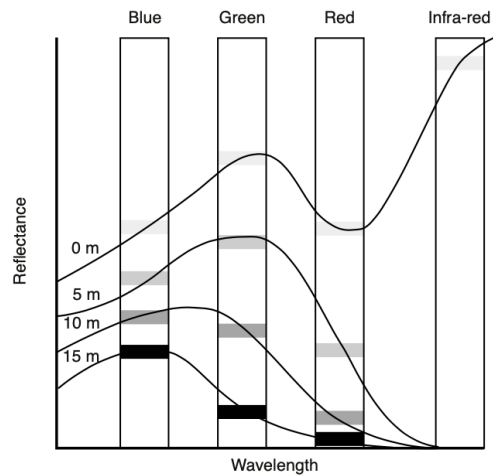


**Figure 2.4:** Absorption spectrum of CDOM (black line), water (blue line) and Chlorophyll-a (green lines) for different concentration. CDOM has an exponentially decrease in absorption with wavelength, while water has an almost exponentially increase in absorption with wavelength. The absorption of Chlorophyll-a depends heavily on concentration, and it has mainly two absorption peaks, in the blue band and in the IR band. Figure reused from [36].

Figure 2.4 shows how different concentrations of Chlorophyll-a absorbs, as well as the absorption from Colored Dissolved Organic Matter (CDOM) and detritus, and the absorption of water itself. The presence of these substances can possibly prevent the observation of sea bottom vegetation and complicate the observation of macrophytes close to or at the ocean surface [35, 36].

Additional to the absorption, and scattering, due the presence of Chlorophyll-a, CDOM etc. in the water column, the radiation experiences attenuation as it propagates through the water column due to absorption by the water itself. The attenuation of the radiation depends on the wavelength of the radiation, where the red part of the EM spectrum attenuates more rapidly than the blue part. Figure 2.5 shows a hypothetical reflectance spectra of a marine vegetation species, and how it might change with depth due to the attenuation effect. The changes seen in the reflectance spectra at different depths, could make it challenging to recognise the same species at 1 meters depth and 10 meters depth. It is possible to correct for this affect if the bathymetry is known for every pixel and the concentration of CDOM etc., i.e. the attenuation is known, but it is not done in this project [35].

This section provided a description of the sunlight's pathway from the sun to the imaging sensor. The next section will introduce the remote sensing



**Figure 2.5:** Hypothetical spectra for seagrass or macroalgae and how it might change with increasing depth. The attenuation differs for each wavelength, making it harder to discriminate between different species. Figure reused from [35].

reflectance and water leaving radiance.

## 2.3 Remote Sensing Reflectance and Water Leaving Radiance

The remote sensing reflectance,  $R_{rs}$ , is the reflection just above the ocean surface. It contains the spectral color information of the water column. The remote sensing reflectance is a function of the water leaving radiance,  $L_w$ , which is the upwelling radiance from ocean to air [31].

In the case of ocean imaging performed by drones where the atmospheric influence of the radiation is disregarded, the water leaving radiance is what the drone measures. The atmospheric influence can be disregarded if the drone have a flight height below a certain height. Research show that for a flight height above 50 meters, significant atmospheric effects appeared [37]. Using these findings, this project further assumes that for images acquired below 50 meters above the sea, atmospheric influence can be neglected, which is the case in this project.

The water leaving radiance is proportional to the backscatter coefficient,  $b(\lambda)$ ,

divided by the absorption coefficient,  $a(\lambda)$  [31]:

$$L_w(\lambda) \propto \frac{b(\lambda)}{a(\lambda)} \quad [\text{Wm}^{-2}\text{sr}^{-1}]. \quad (2.10)$$

The backscatter and absorption coefficient is a sum of the different substances in the water that scatter and absorb, such as the water itself, phytoplankton (Chlorophyll-a), CDOM (Colored Dissolved Organic Matter) and suspended particulates [31]. The absorption and backscatter coefficients are given in Eq. (2.14) and (2.15). The remote sensing reflectance is given by

$$R_{rs} = \frac{L_w(\lambda)}{E_d(\lambda, 0+)} \quad [\text{sr}^{-1}], \quad (2.11)$$

where  $E_d(\lambda, 0+)$  is the down-welling solar irradiance where 0+ refers to just above the ocean surface [31]. The remote sensing reflectance and water leaving radiance can be used in bottom reflectance retrieval. In the case of bottom reflectance retrieval, the remote sensing reflectance can be written by

$$R_{rs} = R_{rs}^w(\lambda) + R_{rs}^b(\lambda) \quad (2.12)$$

where  $R_{rs}^w(\lambda)$  describes the water column interactions and  $R_{rs}^b(\lambda)$  is the bottom reflectance.  $R_{rs}^w(\lambda)$  and  $R_{rs}^b(\lambda)$  can be written by [38]

$$\begin{aligned} R_{rs}^w(\lambda) &= 0.05 \frac{b_b(\lambda)}{a(\lambda) + b_b(\lambda)} \left( 1 - \exp^{-3.2(a(\lambda) + b_b(\lambda))H} \right) \\ R_{rs}^b(\lambda) &= 0.173(\lambda) \exp^{-2.7(a(\lambda) + b_b(\lambda))H}. \end{aligned} \quad (2.13)$$

$a(\lambda)$  is the absorption coefficient,  $b_b(\lambda)$  is the backscatter coefficient, H is the depth of the water in meters and  $\rho(\lambda)$  is the bottom albedo. The bottom albedo is a measure of how much the sea bottom reflects [38]. The absorption coefficient is given by

$$a_T(\lambda) = a_w(\lambda) + a_{ph}(\lambda) + a_{CDOM}(\lambda) + a_{sp}(\lambda), \quad [m^{-1}], \quad (2.14)$$

where  $a_w(\lambda)$ ,  $a_{ph}(\lambda)$ ,  $a_{CDOM}(\lambda)$ ,  $a_{sp}(\lambda)$  are clear water, phytoplankton, CDOM and suspended particle absorption coefficients, respectively [31]. The backscatter coefficient can be written by

$$b_{bT}(\lambda) = b_{bw}(\lambda) + b_{bph}(\lambda) + b_{bsp}(\lambda), \quad [m^{-1}], \quad (2.15)$$

where  $b_{bw}(\lambda)$ ,  $b_{bph}(\lambda)$ ,  $b_{bsp}(\lambda)$  is clear water backscatter, phytoplankton backscattering and suspended particle backscattering, respectively. Backscatter from CDOM is negligible so it is not included in the equation [31].





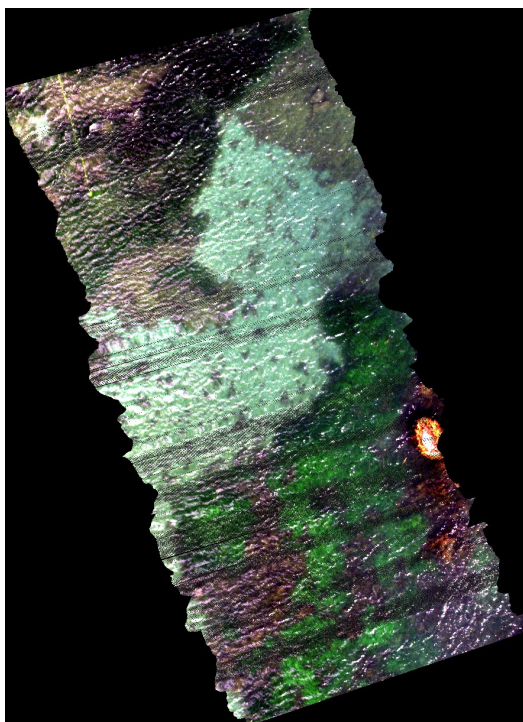
**Figure 2.6:** Turbid water (left) vs. clear water (right). The turbidity makes it challenging to identify the sea bottom vegetation. Image reused from [39].

The equations stated here show how the turbidity (the presence of different substances in the water column) influence the remote sensing reflectance. Figure 2.6 shows turbid water (left) and clear water (right). The figure shows how turbidity makes it challenging to identify the underwater vegetation. If the water is clear, Eq. (2.14)- (2.15) is only left with the water absorption and scattering terms, hence the retrieval of the bottom reflectance is more straightforward. The presence of different substances in the water column and at the ocean surface do, as shown in this section, affect the measured signal. With the presence of different substances in the water column, the remote sensing reflectance consists of a mix of different signals from the different substances, in addition to the signal from the marine vegetation. This affects the ability to identify marine vegetation species.

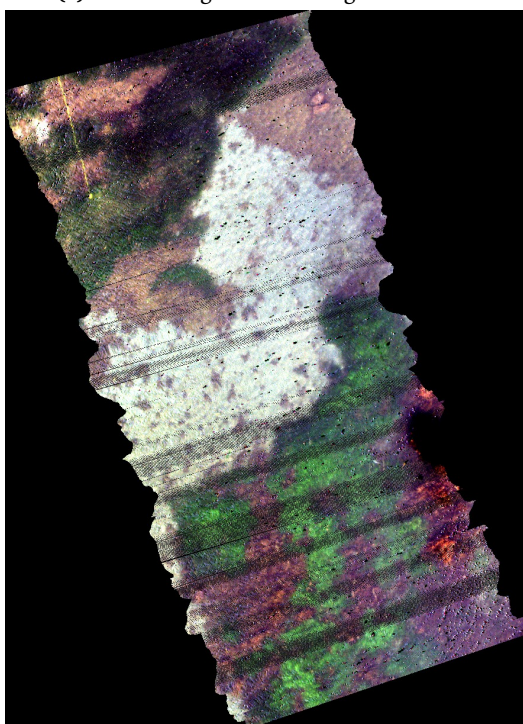
This section introduced the remote sensing reflectance and water leaving radiance. The next section will highlight the problems that occur due to sun glint and how it can be accounted for.

## 2.4 Sun glint

Sun glint is a phenomenon that occurs when sunlight is reflected off a wind-roughened ocean surface into the sensor direction. Sun glint can overwhelm the reflected ocean radiance, obscuring the signal from the sea floor [31]. Because of this, sun glint needs to be avoided, masked out or removed. Wind speed, sensor position and viewing angle all contribute to the amount of sun glint that appears in the image. Sun glint removal can be performed by modelling the capillary waves using wind models, and thus try to estimate how the sun light interacts with the waves. In addition a mask can be applied to remove the pixels that exceeds a specified threshold [31].



(a) Drone image before sun glint removal



(b) Drone image after sun glint removal

**Figure 2.7:** Illustration of how sun glint affects the quality of a drone image. (a) shows the image before sun glint removal. (b) shows the same image after sun glint removal; in this image it is easier to visually identify the different species in the image, especially in the bottom right part of the image. Note that some parts of the image in (b) are masked out, due to saturation of the spectrum at data acquisition or if the sun glint removal have procured an invalid spectrum with negative values. The figure shows one of the images from the dataset used in this thesis. From personal correspondence with Martin Skjelvareid.

Hedley et al. (2005) [33], describes a method for sun glint removal that includes using NIR measurements to estimate the amount of sun glint for each pixel. Assuming all NIR radiation is absorbed by the ocean, the NIR radiation that is reflected to the sensor, will thus be due to sun glint. One also needs to assume a linear relationship between the amount of sun glint in the VIS and NIR brightness. This way, the amount of NIR brightness can be an indicator of how much sun glint there is in the VIS part [33]. Figure 2.7 shows an example from the dataset used in this project of how sun glint disturbs the signal from the sea bottom. The method from Hedley et al. (2005) is applied to the image. Figure 2.7a shows the image before sun glint correction, and Fig. 2.7b shows the result of sun glint removal. Visual inspection of Fig. 2.7(a) and(b) suggests that the removal of sun glint makes it easier to both separate different species and identify which species are present in the image, especially in the dark areas. Note that some areas are masked out in Fig. 2.7 (b), due to mainly two reasons; the spectrum have been saturated during data acquisition or the sun glint removal have resulted in an invalid spectrum with negative values.

This chapter described how the sunlight interacts with the ocean surface, water column and sea bottom. These relations are important to understand when working with remote sensing in shallow waters. The next chapter will provide basic background information on hyperspectral imaging, which is the type of data that is used in this project.



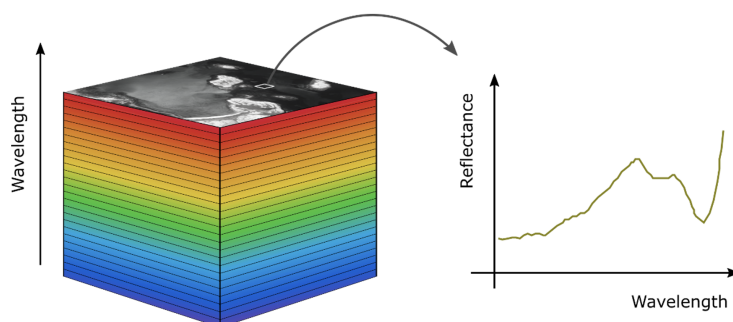
# /3

## Hyperspectral imaging

This project uses hyperspectral remotely sensed imagery in order to classify and detect different marine vegetation. Hyperspectral sensors provide high spectral resolution (a high number of spectral bands with narrow bandwidth) making them a good tool in mapping of vegetation in marine habitats [8]. This chapter therefore provides an introduction to hyperspectral imaging.

Hyperspectral sensors used for remote sensing of the Earth were first developed by the Jet Propulsion Laboratory in the 1980s. These sensors were able to capture detailed and more precise spectral information than the multispectral sensors available [24]. Hyperspectral imaging sensors use imaging spectroscopy to image the Earth with high spectral resolution. Imaging spectroscopy is an imaging technique where, instead of only capturing the classical RGB image, the sensor captures as many as several hundreds spectral bands at the same time for each pixel, with only a few nm in bandwidth. The hyperspectral sensors used for ocean remote sensing typically capture bands from the VIS to the NIR part of the EM spectrum [24].

The hyperspectral image data can be presented by 3D data cube as shown in Fig. 3.1. Figure 3.1 shows an illustration of the three dimensional data cube, where x and y forms the image and z-dimension contains the spectral bands. A pixel contains a stack of wavelengths that make up an almost continuous spectra of the surface. For vegetation studies, this detailed spectral information makes it possible to detect small changes in the physical structure and biochemistry of the vegetation [24].



**Figure 3.1:** 3D Hyperspectral imaging cube shows a stack of bands in z-direction and an image displayed in the x and y direction. One pixel in the image contains a lot of narrow spectral bands, which forms a spectra for the surface the pixel represents. Credit: MASSIMAL project

Hyperspectral remote sensing is a relatively new technology, and its applications for detection and identification of terrestrial and marine vegetation, minerals and man-made materials etc. is currently being investigated by scientists. Due to the detailed spectral information for each pixel in the image, hyperspectral sensors shows promising results for marine vegetation mapping [4, 5, 23]. Compared to multispectral remote sensing, hyperspectral remote sensing has much more detailed spectral measurements. Analyzing multispectral data, it is possible to detect vegetation and possibly distinguish between different species [30]. By utilizing the high spectral resolution in hyperspectral sensors, it is possible to detect small changes in the spectral response of the target [24]. The detailed spectral information is useful for mapping of marine vegetation because it makes it possible to identify the extent and health of the vegetation and possibly distinguish between different species with similar spectral signature [24, 40]. Figure 4.3 shows the spectral signature from three different seagrass species. Their spectra are almost identical, but some parts differ by a small amount. These small differences could be detected by a hyperspectral sensor due to the high spectral resolution [24].

Optical remote sensing of the Earth have some challenges that occurs due to the nature of the surface of the Earth. Shadowing, topographic variations and surface composition are challenges that can confuse interpretation of the image. Shadowing occurs when large objects causes dark areas (shadows) where the imaging sensor is not able to capture any information [30]. Topographic variations refers to height variations in the terrain which can change the spectral reflectance in such a way that some areas in the image appear distorted. Topographic correction is often used as a way of trying to correct for the reduced influence of topography [41]. The composition of the surface often makes it challenging to retrieve the correct information of what the surface is composed of. This is often due to the occurrence of two or more surface types captured by

one pixel in the image. If the spatial resolution is 2 by 2 meters, then inside that square different surface may be present such as clear water, phytoplankton, rocks, grass, etc. Especially in the coastal areas, where land meets the ocean this is a challenge. When a pixel covers both land and the ocean surface, the reflection from land is often a lot stronger than reflection from the ocean surface, which can cause saturation of the signal. A high spatial resolution will reduce this problem because there will be a higher amount of pure water pixels close to the shore. One way to achieve high spatial resolution is to use drones, which operates at a low altitude. For such pixels, that represents a composition of different surfaces, the observed spectra is mixed. There are generally two types of mixing; linear mixing and non-linear mixing. Linear mixing refers to the situation where the combinations of different surfaces are additive, i.e. the surface radiations are separate until they reach the sensor. For this case, it is possible to estimate the ratio of each surface material by linear unmixing. Linear unmixing uses known spectral properties and determines the relative contribution of different species for each pixel. [24]. Further challenges includes radiometric interaction with the ocean surface as described in Sec. 2.3.

Although the detailed spectral resolution of hyperspectral sensors is very useful, some of the information in the measured spectra may be redundant. It can often be challenging to know which spectral bands that are important for the problem at hand [26]. Dimensionality reduction is often used in order to extract the important features (either the spectral bands them self, or a transformation or combination of the spectral bands) (see Sec. 6.1.1 and Sec. 6.1.2).

While this work focuses on hyperspectral sensors mounted on drones, there are hyperspectral satellite sensors that are operational. An example is the Italian PRecursorre IperSpettrale della Missione Applicativa (PRISMA), whose mission is to monitor natural resources and collect data of the atmosphere in Europe. PRISMA has 250 bands with a spatial resolution of 30 meters, and a swath width of 30 km. It was launched in 2019 and is currently operational (2022) [42]. New sensors are developing, and a sensor from the Norwegian University of Science and Technology (NTNU) launched a hyperspectral sensor in January 2022, onboard HYPer-spectral Satellite for Ocean observation-1 (HYPSO-1 ). HYPSO-1 is going to be used for research and map algal blooming and changes in ocean color. It measures spectral bands from 400 nm to 800 nm with spectral bands of less than 5 nm with a swath width of 70 km, and a spatial resolution of less than 100 meters [43]. An upcoming sensor from National Aeronautics and Space Administration (NASA), is the Plankton, Aerosol, Cloud ocean Ecosystem (PACE), which is going to investigate the atmosphere and ocean exchange of carbon dioxide and more. It is planned to launch in the end of 2023. The ocean color instrument of PACE will measure from 340 nm to 890 nm with 5 nm resolution, have a spatial resolution on 1 km and swath width of 2663 km [44].

By looking at the numbers for these sensors, it is clearly seen that the different resolutions often comes at a cost of the others. For example the big swath width of PACE causes a poor spatial resolution of 1 km, compared to the good spatial resolution of PRISMA where the swath width is only 30 km wide. The smaller the swath width, the longer the revisit time is. The revisit time is also important to consider, especially for vegetation when changes in the physical structure can happen in short periods of time (days) [30, 31]. For drones, the spatial resolution can be improved by flying closer to the object, which leads to a smaller area being imaged [45]. The revisit time for drones are as often as the drone can be safely operated.

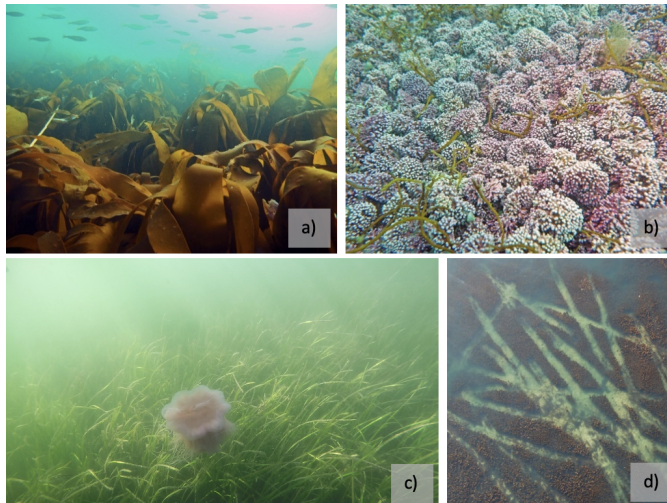
This chapter provided an introduction to hyperspectral remote sensing, which is the type of data this work is based on. The next chapter will look into the biology of marine vegetation and the importance of sea bottom vegetation as well as how hyperspectral data can be used in the mapping and classification of marine vegetation.



# /4

## Blue forests

This work studies underwater vegetation and explores if it is possible to monitor their extent by drone remote sensing. This chapter therefore provides an introduction to underwater vegetation and how it can be monitored by drone remote sensing.

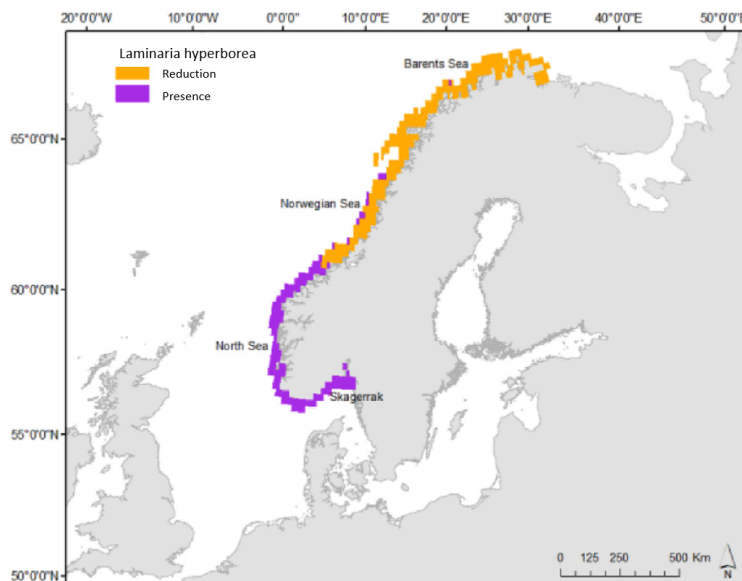


**Figure 4.1:** Illustration of different sea bottom vegetation species. (a) kelp forest, photo by Janne Gitmark, (b) Maerl beds, photo by Eli Rinde, (c) Seagrass meadow, photo by Kasper Hancke and (d) kelp trawl tracks, photo by Arne Follestad. Seagrass and kelp forests are some of the most common and widespread marine vegetation along the Norwegian coast. Credit: MASSIMAL project.

Blue forests refers to underwater vegetation that typically grows in coastal areas, like seagrass, rockweed and kelp. Blue forests provide a key habitat for several marine species while also offering protection from sea level rise and erosion, storing carbon and act as a breeding habitat for economically important species [4, 6].

Figure 4.1 (a)-(d) shows kelp forest, maerl beds, seagrass meadow and kelp trawl tracks, respectively. Out of these species, seagrass and kelp forests are some of the most widespread and important underwater species along the Norwegian coast [6]. Seagrass grows on the soft seabeds and in sheltered bays. All seagrass located in Norway belongs to the eelgrass family. Kelp forests grow in rocky habitats in most polar temperal environments where sunlight is present and available for photosynthesis. Figure 4.2 shows a map of the change in presence of *Laminaria hyperborea*, which is a type of kelp forest, in the past 50 years. Orange indicate a decline and purple indicate no change in coverage. Figure 4.2 clearly shows that *Laminaria hyperborea* have experienced a reduction north of the Polar circle. The reduction of *Laminaria hyperborea* in the northern areas, and other underwater vegetation species, is mainly caused by urchin overgrazing [2]. This creates a less productive environmental system than with the presence of underwater vegetation. Europe has lost 1/3 of their seagrass extent due to factors such as poor water quality and poor coastal management [46].

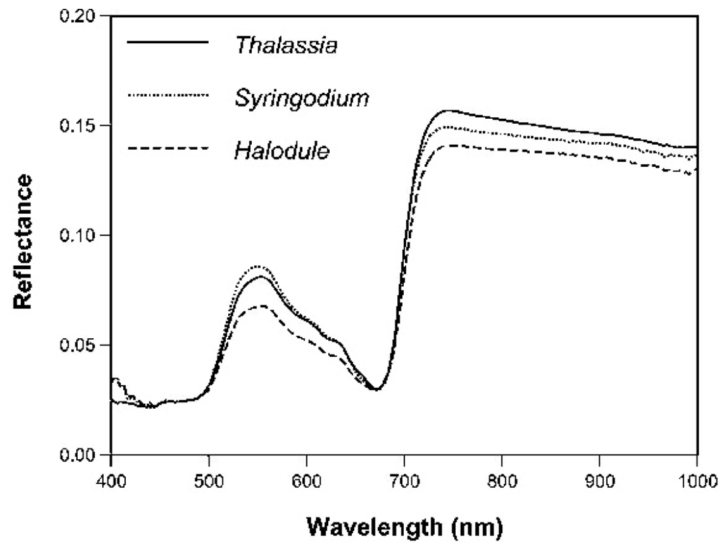
Maerl beds are purple or pink hard seaweeds that lives in open coast areas. The extent of Maerl beds in Norway is not thoroughly mapped, but there has been a decline in the North East Atlantic due to the use of for example agricultural soil conditioner and for use in human and animal food activities. Since Maerl beds have a slow growing rate, they are considered as non-renewable resource [47]. The observed decline in these sea bottom vegetation species underlines the importance of being able to monitor blue forests efficiently and accurately, especially in the Arctic environments.



**Figure 4.2:** A map of *Laminaria hyperborea*, a type of kelp forest, along the coast of Norway. The orange marking indicates reduction of *Laminaria hyperborea* since the 1970 and the purple marking indicated no significant change in extent since the 1970 [2]. The map illustrate how *Laminaria hyperborea* have had a clear reduction in the northernmost areas along the Norwegian coast. Figure reused from [2]

Underwater vegetation exists in the coastal zones and rivers all around the world. It can be rooted to the sea floor or floating in the water column or at the water surface. For blue forests to thrive, a few requirements must be met; access to sunlight (which means they cannot live in deep waters where sunlight is unable to penetrate through the water column), carbon dioxide, oxygen and nutrients. Throughout photosynthesis they transform carbon dioxide, water and sunlight into oxygen which is used by marine animals to thrive. The main nutrients needed by underwater vegetation is nitrogen and phosphorus which are often found in sea bottom sediments [48].

During photosynthesis, Chlorophyll plays the major role in the process. Chlorophyll is the primary photosynthetic pigment which absorbs energy in the VIS part of the EM spectrum, while it has a strong reflectance in the NIR region [31]. Figure 4.3 shows a spectral plot of three different types of seagrass: *Thalassia*, *Syringodium* and *Halodule*. By inspecting Fig. 4.3, a strong reflection around 700 nm is observed. This shift in reflection from low in the VIS part to strong reflection in the NIR part, is called the red edge. The red edge is a characteristic spectral signature that is common for all healthy plants that undergo photosynthesis. It is widely used in identifying healthy vegetation, and is the basis of



**Figure 4.3:** Reflectance plot of three different types of seagrass, *Thalassia*, *Syringodium* and *Halodule*. The red edge is clearly visible at ca. 700 nm for all three seagrasses. This characteristic spectrum makes it possible to detect vegetation by remote sensing. Figure reused from [40].

VI. VIs are spectral transformations that uses the spectral shift around the red edge to identify living vegetation. The most commonly used is the Normalized Vegetation Index (NDVI), which describes the difference in reflection in the VIS and NIR [24] (see Ch. 5).

While it is a relatively easy task to identify living vegetation using for example the VIs, differentiating between different species can be more challenging. As Fig. 4.3 shows, the three spectra are very similar. It is, however, possible to distinguish between different species if the spectral resolution is high enough to identify small shifts in the spectral response. The high spectral resolution of hyperspectral imaging can be used to differentiate between species. The detailed spectral information can also be used to detect small shifts in the spectrum which holds information about the physiological structure and biochemistry of the vegetation. This information is important to capture when mapping the extent and health of underwater vegetation [24, 31].

This chapter provided a background on what blue forests are, why it is important to be able to monitor them, and how it can be done using their spectral signature. The next chapter will introduce algorithms that can be used for surveillance of blue forests.

# /5

## Algorithms

Mapping and surveillance of underwater vegetation is often done by large datasets where the use of computer computation power is needed in order to efficiently analyse the data. This chapter looks into different approaches that are currently used, and that can be used for vegetation mapping and surveillance.

### 5.1 Vegetation Indices

Vegetation Indices are a popular and frequently used method for vegetation surveillance. As mentioned in Ch. 4, VIs use spectral transformation based on the spectral signature of Chlorophyll, the photosynthetic pigment that living plants contains. There exists several different VIs, but the most popular is the Normalized Difference Vegetation Index (NDVI). NDVI is a simple method that looks at the difference between the NIR and RED band while normalizing it. The NIR and RED bands varies slightly from sensor to sensor. For example, for Landsat-8, NIR refers to band 5 (0.845 to 0.885  $\mu\text{m}$ ) and RED refers to band 4 (0.630–0.680  $\mu\text{m}$ ) [49]. NDVI is given by

$$NDVI = \frac{NIR - RED}{NIR + RED} . \quad (5.1)$$

The NDVI index ranges between -1 and 1, where -1 indicates moisture stressed vegetation and a value closer to 1 indicate healthy vegetation. NDVI can thus

be used to identify healthy vegetation. NDVI can be used by farmers to monitor their crops, and help identify which part of the crops that needs water or fertilizer. Other applications include control the damage of forest fires, and for large scale vegetation monitoring of for example rain forests [24, 31]. Examples of other VIs include soil-adjusted VI, weighted difference vegetation index and nonlinear VI [8].

## 5.2 Spectral Angle Mapping

Spectral Angle Mapping (SAM) is a much used method for hyperspectral vegetation classification. SAM compares the individual spectra with the image spectra and measures the spectral angle between them. The pixel is assigned to the vegetation class that it has the smallest spectral angle with. For other pixels, their similarity can be measured by the difference in spectral angle, where a small difference indicate high similarity. The result of SAM is affected by many factors, by which mixed pixels pose a big challenge, typically because most images include many mixed pixels [5, 24].

## 5.3 Machine Learning

Machine Learning (ML) methods are popular for use in classification of remotely sensed ocean data [31]. ML methods are complex methods that can learn the best model for different types of data. This means that it is possible to tune different models to fit the local variations, which is often observed when analyzing remotely sensed ocean data [31].

ML methods are often divided into supervised and unsupervised methods. Unsupervised ML methods find patterns in the data and clusters different data groups together. A popular unsupervised method is K-means. Supervised ML methods learn from example data and aims to map an output from new input data that corresponds to the pattern in the example data. This means that to train a model, an annotated dataset with true class labels is needed. Examples of popular supervised ML methods include Random Forest (RF), Support Vector machines (SVM) and Convolutional Neural Networks (CNN) [50].

This section provided an overview of different methods that can be used for hyperspectral vegetation classification. The choice of method needs to be considered, where the data type, application area and time requirement for the process flow are some of many factors that needs to be considered.

# /6

## Machine Learning for Marine Vegetation Mapping

### 6.1 Machine Learning

#### 6.1.1 An introduction to Machine Learning

In the last decade, ML for vegetation mapping has gained an increasing popularity as it has shown good results in a large number of tasks. ML methods are complex methods that can learn statistical patterns in data, and improve their performance with the inclusion of new representative data [50]. ML for marine vegetation mapping is the main focus in this thesis, this section therefore gives an introduction to ML and typical challenges and considerations that need to be taken.

ML methods can be divided into two main groups - supervised and unsupervised methods. Supervised ML methods use training data with true class labels to learn the most suitable relationship to predict the given classes. Unsupervised ML methods use unlabelled training data, where they predict clusters with data that are similar to each other using for example distance measures [51]. ML uses the features in the data to learn the best way of predicting the patterns in the data. A feature in ML is defined as a measurable and individual property of

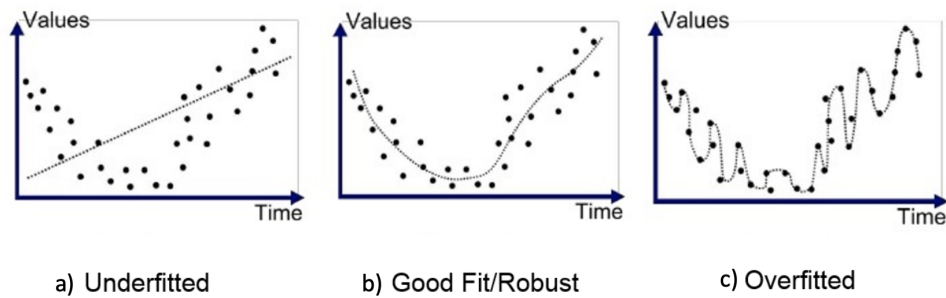
a phenomenon. A feature vector is a vector of features that holds information about an object in the dataset [50]. In hyperspectral imaging, a feature vector can be for example a stack of spectral bands, a combination of spectral bands or a transformation of spectral bands, coordinates, bathymetry etc.

When training a ML method, the dataset is often split into three parts; training, validation and testing. The splitting of data is used to establish a base of training the ML method and then validating/testing the method's performance. The training dataset is used to train the method and fit and optimize the parameters that best predicts the patterns of the dataset. The training dataset should thus be assigned the largest part of the data in order to effectively learn the pattern of the data. The validation dataset is used to tune the hyperparameters of the method in order to enhance the performance of the method. Hyperparameters of a ML method are the parameters the user can adjust in order to control the learning process. The test dataset is used for unbiased evaluation of the established model, i.e. evaluation of the final model [50, 52]. The way the dataset is split into these three parts is important. The split ratio varies and needs to be adjusted for the problem at hand. An example of how to divide the dataset could be 80/10/10, where training is 80% of the dataset and validation and test is 10 % of the dataset each. All parts should contain a balanced amount of data points from each class, such that the model is trained, validated and tested for all classes to avoid overfitting or underfitting of certain classes [52,53]. In this project, the dataset is split into a training part (70 %) and validation part (30 %). Then the methods are applied to an independent dataset for robustness evaluation.

Overfitting and underfitting are common problems in ML. Fig. 6.1 shows an illustration of underfitting in (a), a good fit in(b) and overfitting in (c). As the figure shows, in the case of underfitting, the model is not able to identify the pattern in the dataset and occurs when the model is too simple for the dataset. In Fig. 6.1 (a), a linear model is predicted, when in fact the dataset has a quadratic shape. Fig. 6.1 (b) shows how the model should be predicted, it follows the trend of the data, without perfectly matching the data at every point. In this situation, the model is most likely going to perform well on other similar datasets as well. Fig. 6.1 (c) shows the case of overfitting. Here the model almost perfectly fits the dataset. This leads to a very high accuracy for this dataset, but the model does not perform well on other datasets [50, 51, 54]. Ways to improve underfitting is to include more training data and increase the complexity of the model. To deal with overfitting, the model could be trained in iterative epochs and use early stopping [50].

Uneven distribution of classes in the dataset is a common problem in ML. A significant imbalance in the dataset can cause problems in training the model, as the model is mostly trained on the classes with the most samples. This





**Figure 6.1:** Illustration of (a) underfitting, (b) a good fit and (c) overfitting. As (a) shows, underfitting occurs when the model is too simple to be able to predict the pattern in the data. (b) shows a good fit and a robust model which is obtained when the model has good performance on training data and at the same time is generalized and performs well on new, unseen data. (c) shows a model with good fit to the training data, but it is not generalized and therefore has poor performance on new data. Figure reused from [55].

causes a biased model who is overfitted to the major classes, and underfitted to the minor classes. A way to deal with imbalance is to under-sample or over-sample the dataset. The goal of these techniques is to create an equal amount of samples for each class. Under-sampling techniques under-samples the largest classes, while over-sampling techniques over-samples the smallest classes [56].

When training ML models, the optimal model is found by adjusting the models hyperparameters. There are several available hyperparameter optimization methods. For instance, it can be performed by testing the model for a range of different hyperparameters, and recording how well the model performs. The set of hyperparameters that gives highest score in a chosen performance measure, can be seen as the optimal hyperparameters for the model, applied to the specific dataset at hand. Popular methods that are used in the search of the best hyperparameters include grid search and randomized search. Both methods search through a set of hyperparameters that the user wishes to evaluate [57].

A popular way to evaluate the performance of supervised ML methods is by the confusion matrix. The confusion matrix shows how the dataset is classified and what types of errors are being made by the model. Figure 6.2 shows how a confusion matrix appears for a multi class problem. There are three classes; A, B, C. TP represents the True Positive classifications, i.e. the number of pixels (in an image context) that belong to class A, and that are classified as class A. The error terms  $E_{XX}$  represent the number of pixels that are wrongly classified.  $E_{BA}$  are the number of pixels that belong to class B and are classified as class A.

		Predicted		
		A	B	C
Actual	A	TP <sub>A</sub>	E <sub>AB</sub>	E <sub>AC</sub>
	B	E <sub>BA</sub>	TP <sub>B</sub>	E <sub>BC</sub>
	C	E <sub>CA</sub>	E <sub>CB</sub>	TP <sub>C</sub>

**Figure 6.2:** Illustration of how a confusion matrix for a multi class problem appears. TP are the true positive predictions, while E<sub>XX</sub> represent the prediction errors.

False Negatives (FN) for a class is found by the sum of the row except the TP value. For class A, this would be  $FN_A = E_{AB} + E_{AC}$ . To find the False Positives (FP) for class A, the sum of the column, minus the TP value, belonging to class A is computed;  $FP_A = E_{BA} + E_{CA}$ . The True Negatives (TN) for class A is given by  $TN_A = TP_B + E_{BC} + E_{CB} + TP_C$ , that is the sum of all columns and rows, except for the columns and rows for class A. The Overall Accuracy (OA) for the models prediction is given by

$$Overall\ Accuracy = \frac{\sum_{i=0}^X TP_i}{N}, \quad (6.1)$$

where X is the classes in the dataset, in this case  $X = \{A, B, C\}$ , and N is the total number of data points in the dataset [24, 53] (in an image context, N is the total number of pixels). The Average Accuracy (AA) can be computed by [58]

$$Average\ Accuracy = (TP + TN)/(TP + TN + FP + FN). \quad (6.2)$$

The Cohen's Kappa statistics can also be computed from the confusion matrix. The Kappa statistics compares the OA with the Expected Accuracy (EA). The expected accuracy is the accuracy a random classifier would have based on the given confusion matrix. The Kappa statistics, K, is given by

$$K = \frac{OA - EA}{1 - EA}. \quad (6.3)$$

Eq. (6.3) gives the models ability to classify true positives while controlling for the accuracy of a random classifier based on the given confusion matrix. The Kappa statistic ranges between -1 and 1 where a value of 1 means that

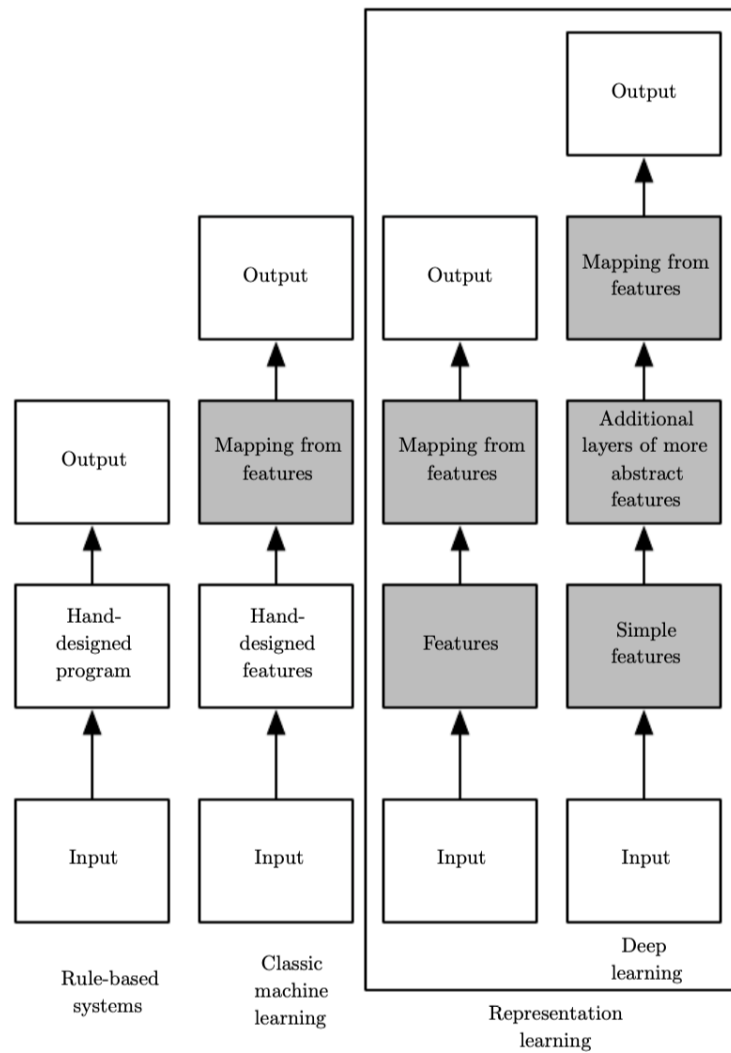
the model has complete agreement. A lower value means less agreement. The ideal model would have a Kappa value of 1 [59].

Although ML methods face the challenges described in the previous paragraphs, ML methods offer a wide range of benefits, compared to simple rule-based systems. ML methods can detect patterns and trends in the dataset, while also handling a multi-variety data and multi-dimensional input. They can be applied to a wide range of tasks. If a method is built in such a way that the user can feed in an input image and receive a classified image from the ML method, then the user involvement is minimal. This automatic process saves time and involvement from humans, which can be beneficial in many cases, especially in the age of "big data" [50, 53].

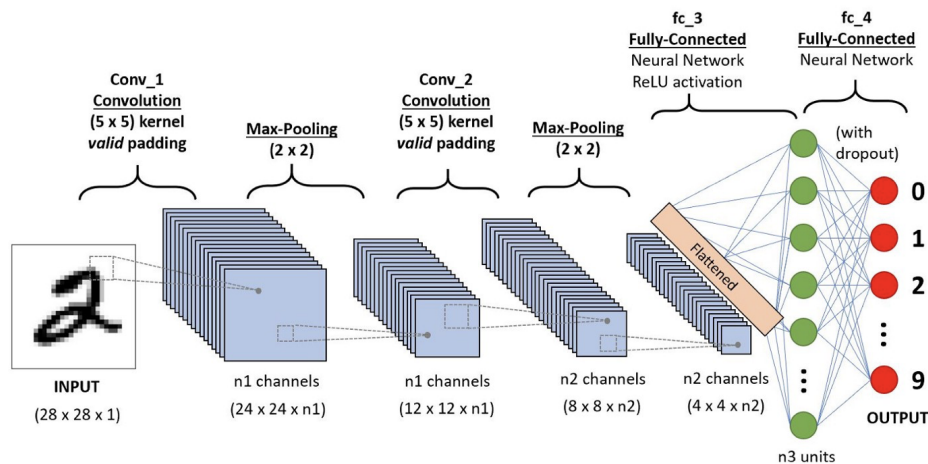
## Deep Learning

Deep Learning (DL) is a subset of ML that can analyze complex problems by learning and making intelligent decisions by mimicking the structure of the human brain. It has been shown through research that DL methods can provide a high accuracy when classifying hyperspectral datasets into multiple classes [26, 28, 60]. DL has an increased number of feature extraction layers compared to the classical ML methods such as SVM and RF. Figure 6.3 shows an overview of how different classification methods can be divided into groups dependent on their complexity. It shows how rule-based systems consists of hand-designed programs which makes a decision based on simple rules, such as the VIs. Classic ML are more complex and consists of feature learning part (the grey box). Then, Representation learning, which is the category DL falls into, have 2 or more feature learning layers that learns the patterns in the data [25].

CNN is a DL method that can be used for remote sensing data. CNNs consist of a series of layers, where the output from one layer is the input to the next. For image analysis, the input to the first layer is the image itself. For each layer, the value of an output "pixel" in the next layer is calculated based on a local neighborhood of pixels, rather than the whole layer. For hyperspectral data, CNNs also considers the spectral information of all pixels in the neighbourhood. This allows for a simultaneous extraction of spatial and spectral information. Research have shown that this method of feature extraction improves the classification of hyperspectral images, relative to the typical pixel-per-pixel classification that traditional ML methods and rule-based methods perform [28]. Figure 6.4 shows an example of how a CNN can be constructed, with convolution layers, pooling layers and fully connected layers. An image with a class label is fed into the network where it undergoes feature extraction and classification. The output is a class label. There are a number of hyperparameters in a CNN



**Figure 6.3:** A comparison of Rule-based systems, ML systems and representation learning systems. Rule-based systems are simple systems which only have a handcrafted part, like Vegetation Indices. ML methods have learning part (Mapping from features) as well as the handcrafted part. Representation learning methods (which includes DL) have a deeper and more complex structure, with additional layers of learning and feature analysis. The grey boxes represent the parts of the methods that is fitted to the training data. Figure reused from [25].



**Figure 6.4:** Example of how the architecture of a CNN might appear. An input image is fed into the network where the feature extraction process occurs in the convolutional and pooling layers, and the classification process occurs in the fully connected layers. Figure reused from [61].

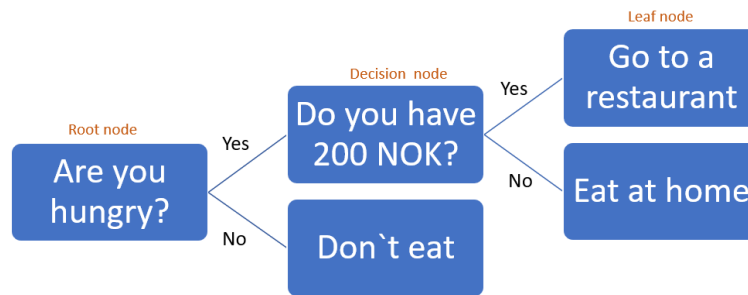
that needs tuning, such as the kernel size of the different layers, learning rate and activation functions [25].

## Dimensionality reduction

When working with high dimensional datasets it is often desirable to reduce the number of dimensions so that only the most important features are included. Dimensionality reduction reduces redundant information and the time and resources needed for data processing and data management [50]. Hyperspectral datasets typically contain several hundreds spectral measurements per data point. Working with all the spectral measurements requires a lot of computational resources and can be time consuming, at the same time as many of the spectral measurements can hold redundant information. Dimensionality reduction is thus often performed on hyperspectral datasets to extract the most important features [23, 26, 28].

### 6.1.2 Machine Learning and dimensionality reduction methods used in this project

This section will give an introduction to the different ML methods that are used in this project, namely Random Forest (RF), Support Vector Machine (SVM), and Convolutional Neural Networks (CNN). RF and SVM have shown



**Figure 6.5:** Example of a simple decision tree. A person that has the following attributes [Hungry = Yes, Have 200 NOK = no] is classified to be in the group that have to eat at home. The decision tree consists of three parts; the root node, decision nodes and leaf nodes.

good results in classifying hyperspectral data for various marine vegetation mapping [3, 4, 26]. CNN has also shown some promising result in hyperspectral data classification [26, 28]. The dimensionality reduction method used is PCA, which is often used for sea bottom vegetation data and hyperspectral data with good results [23, 26, 28, 62].

## Random Forest

RF is a popular supervised ML method due to its high performance among various applications. RF can be used both for prediction and regression. RF is based on decision trees, where it creates different trees from different samples and the majority votes decides which trees are kept for building the classifier. A decision tree is non-parametric method which uses simple decision rules learned from a dataset, to predict a value for a new data point. The complexity of the model depends on the depth of the tree. To demonstrate how decision trees works, an example is shown in Fig. 6.5. Figure 6.5 shows a shallow decision tree with a simple structure. The decision tree classifies a person into three classes; Go to a restaurant, Eat at home and Don't eat. If a person in this dataset has the following features: [Hungry = Yes, Have 200 NOK = No], then it is assigned class Eat at home [53].

A random forest consists of a large number of random decision trees, where each decision tree predicts a certain class for each input. The class with most of the votes is saved as the prediction of the model. Assume two classes; class 1 (Sand) and class 0. If few decision trees predict class 1 (Sand) and most of the decision trees predict class 0, then class 0 is saved as the prediction. This is repeated for a random set of data points from the dataset (sampled with replacement). The process of choosing data points and deciding which model

if best suited, is called Bootstrap aggregation. Combining the results from all decision trees, the final model is chosen by considering the majority of the votes. This sums up the RF algorithm [53].

By the use of multiple decision trees during the training process, RF is less prone to overfitting than other methods. However, a large number of decision trees increase the training time due to the increased complexity of the model [53].

There are several important hyperparameters to consider when training a RF classifier. Examples include the number of trees in the forest, the maximum depth of the trees and the number of samples that is required for a node to split. The specification of these hyperparameters are important because they limit the model and define the complexity of the model. They can also affect the potential for over- and underfitting [53].

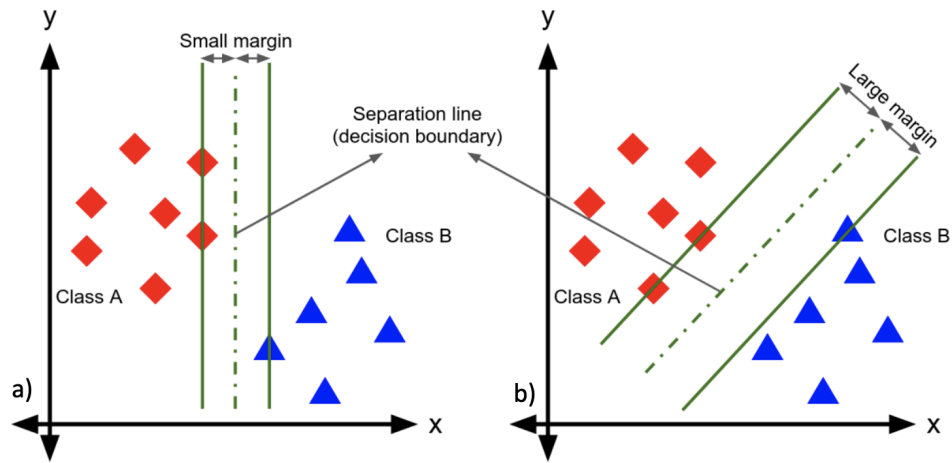
## Support Vector Machine

SVM is a supervised ML method which tries to find the best hyperplane to differentiate between classes. The simplest case of SVM is when it is used to classify two linearly separable classes, as seen in Fig. 6.6. Figure 6.6 show two classes, A and B, that are separated by two different hyperplanes (the green dotted line) [63]. The hyperplane (also referred to as decision boundary) is given by

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = 0, \quad (6.4)$$

where  $\mathbf{x}$  holds the feature vectors of the training dataset,  $\mathbf{w}$  is the vector normal to the hyperplane and  $w_0$  is the intercept of the hyperplane. This equation have multiple solutions, and does not represent a single hyperplane. In Fig. 6.6 (a) the separability of the classes is satisfying but small. Fig. 6.6 (b) has a larger separability of the classes, and it is more likely that a new data point will be classified correctly in Fig. 6.6 (b) than in Fig. 6.6 (a). To evaluate different hyperplanes, the margin is calculated. The margin is the smallest distance from the hyperplane to the closest data points from each class. As seen in Fig. 6.6, the margin in (a) is smaller than the margin in (b). Thus, the hyperplane in Fig. 6.6 (b) a better solution to Eq. (6.4) than the hyperplane in Fig. 6.6 (a). The data points that lie on the margin in Fig. 6.6, are known as the support vectors. These vectors define  $\mathbf{w}$ , in the way that they define where the margin is placed.

For non-linear decision boundaries (if the classes cannot be separated by a



**Figure 6.6:** An illustration SVM in the case of two separable classes with small margin (a) and large margin (b). Decision boundary in (b) is the preferred due to the large margin, thus a smaller chance of misclassification of a new feature vector. Figure reused from [63].

linear decision boundary), kernels can be used to compute high dimensional features, applying the kernel trick. Instead of applying transformations to represent the data in a high dimensional feature space, the kernel trick uses similarity comparisons between the original data points to represent the data in a lower dimensionality space. In other words, the kernel trick simplifies the problem at hand by not explicitly transforming the data into a high dimensional feature space. Examples of kernels that can be used in SVM is the linear kernel, nonlinear kernel and the Radial Basis Function (RBF) kernel [50].

If the classes are not separable, the slack variable,  $C$ , is used to allow certain (outlier) data points to cross the margin and be misclassified. In the case of  $C = 0$ , the margin is "hard" where no misclassification is allowed. Increasing the value of  $C$  allows some outlier data points to be misclassified. This makes the classifier more flexible where a more suitable decision boundary can be found to the problem at hand [50].

For multiple classes the *one against all* or *one against one* approach is used. The *one against all* approach splits the multi class problem into multiple binary classification problems. The *one against all* splits the dataset into binary classification problems for each class-pair. This means that the one against one approach demands more models than the one against all method. The increased number of models demanded by the one against one method, can make it slower and more demanding than the one against all approach. Both of these approaches requires an multiple classifiers to be trained, which might slow



down the algorithm. Another drawback of these methods is that some areas in the feature space might be intermediate, which means leads to feature vectors that fall in these regions might have several  $g_i(\mathbf{x})$  that are positive [50].

Some of the hyperparameters for the SVM algorithm are associated with the kernel function. As mentioned, there are several available kernels for SVM. Other hyperparameters include the slack variable, which might need tuning to achieve the desired accuracy [50,64].

## Convolutional Neural Network

CNN is a supervised ML method within DL that specialize in (among other things) image recognition. The architecture of a CNN varies and needs to be adjusted to the application, but common for all variants of CNN is the convolutional layers. For this project, two convolutional layers, two maxpool layers and two fully connected layers were used.

A CNN can be divided into two parts; the feature extraction step and the classification step. The convolutional layer and the pooling layers both extract features from the input image, hence they are a part of the feature extraction step of the network. The fully connected layers are a part of the classification step of the network.

The convolutional layer applies the convolution operation to extract features from the input image. Figure 6.7 shows an illustration of a 5x5 image that is convolved by a 3x3 kernel. The output is a new 3x3 matrix. The kernel is a matrix with a smaller dimension than the input image. Here, the kernel is the matrix in the middle. The kernel "slides" over the input image, and at each position, the output value is equal to the sum of elementwise multiplication between the kernel and the part of the image currently covered by the kernel. Figure 6.7 shows how the first pixel of the output image is created by convolving the input image with a 3x3 kernel. The 2D convolution operation is given by

$$S(m, n) = (K * I)(m, n) = \sum_i \sum_j I(m - i, n - j)K(i, j), \quad (6.5)$$

where  $S(m,n)$  is a pixel in the output image,  $K$  is the kernel and  $I$  is the input image.  $(i,j)$  is the dimension of the kernel and  $(m,n)$  is the location in the input image. In Fig. 6.7,  $S(2,2) = 6$ . As seen from the figure, the dimension of the input image is 5x5 while the dimension of the output image is 3x3. This is because the kernel performs computations only when the whole kernel covers pixels from the image. In the case of a 3x3 kernel, the first computations are made when the center pixel of the kernel lays over pixel (1,1) in the input image (i.e. the pixel inside the green field with value 5 in Fig. 6.7) due to null-based

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

 $*$ 

1	0	-1
1	0	-1
1	0	-1

 $=$ 

6		

$$7 \cdot 1 + 4 \cdot 1 + 3 \cdot 1 + 2 \cdot 0 + 5 \cdot 0 + 3 \cdot 0 + 3 \cdot (-1) + 3 \cdot (-1) + 2 \cdot 1 = 6$$

**Figure 6.7:** Illustration of how 2D convolution works. Without zero-padding, the dimension of the 2D array is reduced from 5x5 to 3x3 with a kernel of dimension 3x3. While the kernel slides over the image, the output array will be filled with values from the convolution operation. The figure shows how the first value in the new array is calculated. Figure reused from [66].

indexing. A way to counteract this is by adding zero-padding around the input image. Zero-padding refers to adding rows and columns of zeros around the rectangular edge of the image. This procedure increases the input image size by the number of rows and columns of zeros that is added. Looking at Fig. 6.7, zero-padding could be used to increase the dimension of the input image from 5x5 to 7x7. This way, the first computation would start on  $\text{Image}_{7 \times 7}(1,1)$ , which would be pixel (0,0) in the 5x5 image. Then, the output image would have a dimension of 5x5. By using zero-padding, all pixels in the input image can be used in the convolution operation and the input image dimensions are kept identical to the input image dimensions, which is an advantage in some cases [25, 65].

The motivation for using convolutional layers in a network is, among others, the sharing of parameters, sparse interactions and equivariant representations. Sharing of parameters refers to the use of one parameters more than once in the model. Sharing of parameters leads to the need of only learning one set of weights for the model, and not a separate set of weight for each location in the model. The consequence of this is a lowering of memory storage compared to traditional neural nets that use dense matrix multiplication. The model will also have less parameters to train, which makes it less prone to overfitting on training data. This sharing of parameters, lead to equivariance of the convolutional layer, which means that if the input of the convolution changes, then the output of the convolution operation changes with the same property. The convolution operation creates a 2D map of where the features appear in the original image. If the input object is moved, then the feature representation will be moved the

same amount in the output. This property is useful for detecting features in the image that we know can be represented by a small neighbourhood of pixels. If these features appear across the image, then sharing of parameters all over the image is useful. Sparse interactions, have the advantage of being able to detect small, but meaningful features in an image. These features can be edges or corners, and can be detected by kernels that are small compared to the image size. The use of sparse interactions decrease the amount of parameters that needs to be calculated and stored, and thus reduce the memory requirements for the model [25, 67].

After a convolutional layer, it is common to use some kind of pooling layer as the next part of the network. In this project, the maxpool layer is used. The maxpool operation reports the maximum value within a rectangular neighbourhood of pixels that slides over the image. If the neighbourhood is 2x2 pixels, then those 4 pixels are reduced to 1 pixel. This reduces the dimension of the feature maps, which in turn improves the computational efficiency of the network. Another point to notice about the pooling operation, is that it is invariant to translation. This means that if some of the input values are translated by a small amount, the output will, in almost all cases, be the same as before the translation. This is handy when the goal is to identify if a feature is present, and its position is less important [25].

After feature extraction by convolutional layers and maxpool layers, fully connected layers are frequently used as the last part of the network. Fully connected layers uses the feature analysis performed in the previous layers, and applies weights to try to predict the correct label for the input data. Before applying a fully connected layer, the 2D feature map from the last convolutional (or pooling) layer is flattened into a 1D vector (by stacking the matrice columns). The 1D vector shape is the input shape required for a fully connected layer. Each neuron in the fully connected layer is connected to each instance in the input vector. This way the neuron in the fully connected layer changes if some of the neurons in the previous layer changes. The information flow from input layer to the output of a fully connected layer is strong and the class prediction is truly based on all values in the input vector [25, 67].

Activation functions are used in a CNN to decide whether the output of the layer should be activated or not. This means that they help decide if the input of the neuron is important and needs to be considered in the proceeding process, and thus also suppress irrelevant information. There are several different activation functions available, such as the Rectified Linear Unit (ReLU), Softmax, Hyperbolic tangent function and so on. All these have in common that they take input data and projects it onto a certain scale, such as  $[0, \infty)$  for the ReLU activation function. In this project, the ReLU and Linear activation functions are used in the CNN. ReLU is often a popular choice due to its simplicity and

because it is computationally efficient. The ReLU activation function is given by [67]

$$f(x) = \max(0, x). \quad (6.6)$$

What Eq. (6.6) describes is that if the input of the node is less than zero, then it is deactivated, and if they are bigger than zero, then they will be activated. The linear activation function is used in the last fully connected layer of the network. Note, several others were tested, but the linear activation function turned out to be the best one for this problem. The linear activation function is given by  $f(x) = x$ , which means that the output of this activation function is not confined by a range, such as the ReLU activation function. The input value,  $x$ , will also be the output value.

The loss function of a neural network is used to determine how well the network performed on the dataset and is a necessary part of the training of the network. The training of a network is an iterative process where the loss function is minimized. If the network has a high performance, the output of the loss function will be low, and vice versa. Examples of loss function include mean squared error, likelihood loss and log loss. The loss function used in this project is the Sparse Categorical Crossentropy loss function. This loss function can be used for multi class cases where it computes the categorical cross-entropy of the predictions [68].

Lastly, the optimizer of the CNN is an important factor to consider when building a network. The optimizer of the network is used to change the attributes in the network to decrease the loss function. Several optimization functions are available, such as the gradient descent, stochastic gradient descent and the Adagrad optimization function. In this project, the Adam optimizer is used, which is a stochastic gradient descent method [69].

### **Principal Component Analysis**

PCA is a dimension reduction method that performs a linear transformation in order to extract the features that preserves the most variation of the dataset. PCA use eigendecomposition of the correlation matrix,  $R_x$  to extract the eigenvectors,  $e_i$  which are uncorrelated vectors that preserves the highest possible variation of the data. An assumption for using PCA is that the dataset has a zero mean. If the dataset does not have a zero mean, the mean is subtracted before the PCA procedure is performed. This procedure is referred to as a standardization

of the dataset. The correlation matrix for  $\mathbf{x}$  (the dataset) is defined as

$$R_x = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k \mathbf{x}_k^T, \quad (6.7)$$

where  $n$  is the number of data vectors and  $k$  is the  $k$ -th data vector.  $T$  indicates the transpose of the vector. If one needs to subtract the mean from the dataset, this corresponds to the covariance matrix, given by

$$\Sigma_x = R_x - E[\mathbf{x}]E[\mathbf{x}]^T \quad [50]. \quad (6.8)$$

The next step is to compute the eigenvectors of  $R_x$  (or  $\Sigma_x$  if centering is needed). To find the eigenvectors, one uses the equation

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{e}, \quad (6.9)$$

where  $\mathbf{A}$  is the square matrix (either  $R_x$  or  $\Sigma_x$  in this case),  $\mathbf{e}$  is the eigenvector and  $\lambda$  is the eigenvalue that makes Eq. (6.9) true [70]. The value of the resulting eigenvalues,  $\lambda_i$ , specify the variance along the corresponding eigenvector  $\mathbf{e}_i$ . In order to preserve as much variance as possible from the original dataset, the eigenvalues are sorted and the ones with the highest values (and their corresponding eigenvectors) are used [50]. In order to evaluate how many eigenvectors that is relevant to use, one can look at how much of the variance the eigenvectors represent and use a threshold of for example 90% or 95% of the total variance. This can be done by summarizing all the eigenvalues (their sum equals the total variance) and then dividing the eigenvalue by the sum of all the eigenvectors. If for example 3 of the eigenvalues make up 95% of the total variance, the use of the 3 corresponding eigenvectors might be sufficient. It is also important to consider what type of data and what it is going to be used for, in order to decide the relevant number of eigenvectors [50].

## 6.2 Data acquisition

### 6.2.1 Drone imaging

In this project the data source is drone imagery captured with a hyperspectral sensor. A drone is an unmanned aircraft that can be controlled remotely. The use of drones dates back to the first World War and initially the development has been driven by military use [71]. In the last decades the use of drones for civilian purposes have increased rapidly. Today, drones have broad applications including surveillance of vegetation. The use of drones to monitor the health of crops is popular among farmers, as well as for governments, who use the technology for forest and wildlife management [72]. There are several

other applications as well, where one of them is mapping and monitoring of underwater vegetation.

There are several reasons for using drone imagery for mapping of underwater vegetation. Operational coast and data availability are some of the major advantages of drones vs. satellites and air crafts. Due to the frequent cloud cover in the Arctic environment, optical satellite imagery for continuous surveillance of the Norwegian coast (especially in the north) is not always reliable [21]. Air craft imagery is often expensive, and requires a significant amount of resources. The low flight height of drones (up to 50 meters in this project) of drones have several beneficial properties. The absence of atmospheric effects on such a low flight height is one of the major advantages of drone imaging, compared to satellite imagery. As described in Sec. 2.3, the atmospheric effects on imagery acquired less than 50 meters above the surface can be neglected. High spatial resolution is also an advantage; the data used in this project has a spatial resolution of 4.5 cm per pixel. The good spatial resolution is a big advantage when mapping underwater vegetation, because it makes it possible to classify the vegetation extent of the underwater vegetation with a higher accuracy. Another benefit of good spatial resolution the reduction of mixed pixels, which is a problem especially in the coastal areas (see Ch. 3).



**Figure 6.8:** Field campaign at the Norwegian coast; Silje Grue (left) and Sigfinn Andersen (right). Photo: Martin Skjelvareid



Conducting a field campaign for capturing drone data requires careful planning. The planning especially needs to consider the local regulations as one might have to apply for permission to fly the drone in specific areas. The team also needs to ensure that all the needed software and hardware is available and operational for the mission. The weather conditions needs to be ideal for the drone, especially with respect to the wind conditions and precipitation [23]. Figure 6.8 shows a field work at the Norwegian coast (Saltstraumen) of the MASSIMAL project that I participated in when all conditions for data collections were ideal.

The drone used in this project is a MATRICE 600PRO [73], with a Pika L camera [74] mounted on the drone by a gimbal to stabilize the camera. Figure 6.9 shows the drone with the camera mounted, ready for a field campaign. The drone and camera was configured in such a way that when the drone pilot guided the drone into the planned imaging area at the correct height, the drone started to fly in a predefined pattern while imaging. The camera is a pushbroom sensor, where each camera frame is a single line of data captured while the drone flies forward. To convert the lines into a full 2D image the lines are stacked next to each other [74].



**Figure 6.9:** Illustration of the drone used for imaging sea bottom vegetation capturing data for the MASSIMAL project. Photo: Martin Skjelvareid

## 6.2.2 The data

The data in this project was acquired by the drone and sensor described in Sec. 6.2.1. Georeferencing, radiance correction and sun glint correction was done by the MASSIMAL project. Although the Pika L camera measures wavelengths up to 1000 nm, all measurements over 749 nm was removed from the dataset. The measurements above 750 nm was used for sun glint correction (see Sec. 2.4). The light above 750 nm is mostly absorbed by the ocean, thus it makes sense to remove this part of the measurements.

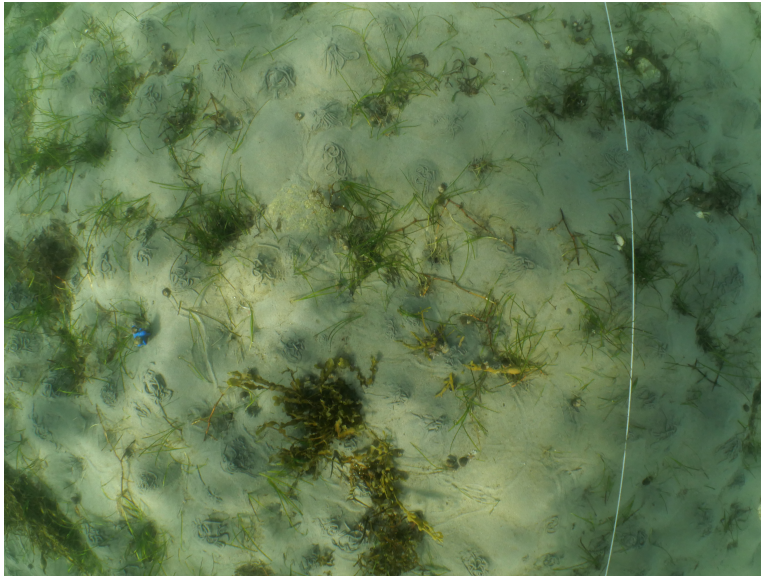
After preprocessing, the dataset was annotated by the MASSIMAL project, using the in-situ data collected simultaneously as image acquisition. Underwater images, as seen in Fig 6.10 were used as ground truth data. Figure 6.10 shows sea grass in the upper half part of the image, and some rockweed in the center bottom part of the image. The white line at the right in the image is a transect, which was used as a ground reference point during underwater imaging and annotation. The images were annotated in such a way that not every pixel in the image is assigned a class, but rather examples of pixels from each class present in the image are marked. Not all classes are present in all images. Figure 6.11 shows an example of how one of the images in the dataset is annotated. Figure 6.11 (a) shows the RGB image, and Fig 6.11 (b) shows the annotated image. As seen by the annotated image, there are relatively big areas of the image that are not annotated, and not all classes are represented in this image. Another factor to notice in this image, there is a lot more samples of sand than the other classes. This is also true for the rest of the dataset which creates class imbalance in the dataset, with the sand class being over-represented relative to the other classes.

The dataset consists of 12 images, where each image contains about 1500x400 pixels. This means that each image covers around 1215 square meters. The spectral measurements range from 390 nm to 749 nm, with 173 spectral bands and a bandwidth of 2.1 nm [74].

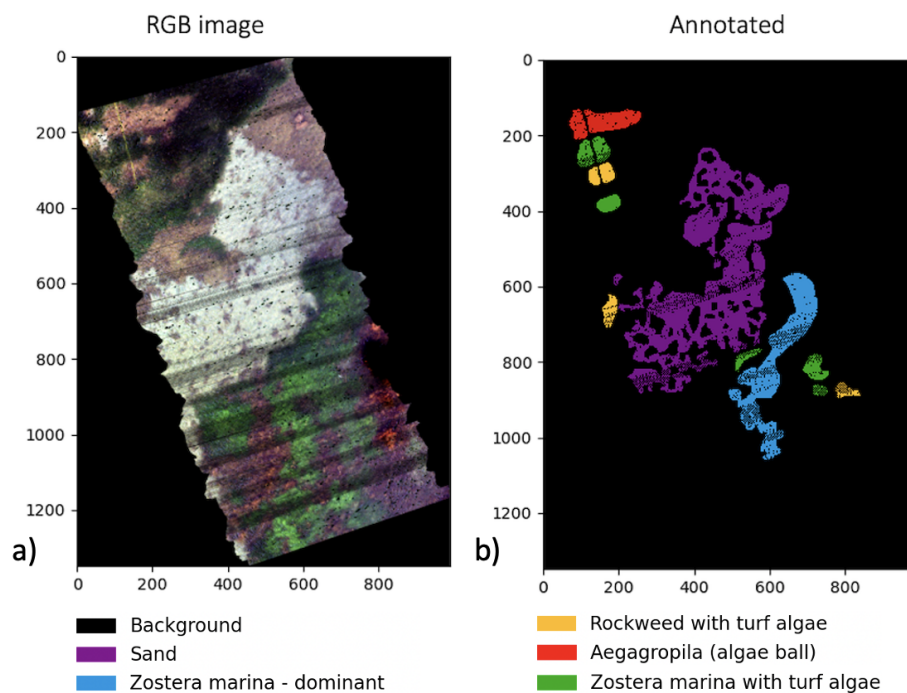
A last thing to notice about this dataset is that during the data collection the spectroradiometer measuring irradiance did not work properly, which in turn made it impossible to calculate the reflectance images in post-processing. The dataset therefore consists of measured radiance and not reflectance, which is what is normally used in ocean remote sensing, as described in Sec. 2.3. The here established methods should be retrained for remote sensing reflectance when reflectance measurements will be available.

Figure 6.12 shows the study site; to Larvik at Ølbergholmen (south east in Norway). The data was gathered on August 25, 2021 by the MASSIMAL project. Figure 6.13 shows where the transects were deployed. The drone was pro-

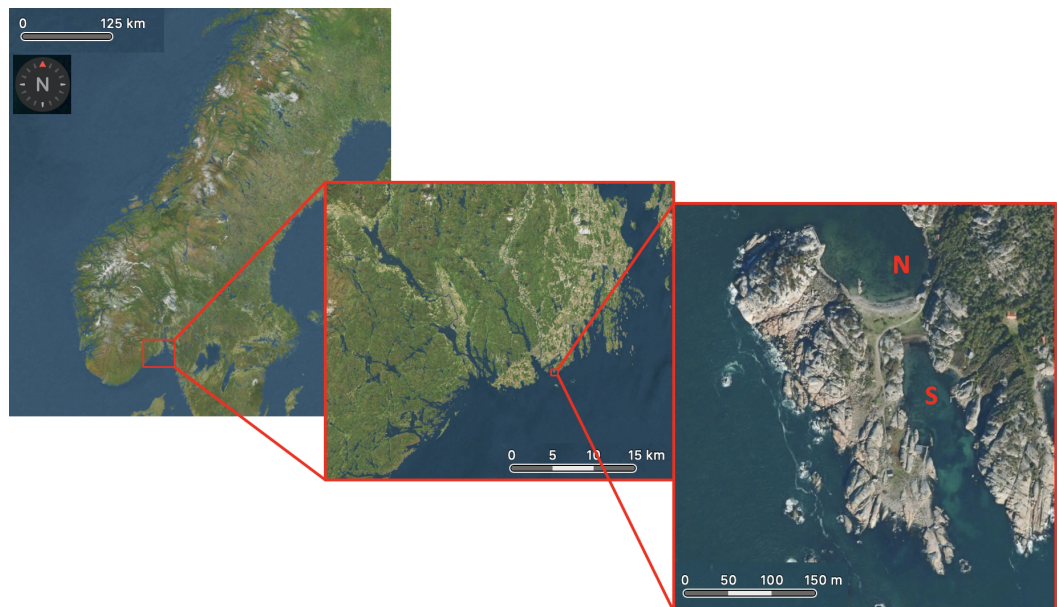




**Figure 6.10:** Underwater photo used in the annotation process. The white line is a transect, used as ground reference point during underwater imaging and annotation of the dataset.



**Figure 6.11:** Example of how the dataset is annotated. (a) RGB representation of one of the images in the dataset and (b) the image is annotated.

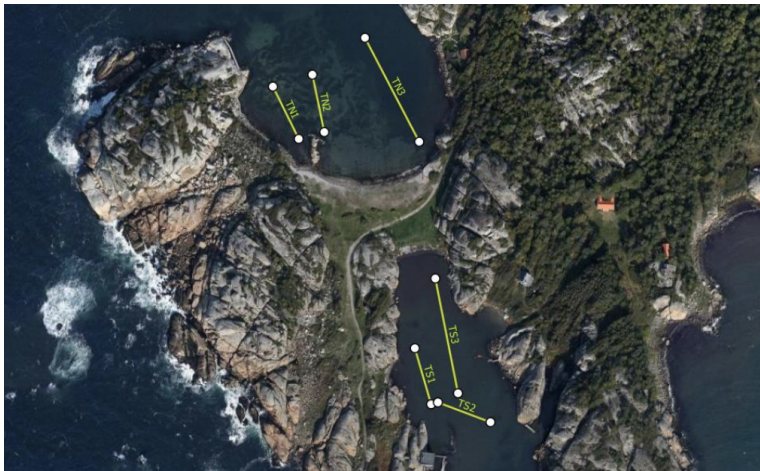


**Figure 6.12:** Study site: Ølbergholmen in Larvik. N = North, S = South. The field campaign collected data both in the northern part and southern part.

grammed to fly in such a pattern that all transects were imaged. In this project, data from the southern part was used for model training and testing. The algorithms were applied to one of the images from the South side for visual validation. Data from the northern part was used for evaluating the robustness of the methods. The image from the North side contained 21 000 annotated pixels, which was classified by the established methods. The data from the southern part consists of large areas with continuous coverage of the same species, while the northern part have smaller patches of the same species. Also, when capturing data from the northern part, there were more waves and the water appeared with more turbidity than the southern part. Because of this, the texture on the southern part appears more fine, while the northern part have a rougher texture (Sec. 2.3 describes how turbidity affects the measured signal). This may affect the classification of the northern part.

### Classes

The dataset consists of several classes, both sand and different vegetation types, as seen from Tab. 6.1. Table 6.1 shows that there is an imbalance in the dataset; class 1 (Sand) has 48 times as many samples as class 8 (Mixed algae classes). This was made up for by under-sampling (see Sec. 6.3). Figure 6.14 shows images of all the classes in the dataset, except for class 4 (Turf algae). Examples of Turf algae is rather shown in Fig. 6.14 (c) and (d) where the blurry parts in the images is Turf algae while the clear parts showing *Zostera Marina* and Rockweed, respectively. As seen from the images, the Turf algae appears



**Figure 6.13:** Map of where the transects were placed. The drone was programmed to fly in such a pattern that all transects were imaged. Credit: MASSIMAL project, Martin Skjelvareid

differently when attached to different species.

**Table 6.1:** The different classes in the dataset. Class 1 - 9 are the original classes in the dataset. Class 10 is a merged class made up of class 2 and 3. Class 11 is a merged class made up of class 5, 6 and 9.

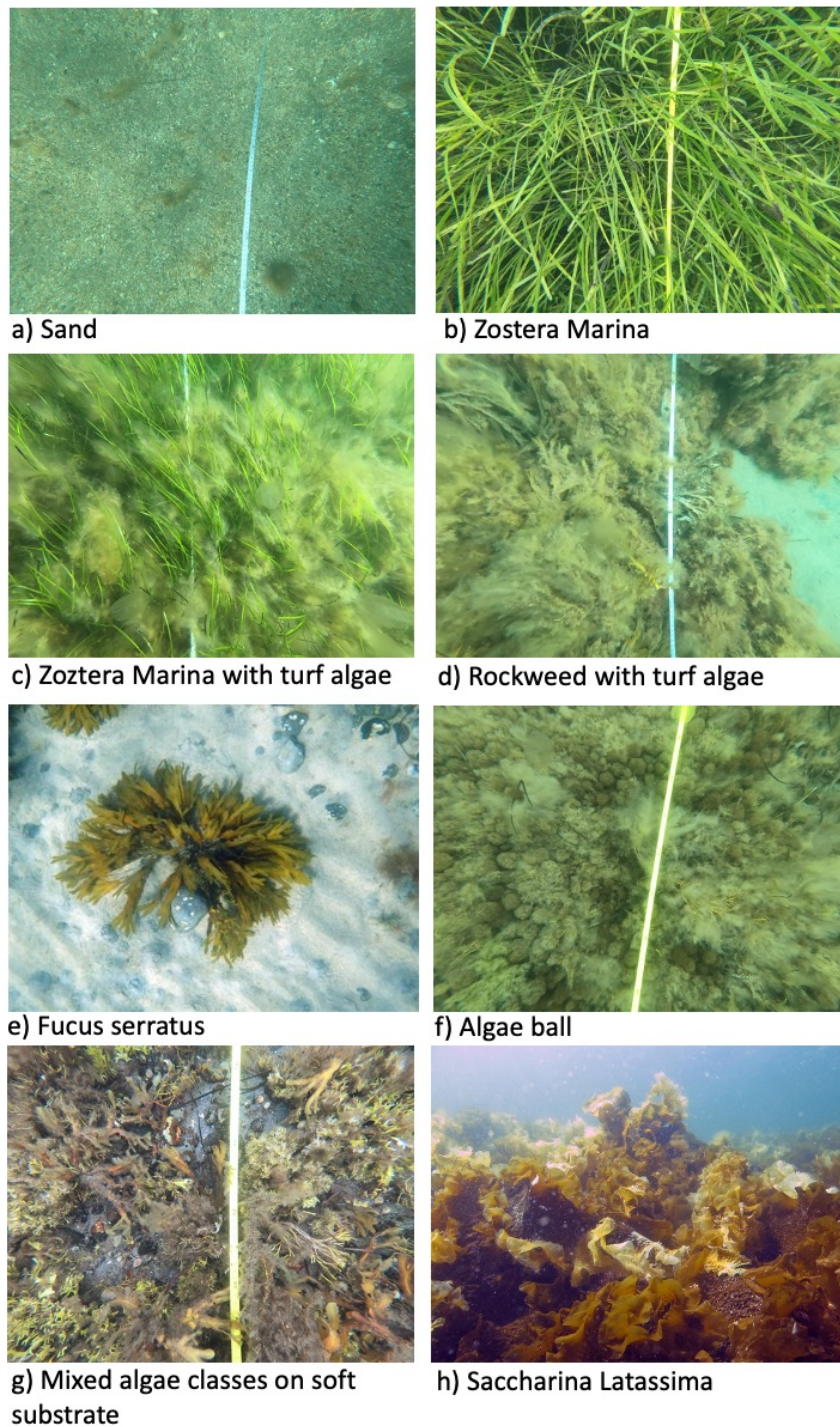
Class label	Class name	Number of samples
1	Sand	979 295
2	Zostera Marina – dominant	151 949
3	Zostera Marina with turf algae	282 007
4	Turf algae – dominant	39 790
5	Rockweed with turf algae	50 534
6	Fucus serratus	27 952
7	Aegagropila (algae ball)	24 700
8	Mixed algae classes on soft substrate	20 658
9	Saccharina latissima	20 903
Class label	Merged Class name	Number of samples
10	Zostera Marina (merged classes 3 and 5)	433 956
11	Kelp (merged classes 5, 6 and 9)	99 389

Classification was done on two versions of the dataset: the original dataset with all classes and a modified dataset where some of the more similar classes were merged. In the merged dataset two new classes are created; class 10 (Zostera Marina) and class 11 (Kelp). Class 2( Zostera Marina - dominant) and class 3 (Zostera Marina with turf algae) were merged into the new class, class 10

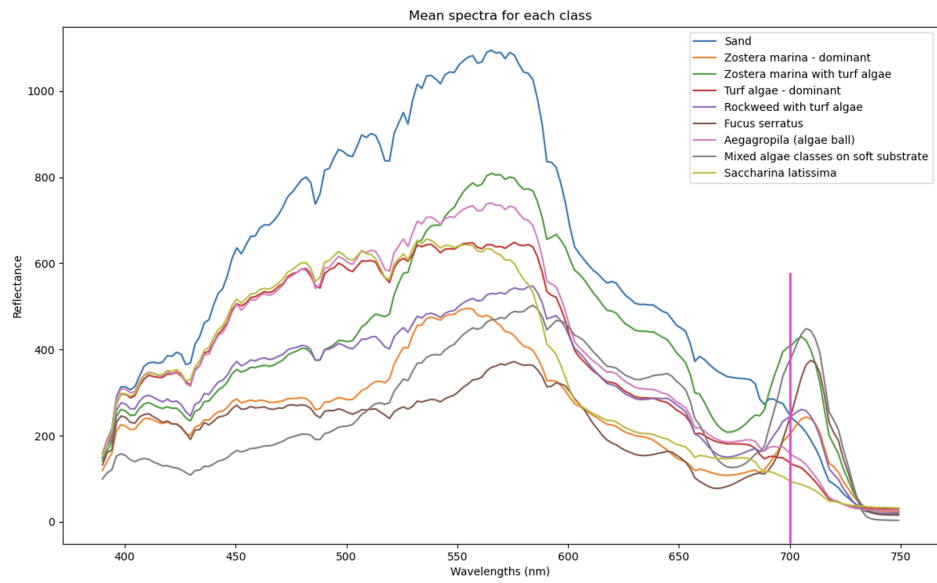
(*Zostera Marina*). Class 5 (Rockweed with turf algae), class 6 (*Fucus serratus*) and class 9 (*Saccharina latissima*) were merged into class 11 (Kelp). The reason for merging some of the classes was to see if the performance increases with fewer classes. It might be hard to identify different species of kelp, while it also could be difficult to distinguish between *Zostera Marina* and *Zostera Marina* with turf algae.

Figure 6.15 shows a plot of the mean spectral response for each of the classes in the dataset. As Fig. 6.15 (a) and (b) show, all spectra have different shape and their magnitude differs. These factors play an important role in the classification of the classes. At 700 nm the red edge is marked with a pink, vertical line in both plots. As Fig 6.15 shows, the red edge appears clearly for *Zostera Marina* (both dominant, with turf algae and merged), Rockweed with turf algae, *Fucus serratus* mixed algae classes on soft substrate and kelp. This difference in reflectance should at least make it possible to distinguish between the classes that have a spectral response that corresponds to the red edge. By further inspection of the spectrum plots, *Zostera Marina* - dominant and *Zostera Marina* with turf algae have similar shape of the spectrum, but *Zostera Marina* with turf algae have a larger magnitude due to the structure of the turf algae. This could cause problems for classification of these two classes.

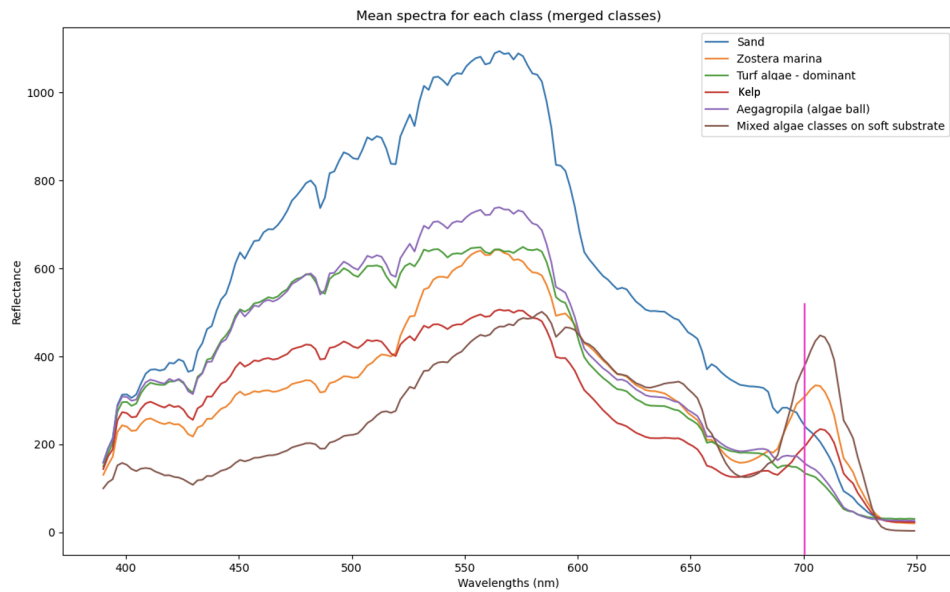




**Figure 6.14:** Illustration of how the different classes appear under water. (a) - (d) and (f) - (g) are captured by the MASSIMAL project using an underwater camera. (e) and (h) are re-used from Store Norske Leksikon [75, 76]. Turf algae is shown in (c) and (d) as the blurry parts.



(a) Mean spectral response for all classes in the dataset.

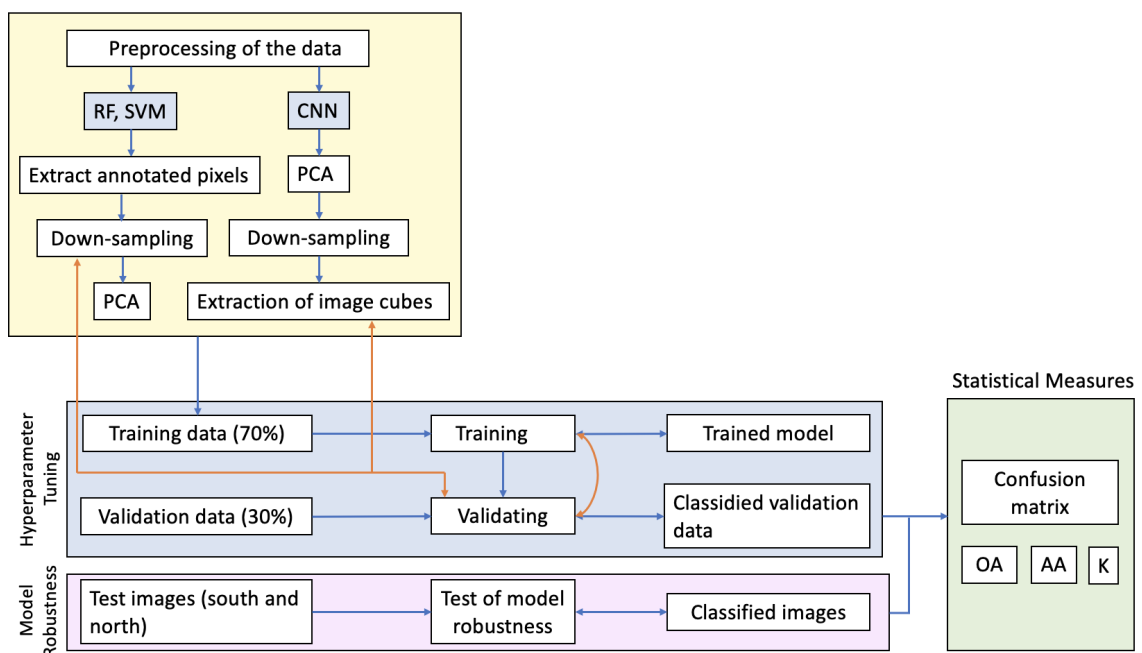


(b) Mean spectral response for the merged classes.

**Figure 6.15:** Mean spectral response for the dataset with all individual classes in (a) and with the merged classes (b). The pink, vertical line marks the red edge, which is clearly visible for some of the classes.

## 6.3 Methodology

This work studied three supervised ML methods applied to hyperspectral drone data with the goal of predicting different classes of underwater vegetation. A flowchart explaining the experimental setup can be seen in Fig. 6.16. The figure shows that whole process was the same for RF and SVM while the process for CNN was slightly different. For RF and SVM, the whole process can be described by; extraction of the annotated pixels from all images, under sampling, PCA, training, validating and testing of model robustness. For CNN the process was as a bit different; first PCA, then under-sampling followed by extraction of image cubes, training, validating and testing of model robustness. Confusion matrix, Overall Accuracy (OA), Average Accuracy (AA) and the Kappa statistics were computed. The procedures were done for all classes and for the merged classes. The implementations was performed in Python and will be open access in the future [64, 77–81].



**Figure 6.16:** Flowchart that shows the process from receiving the data from MASSIMAL to the classified images. Note, that different methods for under-sampling the dataset for RF and SVM were tested after the validation step, and different methods of extraction of image cubes for CNN were tested.

### 6.3.1 Preprocessing

#### Extraction of annotated pixels

For RF and SVM, all annotated pixels from all images (from the South side) were collected in an array, with associated class labels. From this array, under-sampling was performed (see *Under-sampling*).

For CNN, extraction of annotated pixels was done image by image, where for each image, a subset of annotated pixels were randomly picked (see *Under-sampling* for further explanation of this process).

#### Under-sampling

The method used for under-sampling was the `RandomUnderSample` from the `Imblearn` library in Python [82] for RF and SVM. This method randomly picks samples from majority classes with replacement. Different under-sampling methods were tested such as the `NearMiss` and `EditedNearestNeighbours`, but the `RandomUnderSample` method was found to be the most suitable for this case. From the array in which the annotated pixels were stored, under-sampling was performed. After under-sampling each class had 20 600 samples. These samples were stored in an array, which the train/validating split was performed on.

For CNN, under-sampling was done randomly while extracting annotated pixels from the images. For each class 8000 samples were picked randomly with replacement. Much less samples were extracted for CNN than for RF and SVM due to practical reasons such as storage and computational power. The under-sampling was done with replacement, because the collection of samples were done image per image, and not all images had enough samples of each class to fill their quota. The cubes were stored in an array, which the train/validating split was performed on.

#### Train/validation split

For all classifiers, the dataset was split into a training part and a validation part after under-sampling. 70 % of the under-sampled dataset was assigned to the training part and 30 % was assigned to the validation part. The dataset was randomly split in such a way that the train and validation parts have a balanced amount of data points for each class. In this case it means that for every class, 70 % of the data points are assigned to train and 30 % is assigned to the validation part. Note: This way of splitting the data does not guarantee



an independent dataset for the validation. The reason for this is that some data points might originate from right next to each other in one of the images. If one of these data points are used in training, and the other one in validation, this might cause a "leak" of information from the training process to the validation process. This is especially true for CNN, where two data points close to each other in an image will have an overlaying neighbourhood.

### **Dimensionality reduction**

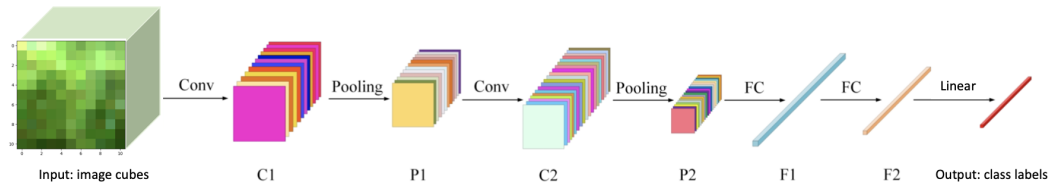
For RF and SVM, PCA was performed after under-sampling. For CNN, PCA was performed before under-sampling due to practical reasons such as data storage and the big amount of computational power which would be needed in order to handle data cubes of  $11 \times 11 \times 173$ . Data cubes of  $11 \times 11 \times 25$  were much more manageable.

Before applying PCA the dataset was standardized which is a requirement for applying PCA (see Sec. 6.1.2). 25 principal components with an explained variance of 95.37 % was used for PCA. Figure 6.19 shows a plot of the first 100 principal components. By inspecting this figure it shows that after approximately 20-30 principal components, not much more information is gained by including more components. 25 components also showed to be sufficient in order to obtain a satisfying accuracy, while keeping down the complexity of the data.

### **6.3.2 Hyperparameter tuning**

Hyperparameter tuning was performed for all methods implemented here. For RF and SVM a grid search of different parameters were performed. For RF a range of values for the number of trees (`n_estimators`) and max depth of trees (`max_depth`) were tested [79]. For SVM different kernels were tested, such as the linear, polynomial and RBF, together with different values for the slack parameter. Since the best kernel for this problem was the RBF kernel, the  $\gamma$  parameter was also tested for a range of values.

For CNN, manual testing of hyperparameters was done, testing different values and combinations of learning rate, epochs, batch size, activation functions, optimiser, loss functions, kernel sizes, the number of nodes in each layer. Both Maxpool with kernel  $2 \times 2$  and kernel  $3 \times 3$  was tested. Different values for the filter size and kernel size for the convolutional layers were also tested as well as different units for the first fully connected layer.



**Figure 6.17:** The architecture of the CNN built in this project. The network consists of two convolutional layers (C1 and C2), two maxpool layers (P1 and P2) and two fully connected layers (F1 and F2). The input is image cubes and the output is predicted class labels. Figure reused from [61] with some small changes.

### 6.3.3 Convolutional Neural Network

The CNN used in this project was built as a sequential model with three different types of layers; convolutional layers, maxpool layers and fully connected layers.

The overall architecture of the CNN built in this project is shown in Fig. 6.17. The figure shows how features were extracted from an input image cube and classified in the last layer of the network. The input image shows an example of how a neighbourhood of class 2 (*Zostera marina* - dominant) pixels appears (shown in RGB representation). The model consisted of two convolutional layers, two maxpool layers and two fully connected layers. The input of the model was image cubes of size  $11 \times 11 \times 25$ , where 25 is the number of principal components generated from the PCA.

The first layer was a convolutional layer with a kernel size of  $3 \times 3$  and 64 filters. This produced an output of  $(9 \times 9 \times 64)$ , which was fed into a maxpool layer. The pool size was  $2 \times 2$ , and thus the output shape of the first maxpool layer was  $(4 \times 4 \times 64)$ . The next layer was a convolutional layer with kernel size  $3 \times 3$  and 64 filters. The output of this layer was  $(2 \times 2 \times 64)$ , which was then fed into a new maxpool layer with pool size  $2 \times 2$ . The output of the pooling layer had a shape of  $(1 \times 1 \times 64)$ . This was flattened into a vector of shape (64), which was then fed into the first fully connected layer. The first fully connected layer had 128 nodes. The output shape of the first fully connected layer was (128). The last layer of the network was where the classification happened, thus this layer had the same amount of nodes as the number of classes in the dataset; 9 or 6. The output of the last fully connected layer was a (9), or (6) vector. The vector was filled with values where the index of the maximum value indicated the predicted class. If the maximum was located at index 3, then it was class 3 (*Zostera marina* with turf algae) that was predicted. Both convolutional layers and the first fully connected layer used ReLu as activation function while the

last fully connected layer used Linear as activation function.

**Table 6.2:** The architecture and layer-specific hyperparameters of the CNN built in this project. The number of classes was 9 and 6, for all individual classes and the merged classes, respectively.

Layer (type)	Hyperparameters	Output shape
Conv2D	filters = 64, kernel = 3x3, padding = "valid", activation = "relu"	( 9, 9, 64)
MaxPool 2D	pool_size = 2x2, strides=None, padding = "valid"	( 4, 4, 64)
Conv2D	filters = 64, kernel = 3x3, padding = "valid", activation = "relu"	(2, 2, 64)
MaxPool2D	pool_size = 2x2, strides=None, padding = "valid"	(1, 1, 64)
Flatten		(64)
Dense	units = 128, activation = "relu"	(128)
Dense	units = #number of classes, activation = "linear"	(9) or (6)

Table 6.2 specifies the hyperparameters that were used for the different layers. The hyperparameters were the same for the all class case and the merged class case, only the number of units in the last fully connected layer differed. A range of different values for the hyperparameters was tested, where the ones shown in Tab. 6.2 are those that resulted in the best model. Different values for learning rate, batch size, epochs were also tested. A learning rate of 0.001, a batch size of 128 and 100 epochs resulted in the most suited model. Different types of optimization and loss functions were tested, where the Adam optimizer (a type of stochastic gradient descent optimizer) and the Sparse Categorical Crossentropy (loss function) were the ones that resulted in the best model.

### 6.3.4 Random Forest

For the RF classifier, two hyperparameters turned out to affect the performance of the model considerably. These were the number of trees (`n_estimators`) and max depth of trees (`max_depth`). The number of trees refers to the number of decision trees that is built before the majority voting. A high number of trees can make the performance better but slow down the computational time. The depth of the trees controls how complex each decision tree can be. A high number for the depth can cause overfitting because each tree learns the pattern of the training data very well. For `n_estimators` 800 was used and for `max_depth` 35 was used. These were the values that resulted in a model which has a good performance on both training and validation data.

### 6.3.5 Support Vector Machine

For the SVM, the hyperparameters that were tuned in this project were the slack parameter (see Sec. 6.1.2),  $C$ , and the gamma parameter,  $\gamma$ . The gamma parameter in the RBF kernel influences how wide the kernel is. A large value for gamma means a bigger decision region [64]. The influence of  $C$  is described in Sec. 6.1.2. The RBF kernel,  $K$ , is given by

$$K(\mathbf{x}, \mathbf{x}') = \exp(\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \quad (6.10)$$

where  $\mathbf{x}$  and  $\mathbf{x}'$  are two feature vectors in the dataset [64, 83].

For  $C$ , a value for 1 proved to be the best in combination with specifying  $\gamma$  as "scale". By specifying  $\gamma$  as "scale",  $\gamma$  is given by

$$\gamma = 1/(N_s \cdot \sigma_x^2), \quad (6.11)$$

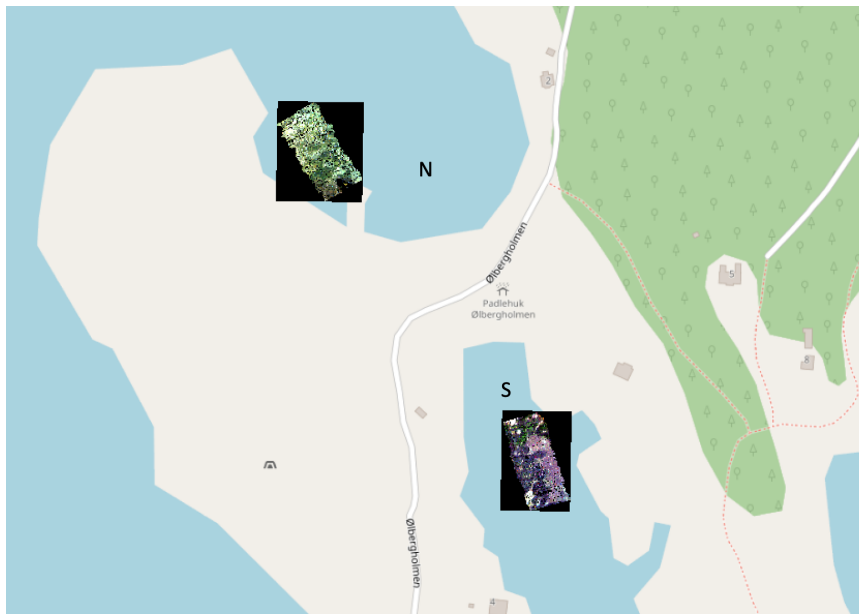
where  $N_s$  is the number of features in the dataset, which in this case is 25 after PCA is performed.  $\sigma_x^2$  is the variance of the dataset (the training dataset) [64, 83].

### 6.3.6 Statistical measures

For this project, three statistical measures was calculated; the Overall Accuracy (OA), Eq. (6.1), the Average Accuracy, (AA) Eq. (6.2) and the Kappa statistic,  $K$ , Eq. (6.3).

### 6.3.7 Algorithm application to images

After the establishment of the methods, they were applied to one image from the South side for visual validation, as well as one image from the North side for robustness evaluation. Figure 6.18 shows an overview of where these images were positioned. The image from the South side was included in the training data, while the image from the North side was not and is therefore an independent dataset that can be used as a robustness evaluation. In addition, the established models were applied in prediction mode on an additional image from the South side and compared to the RGB image and annotation mask for evaluation. This was not the focus of the thesis, but included for evaluation of how the methods are capable of predicting the spatial patterns of the vegetation. These results are presented i the Appendix, Ch. 7, in Fig. 7.2

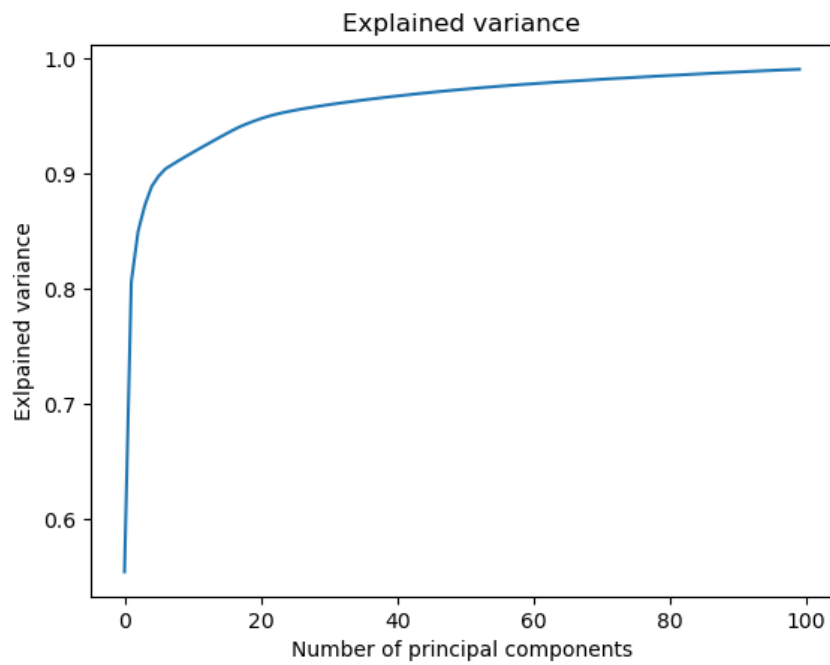


**Figure 6.18:** The image from the South side (S) is used for visual presentation of how the methods perform. The image from the North side (N) is used for robustness evaluation.

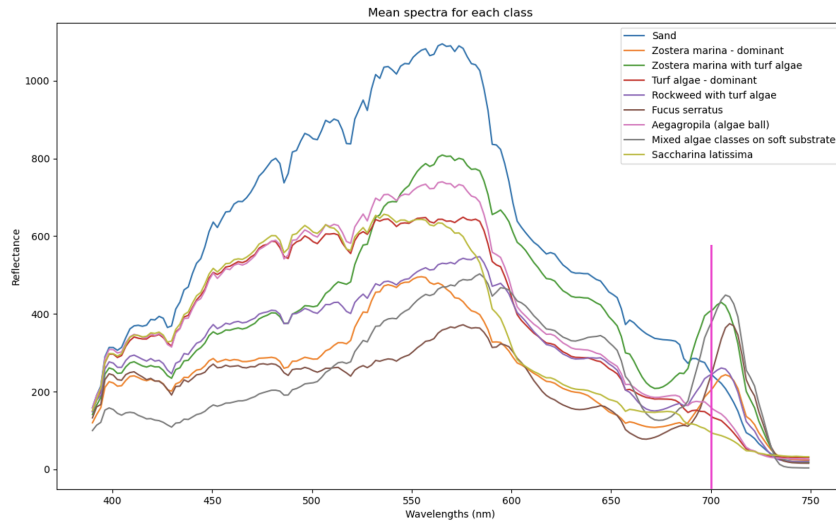
## 6.4 Results

### 6.4.1 Dimensionality reduction

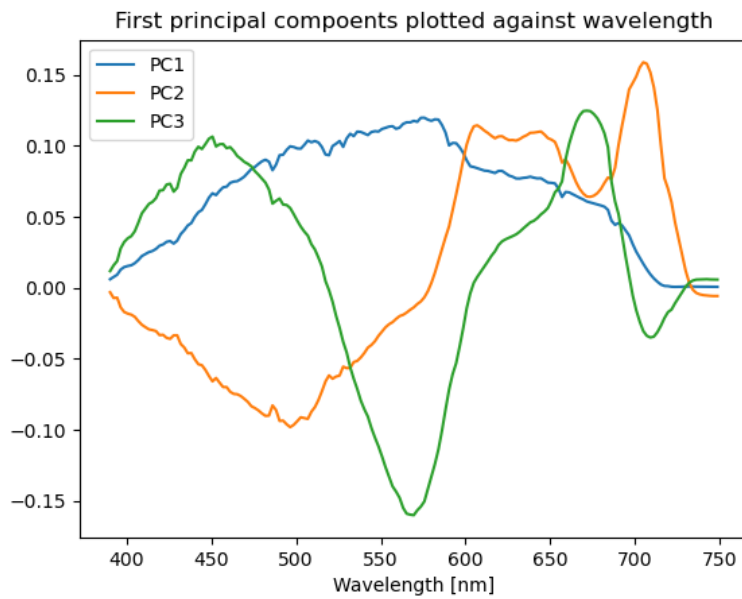
Dimensionality reduction, PCA, was performed on the dataset before applying the ML models. Figure 6.19 shows a plot of the first 100 principal components vs. total explained variance. For this project, the 25 first principal components were used, and by inspection of Fig. 6.19, it can be seen that after 25 principal components, not much extra information is retrieved by keeping more components. Figure 6.20 shows both the mean spectra for all individual classes and the first three principal components plotted against wavelength. By comparing the mean spectra in Fig. 6.20 (a) and the plot of principal components in Fig. 6.20 (b), shows that the first principal components are able to pick up on the pattern in the data. PC1 (the blue line in Fig. 6.20 (b)) represent the overall, common pattern in the reflectance spectrum. PC2 and PC3 picks up on the increase from ca. 400 nm - 500 nm and the decrease from ca. 550 nm to 700 nm. All principal components picks up on the red edge at ca. 700 nm, where there is a sudden shift in the components' magnitude.



**Figure 6.19:** Explained variance for PCA applied to the dataset. In this project 25 principal components are used with an explained variance of 95.37%.



(a) Mean spectral for each class in the dataset.



(b) The first three principal components plotted against wavelength.

**Figure 6.20:** Mean spectra for each of the individual classes in (a) and the three first principal components plotted against wavelength in (b).

**Table 6.3:** The statistical measures that the models were evaluated on. OA = Overall Accuracy, AA = Average Accuracy, K = the Kappa statistic.

Statistical measures			
All classes, Validation			
	<b>CNN</b>	<b>RF</b>	<b>SVM</b>
<b>OA</b>	0,961658	0,926941	0,930581
<b>AA</b>	0,961042	0,926707	0,930305
<b>K</b>	0,95685	0,917805	0,921901
Merged classes, Validation			
	<b>CNN</b>	<b>RF</b>	<b>SVM</b>
<b>OA</b>	0,993193	0,94781	0,954246
<b>AA</b>	0,992134	0,948247	0,957467
<b>K</b>	0,991345	0,933962	0,94315
All classes, image from the South			
	<b>CNN</b>	<b>RF</b>	<b>SVM</b>
<b>OA</b>	0,930383	0,83824	0,840607
<b>AA</b>	0,943694	0,623008	0,534791
<b>K</b>	0,834615	0,533298	0,667969
Merged classes, image from the South			
	<b>CNN</b>	<b>RF</b>	<b>SVM</b>
<b>OA</b>	0,991193	0,944502	0,951548
<b>AA</b>	0,963473	0,750883	0,650362
<b>K</b>	0,956627	0,753747	0,777111
All classes, image from the North			
	<b>CNN</b>	<b>RF</b>	<b>SVM</b>
<b>OA</b>	0,521262	0,709405	0,696409
<b>AA</b>	0,150715	0,427965	0,390961
<b>K</b>	0,053965	0,242583	0,232356
Merged classes, image from the North			
	<b>CNN</b>	<b>RF</b>	<b>SVM</b>
<b>OA</b>	0,541697	0,807127	0,801938
<b>AA</b>	0,297593	0,502103	0,445406
<b>K</b>	0,112597	0,378217	0,362543

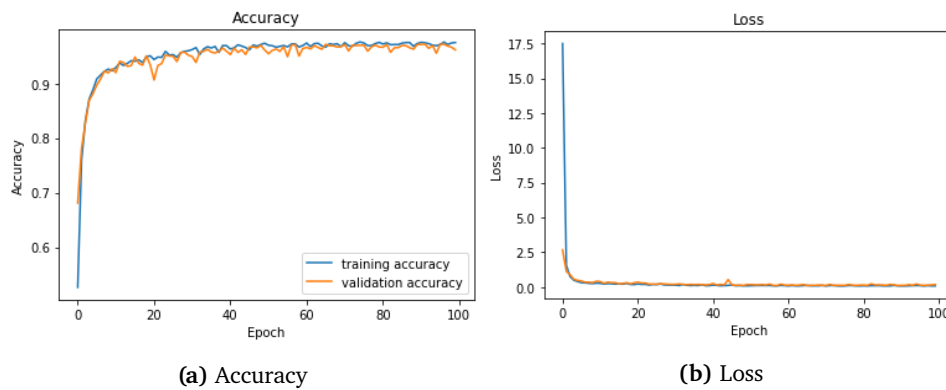
## 6.4.2 Classification results

All methods were evaluated based on three statistical measurements; the OA, AA and Kappa statistic as well as confusion matrices. OA, AA and the Kappa statistics are all defined in Sec. 6.1.1 in Eq. (6.1), (6.2), (6.3), respectively. Table 6.3 presents the statistics for the models. By inspecting the table, RF and SVM appears to have similar performance across all cases. RF and SVM outperforms CNN when applied to the image from the North. However, CNN performs better than both RF and SVM in all other cases presented in the table. Comparing the tables for all classes and merged classes, all classifiers perform better for the merged classes case.

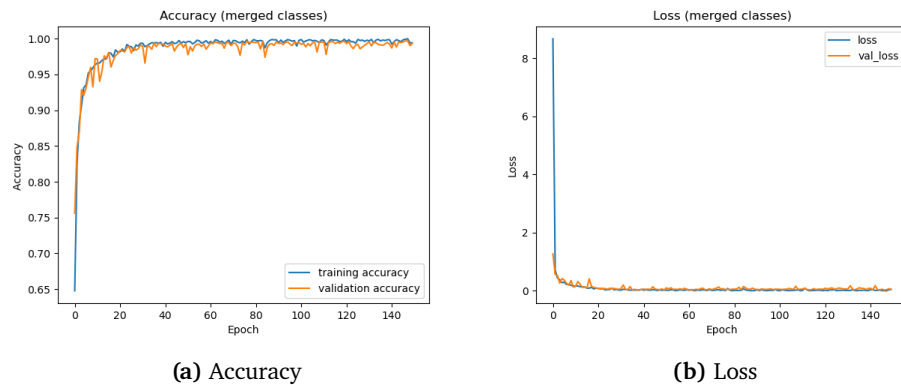
### Accuracy evaluation of CNN

The evaluation of CNN was done by inspecting the accuracy and loss plots, as well as the OA, AA, and Kappa statistic. Figure 6.21 shows the accuracy and loss plot for all individual classes (a) and for merged classes (b). Figure 6.21 (a) and (b) shows that after ca. 60 - 70 epochs the models accuracy and loss stabilizes. Figure 6.22 (a) and (b) shows that the model stabilizes also here after 60 - 70 epochs. However, the model does have some oscillation on the accuracy plot in Fig. 6.22 (a) even after 70 epochs. By inspecting the plot, it appears that the





**Figure 6.21:** Accuracy and loss for CNN for all classes, plotted for 100 epochs



**Figure 6.22:** Accuracy and loss for CNN for merged classes, plotted for 150 epochs. Note: the network was trained for 100 epochs for the merged class case as well. 150 epoch is presented here to show that the accuracy converges.

model is somewhat stable after about 100 epochs. The CNN was trained for 100 epochs for both the individual classes and the merged classes.

### Confusion matrices

The confusion matrices for all models and all cases are presented in Tab. 6.4 - 6.9.

For the case with all classes, Tab. 6.4, 6.5 and 6.6 present the confusion matrices for validation, image from the South and image from the North, respectively.

Inspecting the confusion matrices in Tab. 6.4, it can be seen that all models have a high degree of correct classification, with few FP and FN relative to the amount of TP, in accordance with the statistics reported in Tab. 6.3. In

case of CNN the only class that showed lower classification accuracy was class 5 (Rockweed with turf algae). For RF class 4 (Turf algae) and class 9 (Saccharina latissima) and for SVM, class 3 (Zostera Marina with turf algae) showed slightly lower performance. It can be noticed that most of these classes include turf algae, which might influence the performance of the classifications. Class 7 (Aegagropila (algae ball)) and class 8 (Mixed algae classes) have a high percentage of TP, and are involved in few FP for the other classes.

The confusion matrices in Tab. 6.5 (image from the South) show still a high amount of TP for all classifiers. This matches the reported statistics in Tab. 6.3. Class 3 (Zostera Marina with turf algae) and class 4 (Turf algae) stands out as the class where both RF and SVM perform significantly worse than CNN. For class 7 ((Aegagropila (algae ball))), CNN and RF have high performance with 92 % and 88 % TP, respectively, while SVM have a slightly lower performance of 78 % TP. For the classes that are not present in the image, all classifiers predict FP, although with low percentages.

For the image from the North, which was used independently for robustness evaluation, Tab. 6.6 shows the confusion matrices for all classifiers. The table shows that all classifiers are relatively good at classifying class 3 (Zostera Marina with turf algae) correct. For class 4 (Turf algae), and 5 (Rockweed with turf algae), all classifiers perform poorly, with few TP and many FP and FN. For class 1 (Sand), both RF and SVM perform rather good, while CNN have a small amount of TP, which is likely due to the different architecture of the CNN. For class 4 (Turf algae) non of the models are able to predict with high accuracy, all scoring less than 1 % correctly classified data points for class 4 (Turf algae). The same is valid for class 5 (Rockweed with turf algae), where all classifiers have less than 6 % of correctly classified targets. This shows that the presence of turf algae impacts the performance of classification. For the classes that are not present in the image, all classifiers predict FP with low percentage, except for class 7 ((Aegagropila (algae ball))), which has over 50 % FP.

For the merged classes, Tab. 6.7, 6.8 and 6.9 present the confusion matrices for validation, image from the South and image from the North, respectively.

Table 6.7 presents the confusion matrices for validation. The table shows that all models have a high performance with relatively few amount of FP and FN. CNN has the highest amount of correctly classified targets. This reflects the results presented in Tab. 6.3 where CNN have the highest values for OA, AA and K. This confirms the hypothesis that the classifiers would perform better with some of the classes merged.

Furthermore, Tab. 6.8 shows the confusion matrices for the image from the South. The table shows that CNN has the overall best performance with the

most amount of TP. The same pattern appears here as with the all class case in Tab. 6.5 where class 4 (Turf algae), causes challenges for RF and SVM, while CNN has a good performance on predicting TP for class 4 (Turf algae). For the classes that are not present in the image, all classifiers predict few FP.

For the image from the North, Tab. 6.9 shows the belonging confusion matrices. The same patterns appears in this case as well as in Tab. 6.6 (all classes, image from the North). However, CNN has low performance on class 1 (Sand), while RF and SVM have a relatively good performance. Note, CNN uses an 11x11 window to capture the neighbourhood pixels. In the image from the North, there are relatively few pixels that are annotated with class 1 (Sand), which causes some of the neighbourhoods of sand pixels to be smaller in prevalence than the 11x11 window used for CNN. This might cause the low accuracy.

For class 4 (Turf algae), non of the models are able to predict any TP. For class 10 (Zostera marina), the merged class from class 2 (Zostera Marina - dominant) and 3 (Zostera Marina - dominant and Zostera Marina with turf algae, respectively), Rf and SVM have a good performance with over 97 % TP, while CNN has 59 % TP. For class 11 (Kelp), the merged class of 5 (Rockweed with turf algae), 6 (Fucus serratus) and 9 (Saccharina latissima) all models have a relatively low amount of TP; CNN has 50 %, RF has 15 % and SVM has 13 %. This is in correspondence with the results presented in Tab. 6.6 (all classes, image from the North side), where both RF and SVM showed low performance in the prediction of class 5, while slightly better in predicting class 6 (Fucus serratus) (36 % and 29 % TP for RF and SVM, respectively). CNN did not perform good at either class 5 (Rockweed with turf algae) or 6 (Fucus serratus), with low percentage of TP for both class 5 and class 6 (Fucus serratus, 0.4 % and 0.8 % respectively). For the classes that are not present in the image, all classifiers predict FP.

It can be summarized that all models perform good in validation. For the image from the South, all classifiers have a relatively good performance for all classes. The only class that is causing challenges is class 4 (Turf algae), where RF and SVM have difficulties classifying it correctly, for both the all class case and the merged class case. Then for the image from the North, CNN shows low performance for class 1 (Sand), while RF and SVM have a good performance for class 1 (Sand). This might be due to the differences between the architecture of the methods. For class 4 (Turf algae), non of the classifiers could consistently identify it correctly, for both the all class case and the merged class case. Then for class 11 (Kelp), RF and SVM show low performance, while CNN has 50 % TP. Also, for the classes that are not present in the image from the South and the image from the North, all classifiers are predicting FP, although the total percentage of these FP are low.

**Table 6.4:** Confusion matrices for the validation of the models for all classes. The yellow indicates TP.

All classes, validation									
CNN									
Predicted									
Actual									
	1	2	3	4	5	6	7	8	9
1	0,9856	0,0014	0,0056	0,0005	0,0009	0	0,0056	0	0,0005
2	0,0021	0,9448	115	0	0,0017	0,0004	0,0013	0,0008	0
3	0,0038	0,0168	0,968	0	0,0021	0,0029	0,0025	0,0017	0,0021
4	0,0012	0,0025	0,0162	0,9771	0,0008	0	0,0021	0	0
5	0,0004	0	0,0021	0	0,7996	0,1962	0,0013	0,0004	0
6	0	0	0,0042	0	0	0,9958	0	0	0
7	0	0	0,0049	0	0,0085	0	0,9838	0,0024	0,0004
8	0	0,0004	0	0	0	0	2	0,9988	0
9	0,0011	0	0,003	0	0	0	0	0	0,9959

RF									
Predicted									
Actual									
	1	2	3	4	5	6	7	8	9
1	0,9174	0	0,0008	0,0219	0,0296	0,0045	0,0037	0,0101	0,012
2	0,0013	0,9604	0,0022	0,0325	0,0003	0,0002	0,0005	0,0018	0,0008
3	0	0,0024	0,9767	0,0118	0,0003	0,0037	0	0,0021	0,0029
4	0,0544	0,0406	0,0602	0,7956	0,0015	0,0056	0,0205	0,0055	0,0161
5	0,0272	0,0003	0	0,0008	0,9644	0,0006	0,0028	0,0036	0,0003
6	0,0035	0	0,0018	0,001	0,0008	0,9896	0,0013	0	0,0021
7	0,0357	0,0002	0,0003	0,0168	0,0031	0,0027	0,9378	0,0011	0,0023
8	0,0025	0,0017	0,0035	0,0176	0,0017	0,0005	0,0014	0,951	0,02
9	0,0125	0,0018	0,0048	0,0233	0,0011	0,0059	0,0023	0,1008	0,8475

SVM									
Predicted									
Actual									
	1	2	3	4	5	6	7	8	9
1	0,9854	0	0,0024	0	0,0039	0	0,0005	0,0078	0
2	0	0,9344	0,0628	0	0,001	0,0005	0,0002	0,0013	0
3	0,0015	0,1264	0,8521	0	0,0175	0,0003	0	0,0023	0
4	0	0,0008	0,0002	0,9425	0,004	0	0,0526	0	0
5	0,0179	0,003	0,0076	0,0013	0,8253	0,0539	0,0299	0,0611	0
6	0,0002	0,0003	0,0003	0	0,0323	0,9653	0	0,0016	0
7	0	0,0023	0,0003	0,0068	0,0118	0	0,9556	0	0,0232
8	0,0083	0,0016	0,0029	0	0,0437	0,0033	0,0007	0,9395	0
9	0	0,0003	0	0,003	0	0	0,0235	0,0005	0,9727

- |   |                                |   |                          |
|---|--------------------------------|---|--------------------------|
| 1 | Sand                           | 6 | Fucus serratus           |
| 2 | Zostera marina - dominant      | 7 | Aegagropila (algae ball) |
| 3 | Zostera marina with turf algae | 8 | Mixed algae classes      |
| 4 | Turf algae - dominant          | 9 | Saccharina latissima     |
| 5 | Rockweed with turf algae       |   |                          |

**Table 6.5:** Confusion matrices for the image from the South for all classes. The yellow indicates TP.

All classes, image from the South										
CNN										
Predicted										
	1	2	3	4	5	6	7	8	9	
Actual	1	0,9615	0,00133	0,01775	0,00732	0,001	0	0,01054	0	0,00055
	2	0,00005	0,93058	0,06548	0,00011	0,00084	1,2E-05	0,00278	0	0,00012
	3	0,00674	0,04258	0,90212	0,00107	0,02367	0,0023	0,01685	0	0,00467
	4	0,0017	0	0,0004	0,99744	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0
	7	0,01626	0	0,01452	0,03775	0,00058	0	0,92683	0	0
	8	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0

RF										
Predicted										
	1	2	3	4	5	6	7	8	9	
Actual	1	0,7986	0,003	0,00522	0,00555	0,00788	0,00011	0,17843	0,00067	0,00055
	2	0,00034	0,96831	0,01035	0,00183	0,01001	0,00202	0,00481	0,00481	0,00123
	3	0,01141	0,78762	0,08003	0,00536	0,03852	0,0036	0,06778	0,00038	0,00528
	4	0,01107	0,00341	0,03918	0,38075	0,01704	0,0017	0,546	0,00085	0
	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0
	7	0,01568	0,04355	0,02962	0,00232	0,01974	0	0,88734	0,00174	0
	8	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0

SVM										
Predicted										
	1	2	3	4	5	6	7	8	9	
Actual	1	0,83577	0,00055	0,00166	0,003	0,00055	0	0,1579	0	0,00055
	2	0,00013	0,98215	0,0086	0,00085	0,00147	0,00036	0,00539	0,00047	0,00057
	3	0,01425	0,85632	0,03048	0,00237	0,00735	0,00031	0,08616	0,00023	0,00253
	4	0,00426	0,01704	0,00681	0,03578	0,00596	0	0,9293	0	0,00085
	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0
	7	0,04413	0,0993	0,04588	0,00929	0,00987	0	0,78978	0	0,00174
	8	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0

Class not present in the image									
Class present in the image									

1 Sand	6 Fucus serratus
2 Zostera marina - dominant	7 Aegagropila (algae ball)
3 Zostera marina with turf algae	8 Mixed algae classes
4 Turf algae - dominant	9 Saccharina latissima
5 Rockweed with turf algae	

**Table 6.6:** Confusion matrices for the image from the North for all classes. The yellow indicates TP.

All classes, image from the North										
CNN										
Predicted										
	1	2	3	4	5	6	7	8	9	
Actual	1	0,06522	0,06751	0,67735	0	0	0	0,18993	0	0
	2	0	0	0	0	0	0	0	0	
	3	0,01893	0,00282	0,67543	0,06679	0,0148	0,04559	0,0771	0,00246	0,09609
	4	0	0,14564	0,44213	0	0,11313	0,0104	0,25358	0,03511	0
	5	0,11963	0,01333	0,56667	0	0,00481	0,01074	0,06185	0	0,22296
	6	0	0	0,00946	0	0,9027	0,00811	0,01351	0,06622	0
	7	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0

RF										
Predicted										
	1	2	3	4	5	6	7	8	9	
Actual	1	0,88215	0	0,08467	0	0	0	0,03318	0	
	2	0	0	0	0	0	0	0	0	
	3	0,0024	0,11538	0,85731	0,00066	0,00701	0,00174	0,00126	0,0136	0,00066
	4	0	0	0,60598	0,0013	0,28869	0,0013	0,0052	0,09753	0
	5	0,01259	0,00074	0,92111	0	0,03556	0	0,00259	0,02741	0
	6	0	0,00541	0,04054	0	0,0973	0,36351	0,00135	0,49054	0,00135
	7	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0

SVM										
Predicted										
	1	2	3	4	5	6	7	8	9	
Actual	1	0,7643	0	0,02517	0	0	0	0,00229	0,20824	
	2	0	0	0	0	0	0	0	0	
	3	0,0009	0,13664	0,847	0	0,00425	0,00102	0,0006	0,00551	0,00407
	4	0	0	0,54356	0	0,38101	0	0	0,07542	0
	5	0,0163	0,00037	0,90296	0	0,05296	0,00111	0,00185	0,00815	0,0163
	6	0	0,00541	0,01216	0	0,13243	0,29054	0	0,55946	0
	7	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0

Class not present in the image									
Class present in the image									
1	Sand	6	Fucus serratus						
2	Zostera marina - dominant	7	Aegagropila (algae ball)						
3	Zostera marina with turf algae	8	Mixed algae classes						
4	Turf algae - dominant	9	Saccharina						
5	Rockweed with turf algae								

**Table 6.7:** Confusion matrices for the validation of all models, merged classes. The yellow indicates TP.

Merged classes, validation							
CNN							
Predicted							
Actual							
	1	10	4	11	7	8	
1	0,98467	0,01347	0,00046	0,00046	0,00093	0	
10	0	0,9943	0	0,0038	0,00148	0,00042	
4	0	0,00208	0,99792	0	0	0	
11	0,00041	0,00393	0	0,9954	0,00027	0	
7	0,00041	0,01704	0	0	0,98256	0	
8	0	0,00122	0	0,00041	0,00041	0,99796	

RF							
Predicted							
Actual							
	1	10	4	11	7	8	
1	0,90307	0,00016	0,05926	0,00373	0,00325	0,03052	
10	0	0,96815	0,0215	0,00372	0	0,00663	
4	0,02429	0,018	0,93966	0,0022	0,00645	0,0094	
11	0,00321	0,00161	0,00257	0,98828	0,00144	0,00289	
7	0,03262	0	0,03133	0,00242	0,92927	0,00436	
8	0,0054	0,00218	0,02774	0,00258	0,00105	0,96105	

SVM							
Predicted							
Actual							
	1	10	4	11	7	8	
1	0,91898	0	0,0552	0,00438	0,00406	0,01737	
10	0	0,97235	0,02183	0,00081	0	0,00501	
4	0,0296	0,01757	0,93236	0,00231	0,01139	0,00677	
11	0,00369	0,00161	0,00096	0,99149	0,00016	0,00209	
7	0,03149	0	0,01421	0,00161	0,95124	0,00145	
8	0,00524	0,00177	0,01081	0,0021	0,00169	0,97839	

- 1 Sand
- 10 Zostera Marina (merged from class 2 (Zostera Marina - dominant) and class 3 (Zostera Marina with turf algae))
- 4 Turf algae - dominant
- 11 Kelp (merged from class 5 (Rockweed with turf algae), class 6 (Fucus serratus) and class 9 (Saccharina latissima))
- 7 Aegagropila (algae ball)
- 8 Mixed algae classes

**Table 6.8:** Confusion matrices for the image from the South for merged classes. The yellow indicates TP.

Merged classes, image from the South							
CNN							
		Predicted					
		1	10	4	11	7	8
Actual	1	0,96682	0,02774	0,00067	0	0,00477	0
	10	0,0007	0,99501	9,4E-05	0,00263	0,00152	4,2E-05
	4	0	0,02896	0,97104	0	0	0
	11	0	0	0	0	0	0
	7	0,0122	0,0633	0,00348	0	0,92102	0
	8	0	0	0	0	0	0

RF							
		Predicted					
		1	10	4	11	7	8
Actual	1	0,79572	0,02419	0,00333	0,01076	0,16534	0,00067
	10	0,00171	0,96702	0,00094	0,02086	0,009	0,00046
	4	0,00681	0,05451	0,38075	0,02129	0,53578	0,00085
	11	0	0	0	0	0	0
	7	0,01336	0,10569	0,00174	0,01916	0,86005	0
	8	0	0	0	0	0	0

SVM							
		Predicted					
		1	10	4	11	7	8
Actual	1	0,83034	0,00311	0,00266	0,00211	0,16178	0
	10	0,00211	0,97777	0,00065	0,00568	0,01337	0,00043
	4	0,00852	0,02385	0,03492	0,01107	0,92078	0,00085
	11	0	0	0	0	0	0
	7	0,04413	0,17538	0,00813	0,01394	0,75842	0
	8	0	0	0	0	0	0

Class not present in the image

Class present in the image

- 1 Sand
- 10 Zostera Marina (merged from class 2 (Zostera Marina - dominant) and class 3 (Zostera Marina with turf algae))
- 4 Turf algae - dominant
- 11 Kelp (merged from class 5 (Rockweed with turf algae), class 6 (Fucus serratus) and class 9 (Saccharina latissima))
- 7 Aegagropila (algae ball)
- 8 Mixed algae classes



**Table 6.9:** Confusion matrices for the image from the North for merged classes. The yellow indicates TP.

Merged classes, image from the North

**CNN**

		Predicted					
		1	10	4	11	7	8
Actual	1	0,08352	0,90961	0	0,00686	0	0
	10	0,04882	0,59726	0,01018	0,19427	0,1491	0,00036
	4	0	0,79974	0	0,0039	0,19636	0
	11	0,05698	0,36628	0	0,50959	0,06628	0,00087
	7	0	0	0	0	0	0
	8	0	0	0	0	0	0

**RF**

		Predicted					
		1	10	4	11	7	8
Actual	1	0,88215	0,08581	0	0,00229	0	0,02975
	10	0,00228	0,97568	0	0,0139	0,00036	0,00779
	4	0	0,59948	0	0,3173	0,0013	0,08192
	11	0,00872	0,73343	0	0,15058	0,00116	0,1061
	7	0	0	0	0	0	0
	8	0	0	0	0	0	0

**SVM**

		Predicted					
		1	10	4	11	7	8
Actual	1	0,66362	0,02746	0	0,30892	0	0
	10	0,00078	0,98371	0	0,01162	0,00048	0,00341
	4	0	0,52796	0	0,40182	0	0,07022
	11	0,01134	0,70988	0	0,1343	0,00291	0,14157
	7	0	0	0	0	0	0
	8	0	0	0	0	0	0

- Class not present in the image
- Class present in the image
- 10 Zostera Marina (merged from class 2 (Zostera Marina - dominant))
- 4 Turf algae
- 11 Kelp (merged from class 5 (Rockweed with turf algae), class 6 (Fucus serratus) and class 9 (Saccharina latissima))
- 7 Aegagropila (algae ball)
- 8 Mixed algae classes

### 6.4.3 Algorithms applied to images for visual presentation

The ML methods trained in this project were applied to two images for visualization of the results. Testing of the model robustness was achieved by applying the models to an independent image, namely the image from the North. Figure 6.23 and 6.24 shows the results for the image from the South and North, for all classes, respectively. Figure 6.25 and 6.26 show the results for the images from the South and North, for merged classes, respectively.

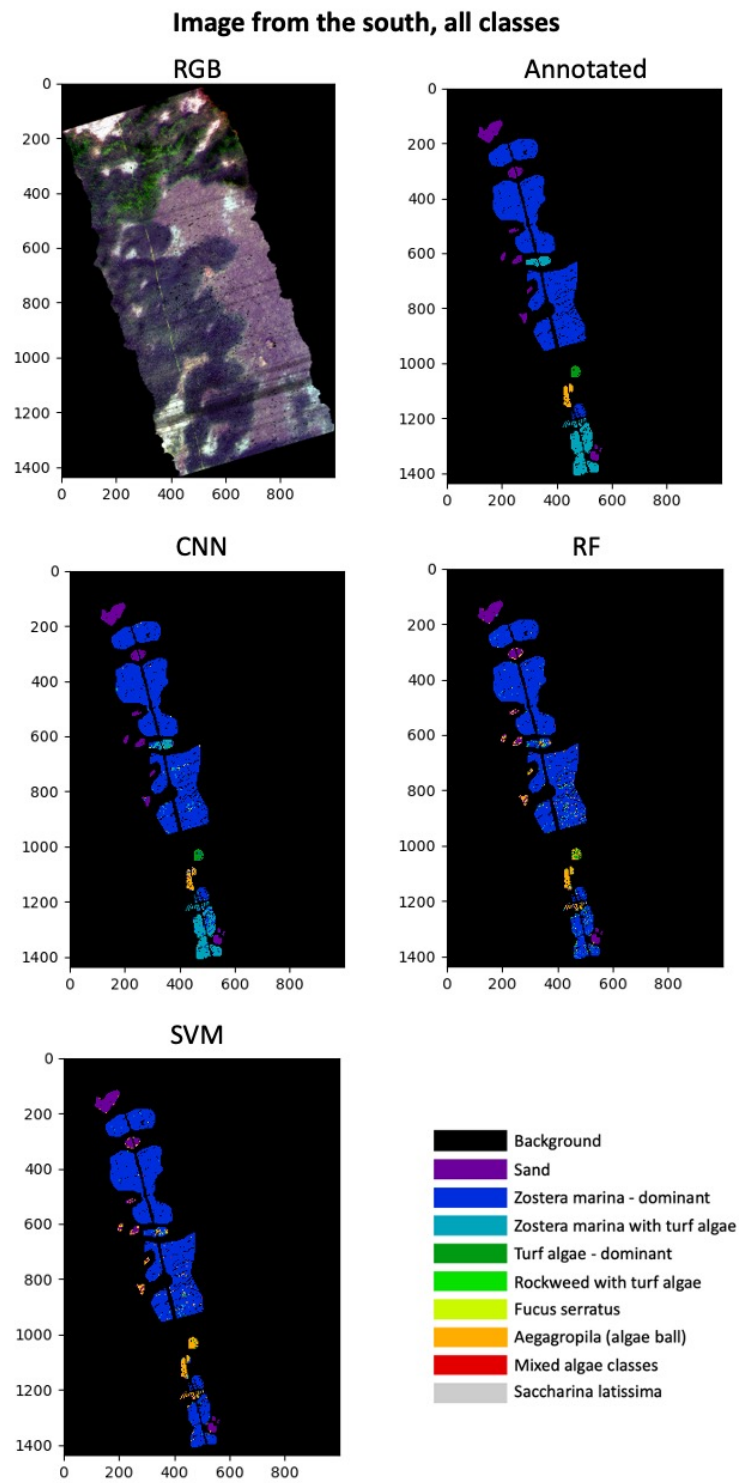
Figure 6.23 shows the classification results on the image from the South, for all classes. It can be observed that CNN is the best classifier to capture the correct pattern in the data. Both RF and SVM classifies class 3 (Zostera Marina with turf algae) as class 2 (Zostera Marina - dominant) (the bottom part that is light blue in the annotation image), which is in accordance with Tab. 6.5. CNN is able to classify sand correctly, where RF and SVM classifies some of the sand as Aegagropila (algae ball). Also, for turf (the small green patch at ca (500, 1000)), CNN classifies it correctly, while RF and SVM classifies it as Zostera Marina with turf algae and Aegagropila (algae ball), respectively. This is also in accordance with the confusion matrices in Tab. 6.5.

Figure 6.24 shows the results for the classification of the image from the North, for all classes. None of the classifiers are able to capture the region of rockweed with turf algae (the green region in the top of the annotated image). CNN classifies it as class 2 (Zostera Marina - dominant), class 3 (Zostera Marina with turf algae) and class 7 (Aegagropila (algae ball)). RF classifies it mostly as Zostera Marina with turf and also SVM classifies it as Zostera Marina with turf algae. CNN is not able to classify the sand region (purple at the top of the annotated image), and rather classifies it as Zostera Marina with turf algae. RF and SVM are able to identify sand correctly. Both RF and SVM wrongly classifies class 3 (Zostera Marina with turf algae) as class 2 (Zostera Marina - dominant) at the light blue patch at ca (600, 1100) in the annotated image. In this area, CNN has some misclassifications where it classifies class 3 (Zostera Marina with turf algae) as Aegagropila (algae ball). This is also the case for the patch that is located above this area, at ca (500, 900). At this area, RF and SVM are able to correctly classify it as Zostera Marina. All these results are in good correspondence with the confusion matrices in Tab. 6.6.

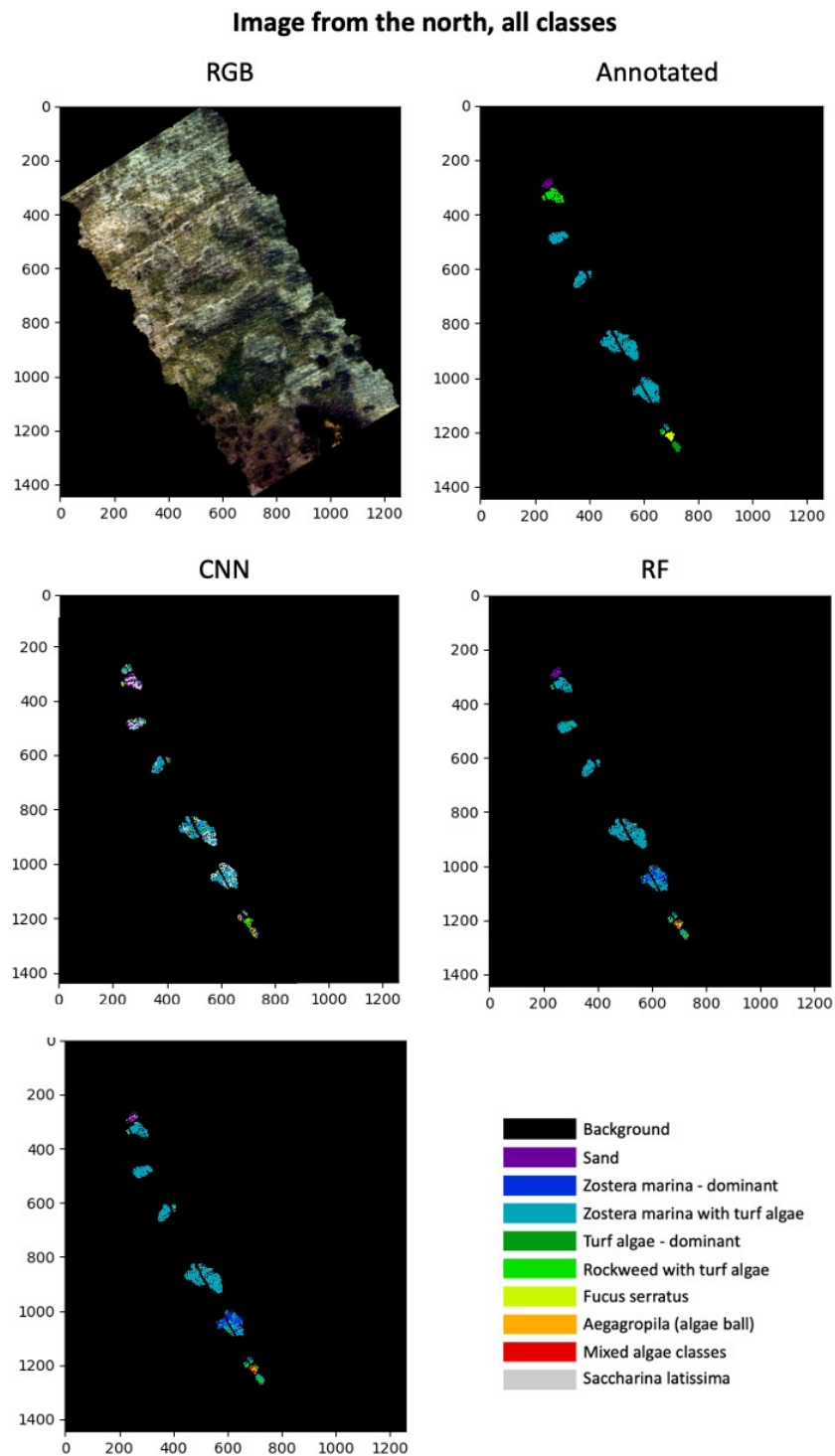
Figure 6.25 shows the classification results for the image from the South, for merged classes. Here, CNN is performing good, picking up the correct patterns in the image. A thorough inspection of the figure shows that it is possible to identify a small amount of misclassifications at ca (400, 400), where some of class 10 (Zostera Marina) have been classified as class 11 (Kelp). RF also performs well, but it has some misclassifications. Some samples of class 4 (Turf algae) is classified as class 7 (Aegagropila (algae ball)), also some of class 1

(Sand) is classified as class 7 (Aegagropila (algae ball)). SVM has the lowest performance, it classifies almost all of class 4 (Turf algae) as class 7 (Aegagropila (algae ball)), which also the case in the class 1 (Sand) area. These results are in accordance with the confusion matrices in Tab. 6.8.

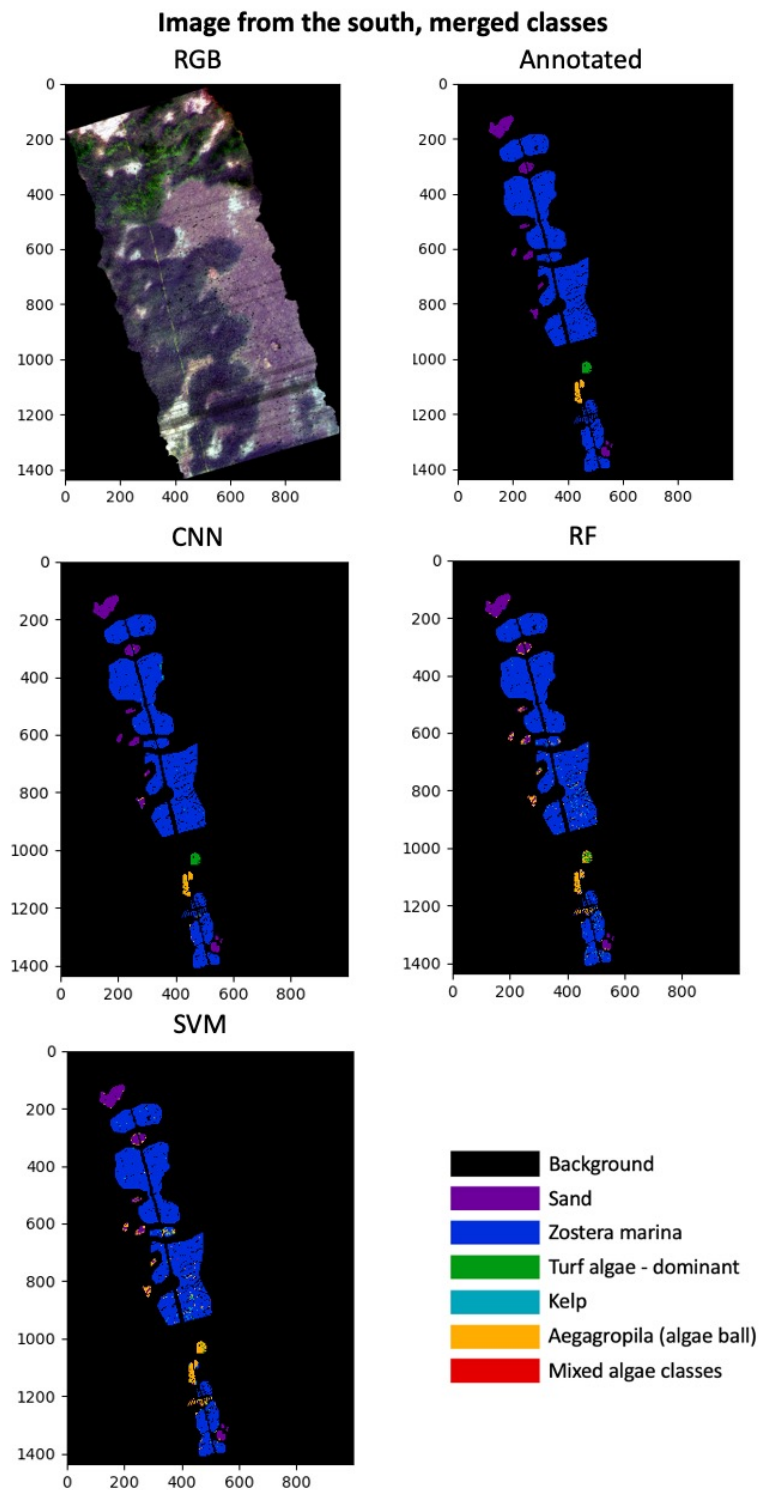
Figure 6.26 shows the classification results for the image from the North, for merged classes. Here, RF has the best performance; it is able to identify and correctly classify class 10 (Zostera Marina) and class 1 (Sand). It does have some difficulties in classifying class 11 (Kelp) and class 4 (Turf algae), where in both cases it classifies it as class 10 (Zostera Marina) and class 7 (Aegagropila (algae ball)). For SVM, the performance is similar to the RF. It does capture class 10 (Zostera Marina) and classifies it correctly for the most part, however for class 1 (Sand), it classifies some of it as class 11 (Kelp). In the bottom part of the image, SVM has the same tendency as RF, classifying class 11 (Kelp) and class 4 (Turf algae) as class 7 (Aegagropila (algae ball)) and class 10 (Zostera Marina). For CNN, there is a lot of misclassifications. For class 1 (Sand), it classifies almost all of it as class 10 (Zostera Marina). For class 10 (Zostera Marina), it classifies much of it as class 11 (Kelp) and class 7 (Aegagropila (algae ball)). Then, for class 4 (Turf algae), it classifies most of it as class 10 (Zostera Marina). These results are in accordance with the confusion matrices in Tab. 6.9 as expected.



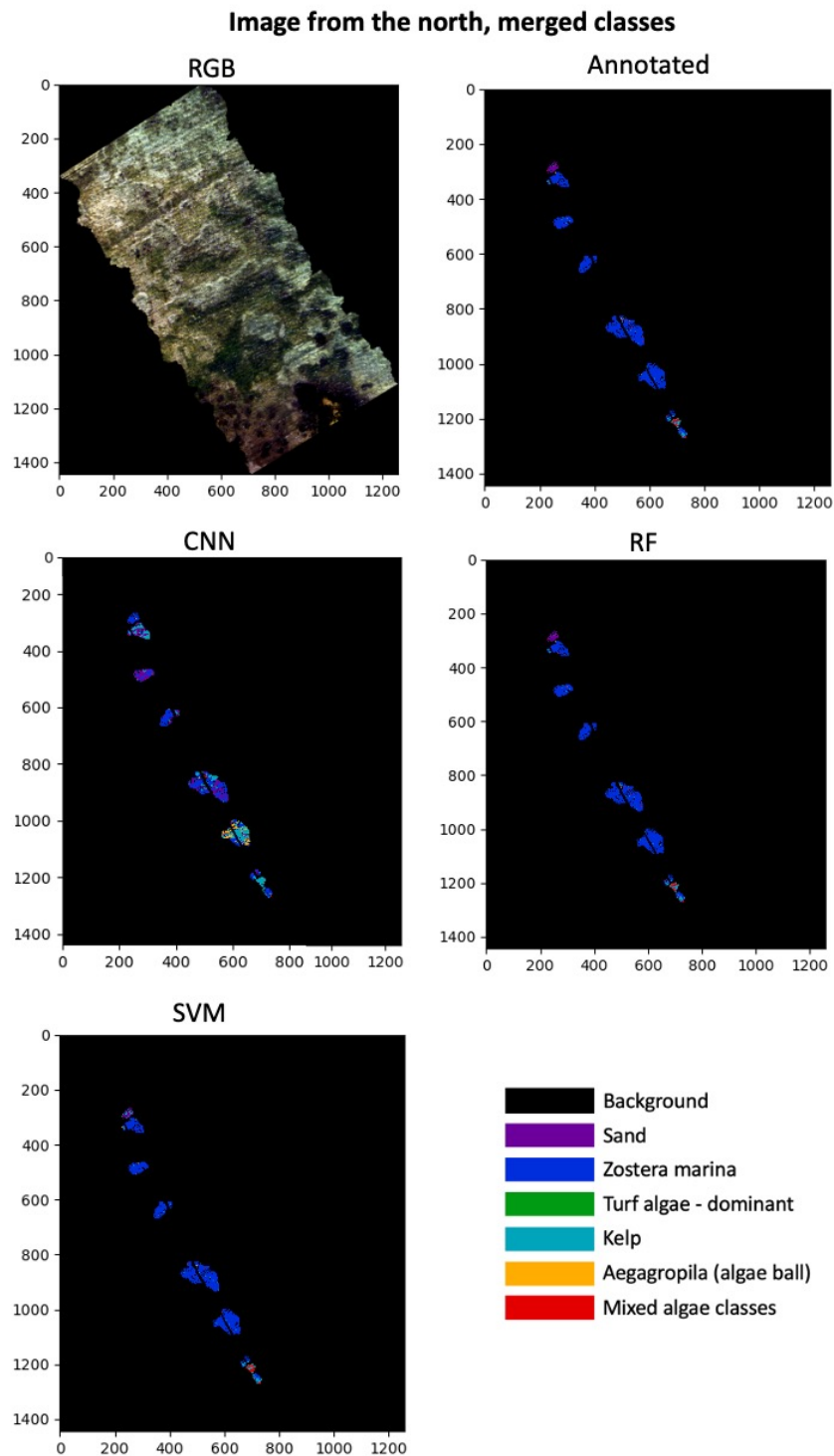
**Figure 6.23:** All classifiers applied to the image from the South, with all classes.



**Figure 6.24:** All classifiers applied to the image from the North, with all classes.



**Figure 6.25:** All classifiers applied to the image from the South, with merged classes.



**Figure 6.26:** All classifiers applied to the image from the North, with merged classes.

## 6.5 Discussion

This project trained and applied three ML models to hyperspectral drone data with the goal of identifying different underwater vegetation species. The three ML models were RF, SVM and CNN. The models were trained and tested on an area South of a small peninsula as shown in Fig. 6.13. The established models were then tested for robustness on an independent image from the North side of the peninsula. Two cases were evaluated; one case with all the classes that were available, and one case where some of the classes were merged.

Dimensionality reduction was performed by PCA, which transformed the 173 spectral bands into 25 principal components explaining 95.73 % of the variance in the dataset. A plot of the first three principal components is presented in Fig. 6.20b, showing how the principal components capture the patterns of the reflectance spectra (Fig. 6.15 (a) and 6.15 (b)).

When applied to the image from the North side for robustness evaluation, the performance of CNN has a significant decrease. For the all class case, CNN does perform good at classifying TP for class 3 (*Zostera Marina* with turf algae). However, this is not true for the rest of the classes. CNN has a low amount of TP for the other classes, and there are several possible reasons why this occurs. Firstly, the window size of 11x11 pixels could be less suitable for the image from the North side. In the image from the North side, there are several small patches of species, rather than big, continuous patches of species as in the area at the South side (see Sec. 6.2.2). In contrast to the South side, where most of the neighbourhoods will appear homogeneous, this could cause difficulties for the CNN because the neighbourhoods will contain a variety of different species. The solution to this could be to train the network with smaller neighbourhoods to evaluate if this improves the classification accuracy. Secondly, the conditions on the North side during data acquisition were somewhat different than the conditions on the South side. During data acquisition from the North side, it was windier and increased turbidity led to more grainy and obscure measurements. The high accuracy in the reported statistics (see Tab. 6.3) for the validation dataset and the image from the South (for both the all class case and the merged class case), and the low accuracy in the reported statistics for the image from the North (for both the all class case and the merged class case), might suggest that CNN is slightly overfitted. An attempt to counteract the overfitting could have been to include samples from the North side in the training and validation, but then the independent robustness evaluation would lose its value. Another possible explanation could be that these classes require information about the texture of the vicinity, i.e. the spatial information, in addition to the spectral information, which is what the CNN offers. It may also be that CNN experience higher degree of "leak" of information from the training data to the validation data than RF and SVM



(see Sec. 6.3.1).

None of the models are capable of precisely and reliably predicting TP for class 4 (Turf algae). CNN predicts class 4 (Turf algae) with high accuracy when applied to the validation dataset and the image from the South. When CNN is applied to the image from the North for robustness evaluation, CNN predicts 0 TP for class 4 (Turf algae), confusing it with class 2 (Zostera Marina - dominant), class 7 (Aegagropila (algae ball)), class 10 (Zostera Marina) and class 11 (Kelp). This is true for both the all class case and the merged class case. This pattern is similar for RF and SVM, where their performances significantly drop when applied to the image from the North in the robustness evaluation. However, RF confuses class 4 (Turf algae) with class 3 (Zostera Marina with turf algae), class 7 (Aegagropila (algae ball)) and class 10 (Zostera Marina). SVM confuses class 4 (Turf algae) with class 3 (Zostera Marina with turf algae), class 5 (Rockweed with turf algae), class 7 (Aegagropila (algae ball)), class 10 (Zostera Marina) and class 11 (Kelp). The problems caused by class 4 (Turf algae) are expected because it is challenging to identify turf algae as a single class. Discussion with Viktor R. Tóth highlighted the difficulties of identifying turf algae as a single class due to the fact that the turf algae appears in different forms when attached to different species in different locations, while the concentration and prevalence varies significantly. This makes it challenging to identify turf algae across multiple locations, which might explain why non of the methods are able to precisely predict turf algae when applied to the image from the North.

RF and SVM had a slightly lower performance than CNN on the validation data and on the image from the South. However, when applied to the image from the North, RF and SVM scores higher in the accuracy evaluation than CNN (for both the all class case and the merged class case). This could be due to the difference in the structure of the input data. RF and SVM classify pixel per pixel, while CNN takes the neighbourhood pixels into consideration as well. This might suggest that RF and SVM are more capable of generalization, perhaps because of a lower capacity for learning and hence less change of overfitting, than CNN for this type of data. The capability of generalization is important if the goal is to train a model which is meant to be applied to different study areas under different conditions (wind, sun, turbidity). The computational power needed to extract, store and classify the 11x11x25 image cubes used for CNN is significantly larger than what is needed for pixel-wise classification for RF and SVM. RF was by far the fastest method to apply, both in training and when applied to the data, with the time needed in orders of a few minutes. Training and applying SVM was a bit slower than RF, with an approximate doubling of minutes. CNN, on the other hand, needed approximately an hour for training, while only a few minutes for classifying the validation data. The extraction of image cubes for CNN was a time consuming process, where a supercomputer

might be needed in order to the process to be manageable for large-scale and near-real time applications.

Comparing the performance of CNN with other relevant research, the method established here shows similar findings. Makantasis et al. [26] applied a CNN to hyperspectral data on different scenes, with a 5x5 neighbourhood. The reported OA is above 98 % for all scenes. Li, Zhang and Shen [28] applied a 3D CNN to a selection of different scenes, using different neighbourhood sizes for the different scenes, obtaining OA greater than 95 % for all scenes, AA greater than 93 % for all scenes and K greater than 95 for all scenes. Comparing these results with the reported statistics in Tab. 6.3, the CNN established here is performing similarly to the aforementioned studies. Merging the classes resulted in even better performance in comparison to the findings in [26, 28]. Note, in this project robustness evaluation was also performed on an independent dataset, which is in contrast to the studies in [26, 28].

As expected, all methods had high scores in the evaluated statistics when applied to the validation dataset and the image from the South. This is in good correspondence with previous research [4, 26, 28, 84], who also reported high scores in the computed statistics for CNN, SVM and RF for hyperspectral data classification and classification of vegetation. The accuracy for all models decreased when their robustness was tested on an independent image from the North side of the peninsula. This was also expected, but the significant decrease in the performance of CNN was more than expected. During validation of the methods, all were more or less capable of differentiating between the classes that showed a clear spectral signature at the red edge (class 2 (Zostera Marina - dominant)), class 3 (Zostera Marina with turf algae), class 5 (Rockweed with turf algae), class 6 (Fucus serratus), class 8 (Mixed algae classes), class 10 (Zostera Marina) and class 11 (Kelp), and those which did not show a clear spectral signature at the red edge class 1 (Sand), class 4 (Turf algae), class 7 (Aegagropila (algae ball)) and class 9 (Saccharina latissima). However, when the methods were tested for robustness (by applying them to the image from the North side), all models were struggling to separate these classes, especially confusing class 4 (Turf algae) with other classes. This confirms the difficulties of classifying turf algae and turf algae contaminated species as described in the previous paragraphs in this section.

The established methods were applied in prediction mode to an additional image from the South side, as seen in Fig. 7.2. Although this was not the focus of this thesis, using the ML methods in prediction might provide an insight how the methods perform in application. The results can be found in the Appendix, Ch. 7.

Including the bathymetry might help improve the performance of the models,

due to the change in reflectance response of the underwater vegetation with the change in water depth (see Sec. 2.2). Other research indicate that the inclusion of coordinates might improve the classification of marine vegetation [4]. The application of bathymetry and coordinates might be explored as an attempt to improve the established models.

It can be summarized that the established models perform good in validation, while their performance decreases when evaluated for robustness on an independent dataset. This could be due to various external conditions which influences the quality of the data, such as wind speed, sunlight, turbidity and different distribution of the species and/or overfitting during model establishment. CNN had the highest performance in validation, while RF and SVM performed better when evaluated for robustness. This might suggest that the CNN is overfitted. Once again, as discussed above, class 4 (Turf algae) was a source of erroneous classifications when the models were applied to the image from the South, and the image from the North. Although there are certain limitations which requires further tuning of the models in order to ensure robustness, the results suggest that the here established methods have the potential to be used in applications.

## 6.6 Conclusion

This project showed that it is possible to image and classify marine vegetation in the Norwegian coast by the use of hyperspectral drone data and ML methods. Three ML methods were evaluated here; CNN, RF and SVM. The performance of CNN in validation and when applied to the image from the South was outstanding (see Fig. 6.18 for an overview of the data). RF and SVM also performed well during validation and when applied to the image from the South. A robustness evaluation was performed with independent data from the North side. The computed statistical measures show good performance for both RF and SVM, while the performance of CNN has a significant decrease compared to the performance of CNN in the validation. This suggests that although CNN has outstanding performance when applied to the validation dataset, it requires further tuning in order to be used in applications. Turf algae (class 4) is for all models causing erroneous classifications, where the models confuse class 4 (Turf algae) with other classes. This is especially the case, when the other classes include turf algae.

The methods were also applied to another image from the South side in prediction mode (see Fig. 7.2) to compare the actual patterns of vegetation with the patterns the models are able to identify. See Appendix, Ch. 7 for the results.

This project provided insight in both the opportunities and limitations in classifying underwater vegetation using ML. While CNN has been extensively used for image classification, it has to the author's best knowledge not been used to classify hyperspectral drone data with the aim of classifying different marine vegetation in Norway. The results show that the studied ML methods applied to this purpose is promising, and might be subject to up-scaling and extensive applications.

## Future Work

A well established pipeline for identifying marine vegetation can be an important method for surveillance and detection of trends that occur in the blue forests along the Norwegian coast. Further development of the here established methods will be needed in order to create reliable system for monitoring the blue forests. To achieve this, more in-situ data is needed to advance the robustness of the methods. Identification of which classes are of importance for the specific applications and tuning of the models accordingly could be done. Identification of turf algae is important for evaluating the physiological state of the blue forests. It should be investigated further how the models could be

retrained with emphasis of identifying turf algae.

Other dimensionality reduction methods could also be tested to evaluate the performance of the models. Feature extraction can be studied for applying it to upcoming hyperspectral (satellite) sensors. If this research is to be applied to multispectral sensors, a feasibility study needs to be conducted whether the spectral information is satisfactory for marine vegetation mapping provided by the freely available spaceborne sensors.

The code used to implement the methods, as well as the scripts for extraction of image cubes etc. will be made open source in the near future. The established methods will need to be re-trained on remote sensing reflectance ( $R_{rs}$ ) (here they were trained on the water leaving radiance ( $L_w$ ) due to unavailability of  $R_{rs}$ ) when available.





# Appendix

## 7.1 RF and SVM applied to additional image from the South side

The established algorithms were applied to an additional image from the South in prediction mode for evaluation of how the methods predict the spatial patterns of the vegetation in the image. Note: the image was included in the dataset, i.e. it is not an independent dataset. Figure 7.1 shows an overview of where the image is placed relative to the image from the North (N) and the image from the South (S). The pink arrow points to the additional image.

RF and SVM were applied for every pixel in the image. CNN was applied every fifth pixel, both in height and width direction, due to the overlap of information for pixel right next to each other and due to the computational burden of predicting on all classes (a cube of 11x11x25 data points needs to be stored and handled for every pixel in the image).

Figure 7.2 shows the result of the prediction for all methods on the additional image. The figure shows that all methods are capable of predicting class 1 (Sand) in the areas where, in the RGB image, it appears that class 1 (Sand) is present. In the bottom part of the image, all methods seem to follow the pattern of class 2 (Zostera marina - dominant) as seen in the RGB image. In the annotated image where class 3 (Zostera marina with turf algae) is marked, CNN predicts it precisely, however, RF and SVM mostly predicts class 2 (Zostera marina -

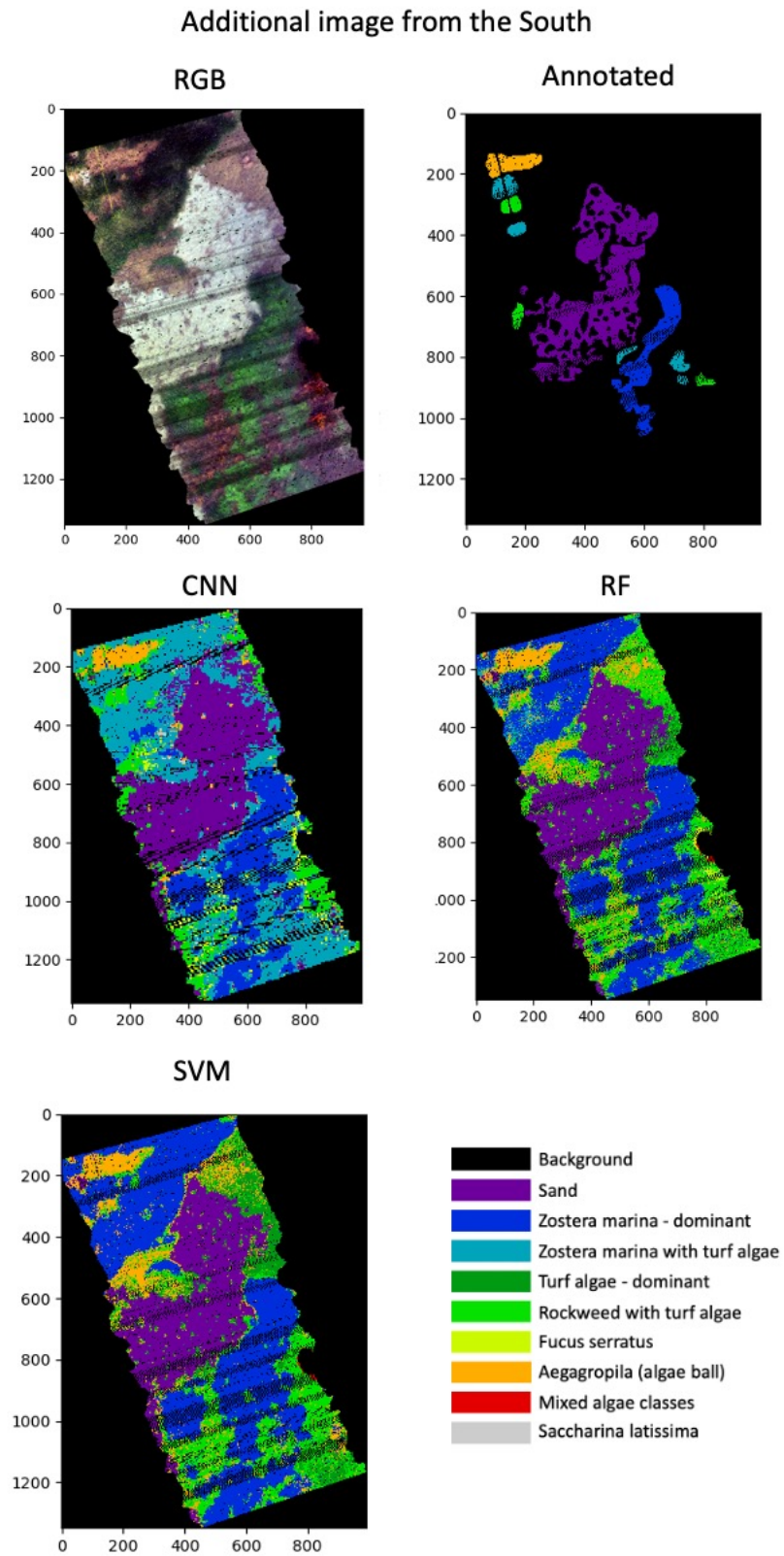
dominant). CNN predicts far more of class 3 (Zostera marina with turf algae) than RF and SVM. Where CNN predicts class 3 (Zostera marina with turf algae), RF and SVM mostly predicts class 2 (Zostera marina - dominant) combined with class 5 (Rockweed with turf algae) and class 4 (Turf algae). At the top of the image where CNN predicts class 3 (Zostera marina with turf algae), it appears as if class 3 (Zostera marina with turf algae) is present based in the RGB image. RF and SVM predicts class 4 (Turf algae) in the left part of the image, at ca (700, 500). In this area, CNN predicts class 3 (Zostera marina with turf algae). Further, all methods correctly classify class 5 (Rockweed with turf algae) in the annotated patch at ca (220, 650). However, in the patch at ca (200, 310) CNN and RF recognise class 5 (Rockweed with turf algae), SVM fails in prediction of class 5 (Rockweed with turf algae), instead predicts class 7 (Aegagropila (algae ball)).

Based on the annotated parts of the image, CNN outperformed both RF and SVM with correctly classified pixels. Rf had more correctly classified pixels than SVM in the annotated parts. The prediction outside the annotated areas seemed to follow a pattern of; RF and SVM are so to speak predicting almost the same. However, CNN predicts a lot of class 3 (Zostera marina with turf algae), while RF and SVM both fails to recognise some of it in the annotated areas, and predicts far less of class 3 (Zostera marina with turf algae). To know exactly which method is superior in this case is challenging due to the lack of ground truth. Still, based on CNN's precise prediction in the annotated parts of the image (and based on visual analysis of the RGB image), the conclusion is that CNN was the overall most suited model for this image. This is in correspondence with the results presented in Sec. 6.4.





**Figure 7.1:** Map showing where the additional image is placed relative to the image from the North (N) and the image from the South (S). The pink arrow points to the additional image that was used for visual presentation of the results.



**Figure 7.2:** The established models (for all classes) applied on an additional image from the South side.

## 7.2 Conferences and field work

I participated in the following conferences with contribution and field works during spring 2022:

- Arctic Science Summit Week, March 2022, with the poster "*Machine Learning for Classifying Marine Vegetation from Hyperspectral Drone Data in the Norwegian coast*"
- Field work in Bodø, March 2022  
Frontiers, May 2022, with an oral presentation "*Comparison of Chlorophyll-a content retrieval algorithms in the Arctic waters*"
- Scientific seminar at the Balaton Limnological Research Institute, May 2022, with an oral presentation "*Machine Learning for Classifying Marine Vegetation from Hyperspectral Drone Data in the Norwegian coast*"
- Field work in Tihany, May 2022



# Bibliography

- [1] R. N. Gibson, R. Atkinson, J. Gordon, F. In, L. Airoidi, and M. Beck. Loss, status and trends for coastal marine habitats of europe. *An Annual Review*, 45:345–405, 06 2007.
- [2] S. Hynes, T. Bekkeby, C. Hartvig, H. Gundersen, W. Chen, and K. van Assche. Ecosystem accounting’s potential to support coastal and marine governance. *Marine Policy*, 112:103758, 02 2020.
- [3] H. Simmons and K. Howell. Machine learning algorithms for habitat type prediction from drone-based multispectral imagery: A comparison of classifiers. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
- [4] K. Clarke, A. Hennessy, A. McGrath, R. Daly, S. Gaylard, A. Turner, J. Cameron, M. Lewis, and M. Fernandes. Using hyperspectral imagery to investigate large-scale seagrass cover and genus distribution in a temperate coast. *Sci Rep* 11, 4182 (2021). <https://doi.org/10.1038/s41598-021-83728-6>.
- [5] N. Oppelt, F. Schulze, I. Bartsch, K. Kuhwald, and I. Eisenhardt. Hyperspectral classification approaches for intertidal macroalgae habitat mapping: A case study in heligoland. *Optical Engineering*, 51:111703, 11 2012.
- [6] Norwegian Blue Forests Network. Blue forests basic. <https://nbf.no/blue-forests-basics/>. Last Accessed: 2022-01-26.
- [7] Norwegian Blue Forests Network. Blue forests: Valuable nature that contributes to food security and helps against climate and poverty. <https://nbf.no/seminar-summary/>. Last Accessed: 2022-05-24.
- [8] R. Pu, S. Bell, and D. English. Developing hyperspectral vegetation indices for identifying seagrass species and cover classes. *Journal of Coastal Research*, 313:595–615, 05 2015.
- [9] K. Filbee-Dexter and T. Wernberg. Rise of Turfs: A New Battlefield for

Globally Declining Kelp Forests. *BioScience*, 68(2):64–76, 01 2018.

- [10] H. Christie, G. S. Andersen, T. Bekkby, C. W. Fagerli, J. K. Gitmark, H. Gundersen, and E. Rinde. Shifts between sugar kelp and turf algae in norway: Regime shifts or fluctuations between different opportunistic seaweed species? *Frontiers in marine science*, page 6:72, 02 2019.
- [11] E. Rinde and H. Christie. Varsko! vi kan få «oslofjord-tilstander» langs store deler av norskekysten. [https://www.aftenposten.no/meninger/debatt/i/V97PK4/varsko-vi-kan-faa-oslofjord-tilstander-langs-store-deler-av-norskekysten?fbclid=IwAR0jZFjezLDCTymUqXeDad9-tU\\_JfTMVEgArVBdkRYa4\\_wfZZYhJWxZr-GA](https://www.aftenposten.no/meninger/debatt/i/V97PK4/varsko-vi-kan-faa-oslofjord-tilstander-langs-store-deler-av-norskekysten?fbclid=IwAR0jZFjezLDCTymUqXeDad9-tU_JfTMVEgArVBdkRYa4_wfZZYhJWxZr-GA). Last Accessed: 2022-05-24.
- [12] K. Kuhwald, J. S. von Deimling, P. Schubert, and N. Oppelt. How can sentinel-2 contribute to seagrass mapping in shallow, turbid baltic sea waters? *Remote Sensing in Ecology and Conservation*, 12 2021.
- [13] E. Moksness, E. Dahl, and J. Støttrup. The Norwegian Programme for Mapping of Marine Habitats – Providing Knowledge and Maps for ICZMP. *Global challenges in integrated coastal zone management*, pages 21–31, 2013.
- [14] B. K. Veettil, R. D. Ward, M. C. Lima, M. Stankovic, P. N. Hoai, and N. X. Quang. Opportunities for seagrass research derived from remote sensing: A review of current methods. *Ecological Indicators*, 117:106560, 2020.
- [15] A. Dekker, J. V. Brando, Anstee, S. Fyfe, T. Malthus, E. Karpouzli, A. Larkum, R. Orth, and C. Duarte. *Remote Sensing of Seagrass Ecosystems: Use of Spaceborne and Airborne Sensors*, pages 347–359. 05 2007.
- [16] H. Tasoff. An opaleye-opening result. <https://www.news.ucsb.edu/2021/020487/opaleye-opening-result>. Last Accessed: 2022-05-25.
- [17] NIVA. Turf algae growth in the sognefjord. <https://www.niva.no/en/reports/turf-algae-growth-in-the-sognefjord>. Last Accessed: 2022-05-25.
- [18] M. Moniruzzaman, S. Islam, P. Lavery, M. Bennamoun, and C. Lam. Imaging and classification techniques for seagrass mapping and monitoring: A comprehensive survey. 02 2019.
- [19] W. Nijland, L. Reshitnyk, and E. Rubidge. Satellite remote sensing of canopy-forming kelp on a complex coastline: A novel procedure using the landsat image archive. *Remote Sensing of Environment*, 220:41–50, 01 2019.

- [20] D. R. Lyzenga. Remote sensing of bottom reflectance and water attenuation parameters in shallow water using aircraft and landsat data. *International Journal of Remote Sensing*, 2(1):71–82, 1981.
- [21] K. Arrigo, M Babin, S. Belanger, J Comiso, M. Forget, R. Frouin, C. Goyens, V. Hill, T. Hirawake, A. Matsuoka, B. Mitchell, D. Perovich, R. Reynolds, K. Stammes, and M. Wang. *IOCCG Report Number 16*, 2015.
- [22] S. Chand and B. Breen. Detecting the spatial variability of seagrass meadows and their consequences on associated macrofauna benthic activity using novel drone technology. *Remote Sensing*, 14:160, 12 2021.
- [23] J. Kok, S. Bainbridge, M. Olsen, and P. Rigby. Towards effective aerial drone-based hyperspectral remote sensing of coral reefs. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pages 1–6, 2020.
- [24] J. Campbell and R. Whyne. *Introduction to Remote Sensing*. The Guilford Press, 2011.
- [25] A. Courville I. Goodfellow, Y. Bengio. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [26] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. pages 4959–4962, 01 2015.
- [27] D. Perez, K. A. Islam, V. Hill, R. Zimmerman, B. Schaeffer, Y. Shen, and J. Li. Quantifying seagrass distribution in coastal water with deep learning models. *Remote Sensing*, 12:1581, 05 2020.
- [28] Y. Li, H. Zhang, and Q. Shen. Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sensing*, 9:67, 01 2017.
- [29] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz. A fast and compact 3-d CNN for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.
- [30] C. Elachi and J. Zyl. *Introduction to the physics and techniques of remote sensing*. John Wiley & Sons, 2006.
- [31] S. Martin. *An introduction to ocean remote sensing*. Cambridge University Press, 2014.

- [32] A. Albert. Inversion technique for optical remote sensing in shallow water. 2005.
- [33] J. Hedley, A. Harborne, and P. Mumby. Technical note: Simple and robust removal of sun glint for mapping shallow-water benthos. *International Journal of Remote Sensing - INT J REMOTE SENS*, 26:2107–2112, 05 2005.
- [34] Absorption spectrum of liquid water. [https://commons.wikimedia.org/wiki/File:Absorption\\_spectrum\\_of\\_liquid\\_water.png](https://commons.wikimedia.org/wiki/File:Absorption_spectrum_of_liquid_water.png). Last Accessed: 2022-05-10.
- [35] E. Green, P. Mumby, A. Edwards, and C. Clark. *Remote Sensing Handbook for Tropical Coastal Management*. Unesco Publishing, 2000.
- [36] H. Dierssen and K. Randolph. Remote sensing of ocean color. *Earth System Monitoring, by Orcutt, John, Springer Science+Business Media New York*, page 439, 2013.
- [37] J. Suomalainen, R. Oliveira, T. Hakala, N. Koivumäki, L. Markelin, R. Näsi, and E. Honkavaara. Direct reflectance transformation methodology for drone-based hyperspectral imaging. *Remote Sensing of Environment*, 266, 2021.
- [38] D. Mishra, S. Narumalani, R. Rundquist, and M. Lawson. High-resolution ocean color remote sensing of benthic habitats: A case study at the roatan island, honduras. *Geoscience and Remote Sensing, IEEE Transactions on*, 43:1592 – 1604, 08 2005.
- [39] Great pound foundation. <https://greatpondfoundation.org/explore-our-work/water-quality-monitoring/>. Last Accessed: 2022-05-11.
- [40] A. Thorhaug, A. Richardson, and G. Berlyn. Spectral reflectance of the seagrasses: *Thalassia testudinum* , *Halodule wrightii* , *Syringodium filiforme* and five marine algae. *International Journal of Remote Sensing*, 28:1487 – 1501, 04 2007.
- [41] C. Dong, G. Zhao, Y. Meng, B. Li, and B. Peng. The effect of topographic correction on forest tree species classification accuracy. *Remote Sensing*, 12(5), 2020.
- [42] K. Herbert. Prisma (hyperspectral precursor and application mission). <https://earth.esa.int/web/eoportal/satellite-missions/p/prisma-hyperspectral>. Last Accessed: 2022-02-13.



- [43] NTNU Norwegian University of Science and Technology. Prisma (hyper-spectral precursor and application mission). <https://www.ntnu.edu/web/smallsat/mission-hyper-spectral-camera>. Last Accessed: 2022-02-13.
- [44] J. Werdell. Ocean color instrument. <https://pace.oceansciences.org/oci.htm>. Last Accessed: 2022-02-13.
- [45] G. Chen. *Advances in Agricultural Machinery and Technologies*. CRC Press, 2018.
- [46] C. De Los Santos, D. Krause-Jensen, T. Alcoverro, Teresa N. Marba, C. Duarte, M. Katwijk, M. Pérez, J. Romero, J. Sánchez Lizaso, G. Roca, E. Jankowska, J. Pérez-Lloréns, J.Fournier, M. Montefalcone, G. Pergent, J. Ruiz Fernandez, S. Cabaço, K. Cook, R. Wilkes, and R. Santos. Recent trend reversal for declining european seagrass meadows. *Nature Communications*, 10, 07 2019.
- [47] V. Peña Freire, J. Grall, J. Hall-Spencer, S. Gubbay, G. Saunders, H. Tyler-Walters, N. Dankers, F. Otero-Ferrer, J. Forde, K. Fürhaupter, R. Haroun Tabraue, and N. Sanders. A5.51 atlantic maerl beds. *European Red List of Habitats - Marine Habitat Group*, 2016.
- [48] W. T. Haller L. A. Gettys and D. G. Petty. *Biology and Control of Aquatic Plants*. Aquatic Ecosystem Restoration Foundation, 2014.
- [49] C. Loyd. Landsat 8 bands. <https://landsat.gsfc.nasa.gov/satellites/landsat-8/landsat-8-bands/>. Last Accessed: 2022-04-8.
- [50] K. Koutroumbas and S. Theodoridis. *Pattern Recognition*. Academic Press, 2009.
- [51] D. J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [52] J. O. Villalobos. Test, training and validation sets. <https://www.brainstobytes.com/test-training-and-validation-sets/>. Last Accessed: 2022-02-17.
- [53] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2014.
- [54] K. P. Murphy. *Probabilistic Machine Learning - An Introduction*. The MIT Press, 2022.
- [55] K. Hoffman. Machine learning: How to prevent overfit-

- ting. <https://medium.com/swlh/machine-learning-how-to-prevent-overfitting-fdf759cc00a9>. Last Accessed: 2022-02-16.
- [56] G. Lemaitre, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research* 18 (2017), 1-5, 01 2017.
- [57] J. Brownlee. Hyperparameter optimization with random search and grid search. <https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/>. Last Accessed: 2022-04-19.
- [58] S. Gutta. Machine learning metrics in simple terms. <https://medium.com/analytics-vidhya/machine-learning-metrics-in-simple-terms-d58a9c85f9f6>. Last Accessed: 2022-05-08.
- [59] M. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22, 3:276–282, 2012.
- [60] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–20, 04 2019.
- [61] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–20, 04 2019.
- [62] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surv Land inf Syst*, 62, 01 2002.
- [63] B. M. Achsan. Support vector machine: Classification. <https://medium.com/it-paragon/grid-search-f24a73a8a0ac>. Last Accessed: 2022-03-07.
- [64] scikit-learn developers. `sklearn.svm.svc`. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>. Last Accessed: 2022-05-10.
- [65] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Pearson, 2010.
- [66] R. Vohra. Convolutional neural networks. <https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e>. Last Accessed: 2022-05-01.
- [67] H. H. Aghdam and E. J. Heravi. *Guide to Convolutional Neural Networks*.

Springer International Publishing, 2017.

- [68] `tf.keras.losses.sparse_categorical_crossentropy`. [https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy?version=nightly](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy?version=nightly). Last Accessed: 2022-05-19.
- [69] `tf.keras.optimizers.adam`. [https://www.tensorflow.org/api\\_docs/python/tf/keras/optimizers/Adam?version=nightly](https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam?version=nightly). Last Accessed: 2022-05-19.
- [70] C. Rorres and H. Anton. *Elementary Linear Algebra with Supplemental Applications, 11th Edition International Student Version*. John Wiley Sons, 2014.
- [71] Imperial War Museums. A brief history of drones. <https://www.iwm.org.uk/history/a-brief-history-of-drones>. Last Accessed: 2022-04-19.
- [72] C. Sánchez-Bou and J. López-Pujol. The coming revolution: The use of drones in plant conservation. *Collectanea Botanica*, 33, 12 2014.
- [73] Da-Jiang Innovations. Matrice 600pro. <https://www.dji.com/no/matrice600-pro>. Last Accessed: 2022-04-8.
- [74] RESONON Hyperspectral Imaging Solutions. Pika l. <https://resonon.com/Pika-L>. Last Accessed: 2022-04-8.
- [75] G. Andersen. sukkertare. <https://snl.no/sukkertare>. Last Accessed: 2022-05-18.
- [76] G. Andersen. sagtang. <https://snl.no/sagtang>. Last Accessed: 2022-05-18.
- [77] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Yangqing Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. <https://www.tensorflow.org/>, 2015. Last Accessed: 2022-04-10.
- [78] scikit-learn developers. `sklearn.decomposition.pca`. <https://scikit-learn.org/>

learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.  
Last Accessed: 2022-05-01.

- [79] scikit-learn developers. sklearn.ensemble.randomforestclassifier.  
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>. Last Accessed: 2022-05-08.
- [80] scikit-learn developers. sklearn.model\_selection.gridsearchcv. sklearn.preprocessing.StandardScaler. Last Accessed: 2022-05-01.
- [81] scikit-learn developers. sklearn.model\_selection.gridsearchcv.  
[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html). Last Accessed: 2022-05-01.
- [82] The imbalanced-learn developers. imbalanced-learn documentation.  
<https://imbalanced-learn.org/dev/index.html>. Last Accessed: 2022-04-27.
- [83] K. Tsuda and B. Schölkopf. A primer on kernel methods. In *in Kernel Methods in Computational*, pages 35–70. MIT Press, 2004.
- [84] A. Juel, G. B. Groom, J. Svenning, and R. Ejrnæs. Spatial application of random forest models for fine-scale coastal vegetation classification using object based analysis of aerial orthophoto and dem data. *International Journal of Applied Earth Observation and Geoinformation*, 42:106–114, 2015.



