NTNU
Norwegian University of
Science and Technology

Using Deep Learning for Automatic Classification of Marine Habitats in HiSAS Imagery.

Margrete Sandsbråten Scheide

Marine Technology
Submission date: July 2018
Supervisor: Martin Ludvigsen, IMT

Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN UNDERWATER TECHNOLOGY

Margrete Scheide

Spring 2018

Classification of seafloor and marine habitats in HiSAS imagery using a convolutional neural network

Work Description

Sonar technology has been one of the most important tools for marine exploration in the last century. For the past decades, the co-development of better sonar platforms and increasing image quality has led to the possibility of large scale mapping and classification of the seafloor. By using a synthetic aperture sonar (SAS) installed on an autonomous underwater vehicle, high resolution sonar imagery can be made of large areas of seafloor. Mapping of marine habitats is important for many scientific and industrial purposes. First of all, it is interesting within the field of marine biology and geology to be able to map and research marine habitats and seafloor. Secondly, it could be important for marine industry such as fish farming and offshore facilities to gain knowledge of the surrounding marine environment more efficiently.

Recent development of neural networks has made it accessible for non computer scientists to design a network based on a chosen data set. A convolutional neural network can classify vast amounts of images in a short amount of time, and with a high accuracy dependent on the amount of data available and network structure.

The aim of this project is to combine the developments within sonar technology with the developments of neural networks, by using a convolutional neural network to classify SAS images. This method can increase the efficiency of seabed mapping that is performed today. The goal is to develop an automatic classification and quantification program for SAS sonar imagery of marine habitats. This will increase the area coverage rate, making it possible to explore the ocean basin more efficiently. It will benefit all scientists, engineers and politicians that has an interest in mapping of the seabed.

The case study for this thesis is the Tautra area in the Trondheim Fjord. It has been mapped by the HiSAS sonar on the Hugin class AUV from Kongsberg in 2012, 2013 and 2017. The data sets gathered will be used to train and test a convolutional neural network. Additionally, sonar imagery from Nordleksa will be used to test the performance of the program on a completely new area. The sonar imagery is geo-referenced, which will be used to create maps of areas with marine habitats. The maps will be created using QGIS.

Scope of the Work

- Describe the background and motivation for the project.
- Do a literature review of relevant literature and previous work on convolutional neural networks and classification of marine habitats.
- Develop a method for optimizing the training of the convolutional neural network.
  - A grid search will be used to decide number of classes, convolution layers, filters and filter size.
  - A random grid search optimizing the learning rate, batch size and number of units in the first fully connected layer will be performed.
  - Evaluate the performance using accuracy, loss and confusion matrices. The confusion matrices will form the basis for calculation of precision, recall and F-score.
- Create of a program for pre and post processing of the sonar imagery, which implements the trained neural network and outputs a georeferenced classified image. The program will also automatically output the area of detected corals.
  - The preprocessing prepares the sonar mosaics for the neural network by rescaling into smaller images.
  - In the postprocessing, the images are assembled into the original mosaics, but with a layer of color depending on the result of classification. Georeference of the mosaic is preserved.
- Mathematical morphology will be implemented in the program for increased accuracy, by altering stand alone classifications that differ from its neighboring values.
  - This will be done using the morphological operation closing, which is a dilation followed by erosion using a structuring element suited for the sonar imagery.
- Compare the results of the automatic classification and quantification on another area than the Tautra field. This will be done with sonar imagery from Nord-Leksa, gathered in 2016.
- Create maps with the classified sonar mosaics, showing the extent of marine habitats at Tautra in 2012, 2013 and 2017. This will also be done for the Nord-Leksa data.

The thesis shall be written in English, and edited as a research report including literature survey, description of mathematical models, description of control algorithms, simulation results, discussion and a conclusion including a proposal for further work. Source code shall be provided on a USB drive with code listing enclosed in appendix. It is supposed that Department of Marine Technology, NTNU, can use the results freely in its research work, unless otherwise agreed upon, by referring to the student's work. The thesis shall be submitted in two copies by July 9^{nd}.

Supervisor: Professor Martin Ludvigsen

Co-Supervisor: PhD candidate Øystein Sture

Acknowledgements

I would like to thank my supervisor Martin Ludvigsen for all his guidance and efforts throughout the process of writing this report. A special thanks to him for bringing me along on the 2017 survey of the Tautra field and arranging a meeting at NGU, where I was introduced to several people that were very helpful. Among these are Emlyn Davies of Sintef that spent much time helping me with my preliminary study of coding a CNN model in MATLAB, Christian Malmquist of Kartverket who helped me understand the various areas of applications of my thesis as well as helping me with QGIS, and Terje Thorsnes and his colleagues at NGU who had valuable insight and experience with sonar imagery of corals. I would also like to address a special thank you to Øystein Sture from NTNU AMOS for all his guidance and encouragements. He has spent much time helping me with de-bugging, understanding and development of my classification model, as well as discussing the numerous aspects of the project. Without him I could not have made it as far as I did.

I thank my friends Julie T. Mynors, marine biologists at Åkerblå and Pratim Patel for invaluable insights, comments and encouragement on this thesis. Finally I thank my boyfriend Sigurd Aurvåg Sørum for always being supportive. Your help has been invaluable.

Author,
Margrete Sandsbråten Scheide

Summary

This thesis aims to utilize an established image classification method to create a methodology for classifying cold-water corals in synthetic aperture sonar imagery. This will be performed by combining high-resolution sonar imagery with a convolutional neural network classification algorithm. The result will be an automatic classification model that will decrease the need for human interpretation of the sonar images. The method can be used for quantification of marine habitats in sonar imagery.

A case study was performed with three data sets from the Tautra area in the Trondheim fjord. The data sets are from 2012, 2013 and 2017. Most of the data was used to form a dataset with sub-images of 100x100 pixels used to train a CNN model. The rest of the images were saved for verification of the classification performance. The estimated accuracy on the verification images were 83%, 92% and 95% for 2012, 2013 and 2017 respectively.

A fourth case study was performed on sonar imagery from Nord-Leksa, another region of the Trondheim fjord, which was used to verify the classification performance on a different area than the Tautra ridge. The accuracy of the Nord-Leksa verification image was 89%.

An estimate of the coral coverage for each data set was made, but the results could not be compared because the total area coverage for each year differed. The estimated coral coverage was 0.8 km^2, 0.4 km^2 and 0.7 km^2 for the 2012, 2013 and 2017 Tautra data sets. The classified sonar mosaics showed a dense region of coral in locations where coral reefs have previously been verified. The Nord-Leksa coverage was estimated to be 2.3 km^2, approximately 50% of the total area covered. Similarities of verified coral reefs and the predicted coverage from the classification model occurred in the Nord-Leksa case study as well.

Two sources of error in the methodology was identified. The random initialization of the neural network weights and the random shuffling of the training data set before each epoch of training caused some local variations in CNN accuracy. The model used to classify sonar imagery for this thesis reported an accuracy of 95%, with an expected ± 2-3% variation caused by the aforementioned reasons.

Sammendrag

Denne oppgaven tar sikte på å utnytte en etablert bildeklassifikasjonsmetode for å lage en metode for klassifisering av kaldtvannskoraler i syntetisk aperture sonarbilder. Dette vil bli utført ved å kombinere høyoppløselige sonarbilder med et convolutional neural network. Resultatet blir en automatisk klassifikasjonsmodell som vil redusere behovet for menneskelig tolkning av sonarbildene. Metoden skal kunne brukes til kvantifisering av marine habitater i sonarbilder.

Et eksempelstudie ble utført med tre datasett fra Tautra-området i Trondheimfjorden. Datasettene er fra 2012, 2013 og 2017. De meste av sobarbildene ble brukt til å danne et datasett med underbilder på 100x100 piksler som ble brukt til å trene klassifikasjonsmodellen. Resten av bildene ble lagret for å verifisere klassifiseringspresisjonen. Den estimerte nøyaktigheten på verifiseringsbildene var henholdsvis 83%, 92% og 95% for henholdsvis 2012, 2013 og 2017.

En annet casestudie ble utført på sonarbilder fra Nord-Leksa, en annen region i Trondheimfjorden. Disse bildene ble brukt til å verifisere klassifiseringsprestasjonen på et annet område enn Tautra. Nøyaktigheten av verifikasjonsbildet fra Nord-Leksa var 89 %.

Et estimat på korallforekomsten ble gjennomført for hvert datasett , men resultatene kunne ikke sammenlignes fordi totalt areal undersøkt for hvert år varierte. Den estimerte koralldekning var 0.8 km^2, 0.4 km^2 og 0.7 km^2 for datasettene fra 2012, 2013 og 2017 på Tautra. De klassifiserte sonarmosaikkene viste et tett område med koraller på steder hvor korallrev tidligere har blitt verifisert. Forekomsten av koraller på Nord-Leksa ble anslått til $2.3km^2$, omtrent 50% av det totale arealet som ble undersøkt. Det var likheter mellom det klassifikasjonsmodellen beregnet til å være korall og det som er rapportert for Nord-Leksa.

To feilkilder i metoden ble identifisert. Den tilfeldige initialiseringen av de nevrale netterkvektene og den tilfeldige blandingen av treningsdatasettet før hver epoke av trening, forårsaket noen lokale variasjoner i modellens rapporterte nøyaktighet. Modellen rapporterte en nøyaktighet på 95 %, med en forventet variasjon på ± 2-3 %, som er forårsaket av de overnevnte grunnene.

Preface

This master's thesis is the final part of the study program Marine Technology, in the specialization Underwater Technology at the Department of Marine Technology within the Faculty of Engineering Science and Technology. The work was carried out in the spring semester of 2018. The topic was introduced to me by Martin Ludvigsen. I already had an interest in sonar technology and computer programming, and mapping of marine habitats in sonar imagery seemed like an interesting case for a combination of these interests. The work has been very challenging because of the complexity of the classification model and the data processing, but it has also been very rewarding.

Trondheim, July 6th, 2018

Margrete Sandsbråten Scheide

Contents

List of Tables

List of Figures

Listings

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
BN	Batch Normalization
BS	Backscatter Strength
CNN	Convolutional Neural Network
CTD	Conductivity, temperature and depth
FC	Fully Connected
GPU	Graphics Processing Unit
MBES	Multibeam Echosounder
ReLU	Rectified Linear Unit
RT	Random Forest
SAS	Synthetic Aperture Sonar
SE	Structuring Element
SGD	Stochastic Gradient Descent

SOM	Self Organizing Map
SSS	Side Scan Sonar
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Background

Mapping of marine habitats is important for both scientific and industrial purposes. Coral reefs are known to be rich in biodiversity, as they provide shelter and feeding grounds for many commercially important species of fish [9]. The cold-water corals can form large reefs in depths from a few tens of meters up to several thousand meters, most often determined by a temperature between 4 °C and 12 °C [10]. Some important threats to the cold-water corals are deep water trawling, digging on the seabed, pollution and ocean acidification [10, 11]. Deep water trawling is already known and widespread, and efforts to prevent it has been initialized by the government in Norway. Industrial changes of the seabed and pollution can cause particle containment of the water, which can deposit on the corals. Ocean acidification and its effect on deep water corals is a less understood threat [10]. However, the need for knowledge of marine habitat location and size is important for preservation and further investigation of environmental effects.

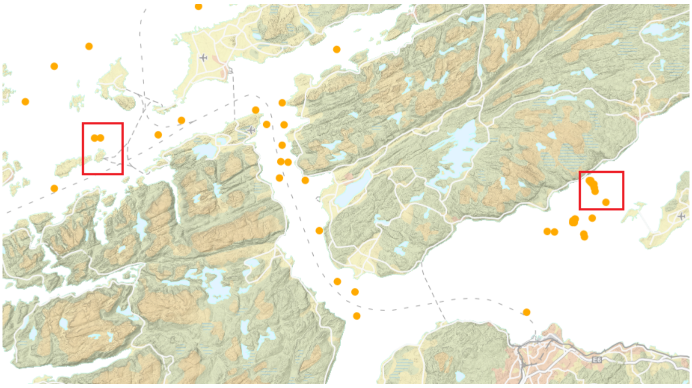
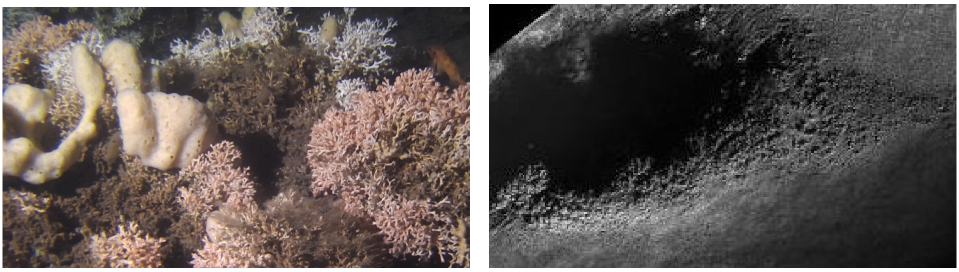


Figure 1.1: Map showing the Nord-Leksa coral reef in the left red rectangle, and the Tautra reef to the right.The map is collected from the Mareano database, showing the features of the Trondheim Fjord, including verified coral reefs in the Tautra area (orange dots) [1].

1.1.1 The Mareano Project

The Mareano (Marine AREA database for NOrwegian coast and sea areas) project is a government funded project for seabed mapping in Norway. The Institute of Marine Research, the Geological Survey of Norway and the Norwegian Mapping Authority are responsible for the scientific activities of the project. They have investigated the Trondheim Fjord, and confirmed the presence of *Lophelia pertusa* in the Tautra field as illustrated in Figure 1.1, which is a map collected from their website [1]. The orange dots in the figure illustrates the confirmed presence of coral reefs. The *Lophelia pertusa* is a scleratinian coral, also known as stony coral [12]. They are known to form large coral reefs, which supports highly diverse ecosystems [11].

The corals in the Trondheim fjord have been thoroughly mapped due to their proximity and availability for biologists, geologists and engineers with connection to the Norwegian University of Science and Technology (NTNU). Tautra has been a test site for new mapping technology. Through field trips in 2012, 2013 and 2017 the Tautra field was mapped using the synthetic aperture sonar HiSAS from Kongsberg Maritime [13, 14, 15]. In 2016, ground-thruthing was performed using optical cameras on an ROV, which along with navigational data could be used to confirm corals in the sonar imagery. A screen shot from the ROV video is shown in Figure 1.2a, illustrating the shape and size of the Lophelia



(a) Screenshot from the optical camera on an ROV covering part of the coral reef on Tautra in 2012. The image shows live *Lophelia pertusa*, sponges and a small fish in the background. The image also illustrates the general round shape of each coral colony. The image was collected during the field trip in 2012 []

(b) Example of coral reef visible in sonar image, collected in 2012 with the HiSAS sonar on HUGIN AUV. The coral reef is approximately 160 m long and 30 m wide. The image was created using the 2012 dataset and Reflection software.

Pertusa coral. Due to the complex three-dimensional shape of the corals, they are easily identified in sonar imagery. An example of how the corals are shown in the sonar imagery is presented in Figure 1.2b.

Nord-Leksa was covered with the HiSAS sonar in 2016 [15]. The field trip included a survey with an ROV that verified the presence of the *Lophelia Pertusa*, similar to the Tautra area. Nord-Leksa is another well known area where presence of coral reefs has been confirmed by the MAREANO project.

Even though both Tautra and Nord-Leksa are dominated by the *Lophelia pertusa* coral, other coral types and organisms grow in the same area. This is evident in Figure 1.2a, where sponges are visible. The classification program will not distinguish between types of corals or growth with similar characteristics. It will in addition not be able to detect if the corals are alive or not. *Lophelia pertusa* might be dead without losing its shape, and it is common for live corals to grow on top of dead corals [12].

1.2 Objectives

The objective of this thesis is to utilize a combination of convolution neural networks and the HiSAS system of Kongsberg to create a classification method for marine habitats. The program will automatically calculate the area of corals classified in the sonar imagery. The program is a contribution towards automatic interpretation of sonar imagery, which can relieve the need for human interpretation. Qualitative comparison of sonar imagery will be possible using the model, with a higher degree of repeatability than using human interpreters.

A method for sonar data processing and image analysis for the purpose of classifying cold water corals in HiSAS imagery will be presented. The convolutional neural network created for classification will be designed based on the current best practice available. The three hyper-parameters learning rate, batch size and units in fully connected layer will be optimized for increased classification performance. Additionally, the mathematical morphological image processing technique *Closing* will be applied to the classified imagery in order to remove anomalous classifications.

By using Reflection (trademark Kongsberg Maritime) for post-processing of the HiSAS sonar data, geo-referenced sonar mosaics will be available for classification. The classification method will be able to input geographically overlapping sonar mosaics and calculate the coral area coverage while taking the overlapping areas into consideration. Thus the program will output an estimate of total coral coverage for the entire area covered by the HiSAS sonar. This will contribute to making the mapping of marine habitats more efficient.

The classification odel will benefit everyone interested in mapping of marine habitats. This includes biologists, geologists and politicians, as well as the marine and offshore industry. One of the goals of the study is to develop a methodology that is uncomplicated and accessible for the diverse group of possible benefactors. If complex preprocessing of sonar data can be avoided and the accuracy of the classification tool is sufficient compared to other methods, the study will be considered successful. The target is to reach 85 %

accuracy on the classification scheme. This will be evaluated using *F-score*, which is calculated based on *precision* and *recall*.

Two areas in the Trondheim fjord will be used as test sites for the program. Three sonar datasets from Tautra will be used to create a training dataset, while one dataset from Nord-Leksa will be used to evaluate the performance of the program on completely unknown data. The classification and quantification of the complete datasets will be presented in a map created in QGIS.

A project thesis was conducted during the Autumn semester of 2017. This was meant as a preliminary work for this master thesis. Some content of the project thesis has been used directly, or with slight modifications, and presented as part of this master thesis for continuity purposes.

1.2.1 Previous Work on Classification of Coral Reefs

One of the goals in this project is to create a user friendly classification program that can be used by anyone with access to SAS imagery of the seabed. This will first and foremost be used in search of corals, as it will not be trained on other types of seabed characteristics. The Geological Survey of Norway are using a different method for a similar type of classification. They utilize bathymetry data and the machine learning method *Random Forest* to create maps of areas with a high likelihood of coral reefs [16]. The resolution of their survey is 5m, while the sonar imagery used in this project has a resolution of 4 cm. The large difference in resolution indicates their difference in potential. SAS imagery classification might have a better quantification property due to the high resolution. It may also make change detection more plausible as corals grow around 5-10 mm per year [17].

1.2.2 Why Choose a Convolutional Neural Network for Classification

For the purpose of classifying corals in the sonar data, the training of a convolutional neural network and using that to classify images was chosen as the method. If the CNN is properly trained it will be relatively simple to use it as a classifier on mosaics from SAS-systems. Other neural networks were not considered, as previous studies have had varied results [18]. A CNN requires less complex pre-processing of the sonar data, as the mosaics are used instead of raw data from backscatter and bathymetry. The choice is also inspired by the textural analysis performed in [19], as a convolutional neural network will perform a textural analysis of the images during the processing and learning. If the CNN performs well, it will make classification of sonar imagery very accessible for various scientists and industries.

The CNN will be trained in recognizing corals in gray-scale imagery. The method was inspired by other successful image classification regimes made by trained CNNs [20, 21, 22, 23]. Because a CNN has the ability to extract features autonomously, it will be a very efficient way of classifying images if it proves successful.

1.3 Structure of the Thesis

Chapter 2 presents the theory behind synthetic aperture sonars, followed by a study of image artefacts that may appear in sonar imagery.

Chapter 3 discusses how filters can be used for image processing, as well as the morphological operation *closing*.Additionally, a presentation of a classification method using texture analysis will be presented.

Chapter 4 presents a general overview of the background of convolutional neural networks. The definitions of artificial intelligence, machine learning and deep learning will be presented, followed by a study of convolutional neural networks and the relevant hyper-parameters.

Chapter 5 describes the method and approach used to create a classification method. Data processing from sonar raw data into georeferenced sonar mosaics will be described, followed by data labelling as a preliminary stage towards training a convolutional neural network. A basic network structure will be created and optimized, using grid search and random search to optimize the hyper-parameters. The performance metrics used to evaluate the classification accuracy will be described, followed by a presentation of how the method will be used to find the coral coverage in each of the four datasets.

Chapter 6 presents the results. A verfication dataset will be used to estimate the F-score of the four datasets. This was calculated before and after using *closing* to remove anomalous classifications. Furthermore, an estimate of the total coral coverage in each dataset will be calculated and presented. Finally a comparison of a certain coral mound will be performed, using imagery from 2012, 2013 and 2017 at Tautra.

Chapter 7 is a discussion of the results and the method. How the method performs compared to the goals will be discussed, as well as the sources of error and problems that was encountered during data processing and training. The effect of *closing* will be studied for each of the four verification images. The quantification accuracy of the network will be discussed, with reference to the results of the comparison of the same region covered in each dataset from Tautra.

Chapter 8 concludes the work of this thesis, and is followed by Chapter 9 which includes recommendations for further work.

The Appendix includes some additional imagery related to the method and results of the thesis.

Chapter 2

Sonar Imagery

The following sections are collected from the project thesis by the author of this report [24]. Some minor modifications have been made. It is included for continuity purposes for this report.

2.1 Sound Navigation and Ranging

Sonar is a sensor for underwater detection and location of objects using sound signals. The acronym SONAR stands for **SO**und **N**avigation **A**nd **R**anging, and refers to its ability to use sound signals as a measurement technique [25]. Active sonars can calculate distances to targets and the intensity of the backscattered signal. There are many types of sonars available today, with different configurations and signal processing techniques. Up until recent years, the most common sonars in use for seabed mapping purposes have been the multibeam echosounder (MBES) and the sidescan sonar (SSS) [26]. Due to technological advancements in the last decade, another type of sonar called synthetic aperture sonars (SAS) has become available for commercial use [4]. These are well suited for seabed mapping and are the main sonars in focus of this report.

The concept of SAS followed the development of synthetic aperture radars in the 1950s. However, it took half a century before the concept evolved into a functioning sonar [26]. The SAS system distinguished itself from SSS and MBES by having higher resolutions for lower frequencies. This is possible because the array of transducers are synthetically longer on a SAS, thus increasing the resolution.

Sonars create images based on the recorded backscatter data. A transmitter is used to send out an acoustic signal, also known as a ping, with a specific frequency, pulse length and bandwidth. The echo of the transmitted signal is reflected from the seabed or object dependent on the characteristics, e.g. the sediment type, object size, shape and density as well as the bathymetry of the seafloor. The recorded echo is a measure of intensity, which can be used to classify the nature of the seabed or objects [2]. The reflected signals can also be used to estimate the bathymetry and water depth.

The measured intensity, or reflectivity, is dependent on many parameters and is the most complicated part of sonar signal processing. Acoustic and geophysical processes will affect the signal as it progresses from the sonar and through the water. Acoustic parameters like propagation loss due to the environment must be accounted for, as well as the particular acquisition geometry of the sonar. The reflectivity is a measure of backscattered energy and will depend on the roughness, density and shape of the seabed [2]. The transmission level, the reception sensitivity, beam aperture and signal duration are sonar parameters which will affect the observable intensity of the received echo. Also, the target reflectivity must be estimated separately.

2.2 Elements of a Sonar that Affects the Image

2.2.1 Angular Dependence

The angular dependence of backscatter data can be utilized to determine the nature of the seafloor, as well as being a constraint in the form of disturbance in the graphical representation.

If the seabed has high roughness and hardness, it is typical that the sound waves are reflected homogeneously in all directions, and the echo level will not depend on incidence angle. A soft and flat fluid-like sediment will result in a mirroring effect, where the intensity is maximum at the vertical line and minimal at oblique angles, giving a rapid decrease in strength on each side of the sonar image [2]. This dependency is illustrated in Figure 2.1. To make a flat seafloor appear so in graphical representation, it is sometimes necessary to compensate for the angular dependence of the signal during the processing. As a result, the angular dependence will provide information both on the characteristics and bathymetry of the seafloor.

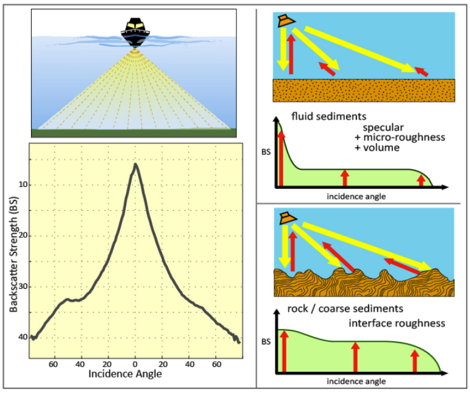


Figure 2.1: Illustration of how the backscatter intensity may vary with incidence angle, depending on the material of the seafloor, for a MBES. Courtesy of [2].

2.2.2 Spatial Facies and Mapping

Spatial facies are areas with the same acoustic responses and measurable characteristics [2], which can be used for indication and basis of habitat maps. It may not be accurate enough to be used for classification, as acoustic signals cannot measure the habitats directly. Habitat maps in accordance with geoscience and biology criteria must, however, use other tools to verify the nature of the seafloor [2].

2.2.3 Processing of Sonar Imagery

Mosaics can be defined as ”geo-referenced, gray-scale image of seabed reflectivity” [2]. What this means is that the images contain information on location as well as the backscatter information of the survey. The color scale used in this thesis is the same as the norm defined in [2], where white is the highest intensity, and black is the lowest.

2.2.4 Angular Response

Angular response is defined as the ”backscatter strength (BS) as a function of angle of incidence (θ) with the seafloor” [2]. As a consequence of the angular dependency, visual interpretation and image processing algorithms might be hindered by the strong along-track banding that occurs in a geo-referenced representation of BS. However, if the accuracy is sufficient, the BS(θ) has the potential to be utilized for seabed classification and characterization, as the seafloor type will give distinctive responses of BS. This is dependent on a very accurate bathymetry, and depending on what the data is used for might require physical sampling of the seabed. The primary challenge with this approach is to be able to recognize the acoustic signature of a particular type of seabed. The theoretical models behind BS use a large number of parameters, which can be difficult to obtain. Correction for angular dependency can improve the imagery as illustrated in Figure 2.2.

The processing of the image is strongly dependent on the assumption of the bathymetry

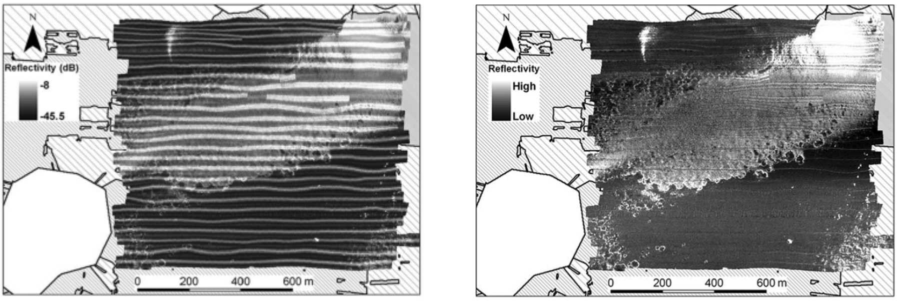


Figure 2.2: Illustration of how correction of angular dependency can affect the image quality. Assumptions were made regarding spatial homogeneity of the seafloor, which improved the image although some residual artifacts are still visible. Courtesy of [3] and [2].

of the seabed. By using adaptive compensation for angular dependency in large seabed mappings, the number of artifacts will strongly reduce in comparison to processing done with one assumption for the entire seabed, as illustrated in Figure 2.2. However, in the transition areas, there will usually be a visible disturbance. There is no standardization for how angular dependency correction should be performed. A number of methods are presented in Chapter 6 of [2]. The choice of method depends on the use, i.e. if there is a biologist, geologist or acoustician that will create and utilize the imagery.

Examples of corrections that can be done in the post-processing of mosaics are de-speckling, anti-aliasing and filtering (low-pass) [2, 6]. This will improve the pixel quality and hopefully remove some of the overlapping lines. Different techniques are used for different purposes. If the user wants to produce appealing visual imagery it is possible to omit a lot of steps in the processing, ending up with a meaningless BS level. If the purpose is classification or characterization, it is more important to restore a correct BS level by using information on gain, source level and beam pattern. The former is regarded as a *qualitative product*, while the latter is regarded *quantitative* [2].

2.3 Synthetic Aperture Sonar Imagery

The SAS system is to some degree a combination of the MBES and the SSS [4], as illustrated in Figure 2.3. This is because it utilizes the MBES ability to scan a sector, at the same time as several pings are used to insonify the same area. An SSS can use only one element with one beam to perform the imaging, while the MBES uses several elements in an array to scan a sector. The SAS has several elements that scan a sector, but it also includes information from several pings in a virtual array, hence increasing the resolution. Each pixel in an image is constructed with data from several pings [27].

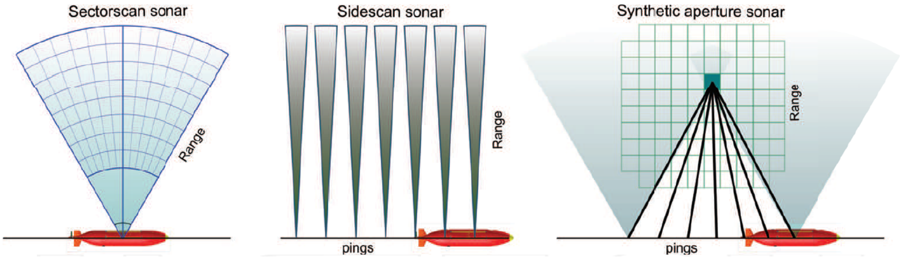


Figure 2.3: Illustration of how a SAS can be considered as a combination of MBES and SSS technology, courtesy of [4].

2.3.1 SAS Sampling and Coverage Rate

A essential part of the data processing for sonars is a well-defined sampling rate. Sampling is, in this case, the distance between the elements of the receiver array and their directivity. Alias lobes occurs if the sampling rate is too small, creating distorted information which affects the image quality. An array should be sampled with $D \leq d/2$ in order to prevent alias lobes, where D is the length of the virtual array and d is the receiver element size [4].

The multi-element receiver array in SAS consists of transmitter-receiver pairs that are moved along-track and emit and receive pulses repeatedly, forming a synthetic antenna [4]. These pairs can be combined in a virtual array, which is half the length of the receiver array and consists of combined transmitter-receiver elements. This is called phase center approximation [28]. The receiver and transmitter field of view is what defines the maximum length that can be achieved for the synthetic aperture, and hence the size of the receiver and transmitter is what determines the resolution [27].

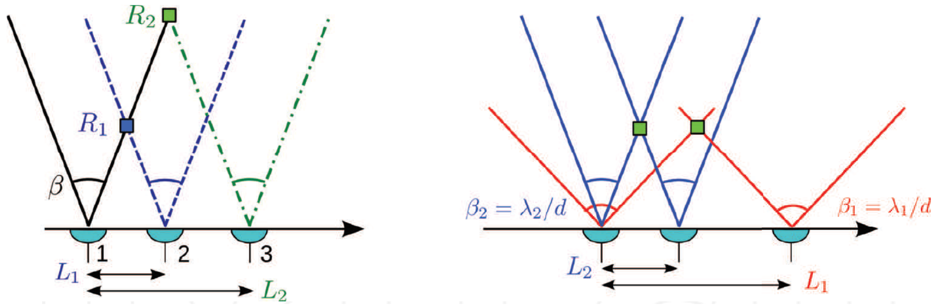


Figure 2.4: Illustration of how the SAS configuration has an along-track resolution that is independent of range (left) and frequency (right), courtesy of [4]

Along-track resolution for a SAS system is solely given by the element size and is independent of range and frequency. This is illustrated in Figure 2.4. To the left, the range independence can be explained by considering the length of the aperture. The length of the aperture for the range R_1 is L_1, while for the range R_2 it is the larger length L_2. Hence the length of the synthetic aperture is what defines the maximum range for a given along-track resolution defined by Equation (2.1) [4].

$$\delta x = \frac{d}{2} \tag{2.1}$$

The image quality of the SAS imagery is defined by the range because of the signal to noise ratio, as longer range introduces more noise to the received signal. However, the resolution is the same, and hence geometrical aspects of targets are the same within the entire swath [27]. This is useful for classification purposes.

The SAS platform also limits the image quality, and even though the theory has been known for decades, the appropriate platform has not been available until the AUV technology improved at the beginning of the 21st century. The most significant challenge to overcome was the navigation properties of the platform. The image quality of a SAS sonar depends heavily on accurate positioning and stable movement [4].

2.3.2 SAS Imagery Challenges

Some of the most important challenges for SAS imagery are listed in [4], and how they affect the imagery in different ways will be presented in the following.

The sampling rate must be sufficiently large in order to prevent aliasing effects in the sonar image. The sampling rate is dependent on how the platform and sensor operate during the data collection and provides an upper limit of range for a given speed [29].

The image quality is highly dependent on accurate navigation and in order to meet the requirements SAS technology can include micronavigaiton as a part of their system [4]. This is performed by overlapping the virtual array (PCA) with the next array, a method known as displaced phase center antenna (DPCA) [28]. This method causes a trade-off between navigation performance (image quality) and area coverage rate.

Furthermore, if the sensor platform is moving in a non-straight line, the synthetic aperture will be non-straight, and hence the image geometry will be affected by the seabed topography [4]. By knowing the bathymetry and position of the platform prior to mapping, e.g. by using an interferometric sonar to map the area in advance, the SAS processing can be corrected for the non-straight track. Problems will also occur if the platform runs in a straight line, but with a non-straight heading, e.g. due to currents. Non-linear effects will occur for the synthetic aperture due to the change in heading, and the processing becomes dependent on the topography [4].

The speed of sound has to be calculated continuously and accurately, in order to prevent de-focusing in the image [4]. The speed of sound can vary with depth and horizontal distance, and hence a CTD is usually installed on the platform for continuous monitoring of the environment [25]. Multipath is another sound-related problem that might occur in shallow waters, as several sound rays might be reflected off the surface before reaching the target [25]. This can affect the temporal coherence of pings in the micronavigation, it can reduce the spatial coherence in interferometry and add unwanted signals to the SAS images by removing contrast in shadows and loss of fidelity [4].

2.4 Previous Work Using SSS and MBES

2.4.1 Classification Using Sidescan Sonar

A lot of seabed mapping and classification have up until now been performed using MBES and SSS. The classification methods used are often considering the texture of the seabed. In [19] a SSS was used to map cold-water corals at depths of 500-1200 m. Texture analysis based on co-occurrence matrices of the grayscale images proved to be the most efficient method for making a quantitative analysis of the corals. The mapping of the areas showed the presence of mounds, which were identified as coral banks [30]. Cold water corals are often situated on mounds and in areas exposed to a current [31].

Using several image processing techniques, the purpose of the study in [19] was to identify what acoustic textural features that could be identified for different kinds of mounds, and how that could be used to make a quantitative assessment of the appearance of mound formations. Distinct geo-morphological characteristics could be identified. The methodology included a visual interpretation of the dataset before selecting regions of interest. The tonal and textural properties of these regions were investigated. The tonal property is related to the amplitude of backscatter energy. As a part of the processing, quantification of textural measurements was performed using first order statistics and edge detection using image processing techniques was used.

2.4.2 Classification Using Multibeam Echosounder

As the signal processing of an MBES developed during the first decade of the 21st century, its popularity increased for use in seabed mapping. The possibility of combining bathymetry and backscatter data increased the along-track resolution that previously could not compete with the SSS resolution [32].

Many different classification methods have been developed, and [32] divides them into two main categories: geoacoustic and feature-based approaches. The former involves comparison of the backscatter waveform to known seabed types, defined by grain size, density, porosity etc. There are commercially available software that include seabed type signatures, and these can be used to evaluate the MBES data. Feature-based approaches involve studying features at either a local or regional level. The bathymetry alone can be used to classify a seabed type. The bathymetry can be evaluated by matching templates, spectral signatures and fractal analysis [32]. In addition, the backscatter data can be combined with the bathymetry in order to assess angular response, texture, Markov random fields etc. ([32] and references therein). The classification techniques are generally involving data cleaning, image enhancement and automated classification techniques [32].

Classification of MBES data using artificial neural networks was studied in [18]. They combined backscatter data based on angular response and compared it to the corresponding bathymetric data using a self-organizing map technique. The importance of pre-processing the data was highlighted, which involved removing data from the near-nadir region because the acoustic signal suffered from high sidelobe interference. Then normalized backscatter data and bathymetric beam-level data was used as input for the ANN. The type of ANN used is called a self organizing map (SOM). It has the ability to convert complex non-linear statistical relations into simple geometric relations, making it possible to downsize high-dimensional data [18, 33]. Other ANNs were compared to the SOM method in [18], and SOM provided an accuracy of 100% along with a competitive neural network approach, while learning vector quantisation and the ISODATA method gave poorer results.

Chapter 3

Image Processing Techniques

Parts of the following sections are collected from the project thesis by the author of this report [24]. They are included for continuity purposes of this report.

3.1 Image Processing Using Filters

Jocelyn Chanussot et al. [34] studied the effect of scalar image processing filters for the purpose of speckle reduction in 2002 [34]. They evaluated the effect of several filters in SAS images containing mines. The main effect of the filters presented were smoothing filters of various strengths.For some filters the smoothing weakened the meaningful backscatter, making it difficult to detect the mine. In others, the resolution decreased. Some filters, for example the adaptive weighted filter, was able to smooth the image and enhance edges simultaneously. This method can be used for image segmentation, which might be useful in classification of SAS images.

3.2 Closing

Closing is a morphological image processing technique where *dilation* and *erosion* are performed in succession [6]. The operation can close small holes inside a specific boundary area. A structuring element (SE) is a form of filter that is used to search an image for properties of interest [6]. They can be in shapes like circles, rectangles, crosses, pyramids etc. SEs are converted to rectangular arrays when working with images, meaning that background pixels are added to make a rectangular shape of the desired SE shape.

The basic morphological operations dilation and erosion are fundamental techniques used for many purposes in morphological image processing. A basic illustration is presented in Figure 3.1, showing how the structuring element formed like a cross will affect a rectangular image in erosion and dilation. Erosion includes all the pixels that are completely represented when placing the SE in that pixel. This is illustrated in the figure by the ·. The center of the original image is the only pixel that can completely represent the structuring element. Those pixels not encapsulated will be discarded in the eroded image. Dilation is the opposite, as the SE will be added to all the pixels in the original image. Erosion and dilation can be combined and utilized to *close* holes, by first dilating the image before eroding it. This ensures that the boundaries remains the same, but single missing points within the boundary can be filled. The processing depends on the structuring element, which can be specialized for the relevant scenario.

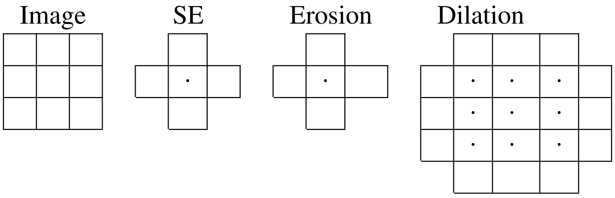


Figure 3.1: Illustration of erosion and dilation with a cross-shaped structuring element.

3.3 Textural Analysis

A textural analysis was performed by Philippe Blondel on SSS imagery in 2002. The analysis was performed using edge detection, Fourier transform and gray level co-occurrence matrices [19]. Edge detection is an image processing technique where a kernel is convolved over the image, calculating the second order derivative in all directions, which highlights rapid changes in intensities [6]. Edgeness per unit area was also measured in [19], providing information on the number of edges per area, indicating the roughness of the texture. The Fourier transform converts the image from spatial domain to frequency domain. The objective was to investigate if signature frequencies could be obtained from the corals. However, the results of the transform were highly affected by acoustic noise that made it difficult to analyze. However, if the noise is removed from the image in the spatial domain, it might be possible to use frequency analysis on sonar imagery.

The gray level co-occurrence matrix is another of the techniques that were investigated. It is a statistical method for texture analysis and can include information on the textural entropy and homogeneity of an image. In the study by Blondel, they had good results with this technique using 20×20 pixels windows moving over the image [19]. The co-occurrence matrix can be described by statistical indices previously defined, where ground-truthing has been performed along with the measurement of indices for verification [35, 36]. The entropy is a measure of lack of spatial organization within the kernel, outputting high values for rough texture. Homogeneity indicates the local similarities within the kernel. A high value of homogeneity refers to a uniform backscattered area. Software is developed within this field of sonar image processing [19].

3.4 Image processing using artificial intelligence

The use of artificial intelligence for sonar image classification has been discussed for many years [37, 18]. However, it is not until recent years that this technology has been commercially available, readily applicable and computationally feasible. Artificial intelligence is a class of systems that exploits computers ability to do symbol processing as well as number processing [38]. These systems are referred to as expert systems, where computers are capable of mimicking intelligent human behavior, which formed the basis of the term Artificial Intelligence (AI).

Chapter 4

Artificial Intelligence and Deep Learning

Artificial intelligence (AI) is a general term including many types of approaches in computer science that is inspired by human behavior [39]. Figure 4.1 illustrates how one can categorize different types of AI, as it mimics humans or rational thinking or behavior. Rationality is a measure of ideal performance and depicts how a computer performs in relation to a defined correct answer [40]. Rationality distinguishes from the human cate-

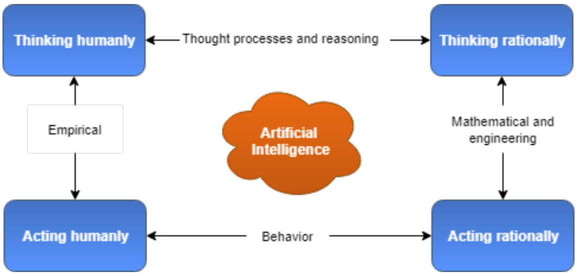


Figure 4.1: Illustration of the four main approaches of artificial intelligence, divided into two main categories: human and rational mimicking computer programs. They can have different aims, depending on their purpose. It is split into behavior versus thinking and reasoning.

gories, which includes the ”imperfectness” in human thinking and behavior. This thesis is concerned with the ”Acting Rationally” part of AI, as it aims for ideal performance based on rational agents. A rational agent means a computer program that autonomously acts towards a specific goal, and aims to make the best decision [40].

Machine learning is the part of AI that covers all the algorithms that can learn from a set of data, and predict or determine some form of answer. The term originates from Arthur Samuels in 1959, who created a machine that learned to play chess by itself [40]. After some time, it could beat opponents at a strong amateur level, which at this point proved that the machines could learn.

There are many types of machine learning algorithms, and they are often categorized into supervised, unsupervised and reinforcement learning. As the names reflect on, the difference between supervised and unsupervised learning is whether the input data is labelled or not. Regression and classification programs are dominated by supervised learning, while pattern recognition and descriptive modelling are typically unsupervised [40]. Reinforcement learning is based on punishment and reward, where the agent learns by recognizing why and when it did something right or wrong.

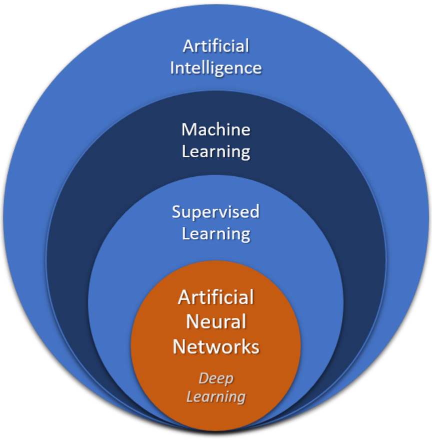


Figure 4.2: Overview of how artificial intelligence, machine learning, supervised learning and artificial neural networks (deep learning) are connected.

4.1 Supervised Machine Learning

For this project, the training dataset is labelled into categories. The goal is to make a classification program, and hence supervised machine learning is used. Three general categories of classification algorithms include linear, hierarchical and nonlinear models [41]. The choice of algorithm depends on the input data and how complex it is. Artificial neural networks (ANN), support vector machines (SVM) and random forest (RT) are popular algorithms used for classification.

ANNs are inspired by the central nervous system of mammals [39]. It is made up of layers, which in turn contains neurons. These neurons activate and send a signal if certain conditions are met. ANNs can be both supervised and unsupervised, meaning that the input can be labelled or not [40]. The neural network aims to generate a function that predicts the output based on the input data. If the ANN is unsupervised, it will search for patterns and try to create a function that fits the input data, e.g. autoencoders.

4.2 Deep Learning

Deep learning is another term for ANNs with more than 3-5 layers [39]. Even though the first deep learning programs were developed in the 1940s, the development of computers has been crucial for commercial and scientific use of neural networks [40]. The performance is mostly dependant on the amount of data used for training a neural network, hence the development of computer engines such as the GPU has been crucial for ANN implementation [42]. GPUs are very fast at matrix and multiplication operations and can decrease computational time by a factor of 50 compared to a CPU, depending on the engine [42]. This is the GPU is designed for handling graphic information in parallel, while the CPU is designed for sequential processes.

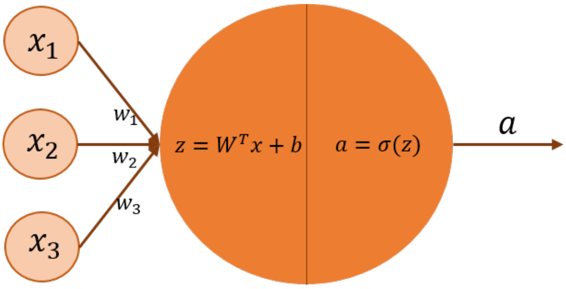


Figure 4.3: An illustration of a perceptron with three inputs that each will have a weight associated with it. The perceptron has an activation function σ and will output a scalar value **a**.

4.2.1 Neurons and Activation Functions

The simplest form of ANN is a perceptron and consists of only one neuron. The perceptron is illustrated in figure 4.3. The node will go through two steps in order to make an output. First, it will take a weighted sum of the input values using Equation (4.1), before it evaluates the sum with an activation function. The output will then be adapted to a desired range, e.g. zero and one for a binary classification scheme. There are many types of activation functions, allowing the range to include the entire range between zero and one, or to simply output one or the other depending on the weights. The former is known as a logistic activation function, while the latter is then known as a hard threshold. Two examples of activation functions are the Sigmoid and Rectified Linear Unit (ReLU) functions given by Equations (4.2) and (4.3). A graph for each and a threshold function is given in figure 4.4.

Weights and biases are randomly initialized for all ANNs. The weights will have values close to, but not actually, zero [38]. This is because the weights can have both positive and negative values as they evolve, and hence they should begin close to a neutral value. Due to the mathematical operations, the initial value cannot be zero. The biases represent constant terms of the function the neural networks aims to optimize, and are usually initialized to 1 or -1 [43].

$$a(x) = \begin{cases} 1, & \text{if } wx + b > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

$$\sigma_s(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

$$\sigma_r(x) = max(0, x) \tag{4.3}$$

$$\sigma_{soft}(x) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}}, \text{i = 0, 1 ... k} \tag{4.4}$$

The Sigmoid function is quite similar in shape to the threshold function, but due to the continuous shape it can allow the output of the neuron to output continuous values. ReLU also has this ability, but it will not allow activation of the neuron if the input is negative. ReLU has proven to be faster and perform better than the Sigmoid function in neural networks [42, 5]. In classification problems, it is common to include a Softmax activation function in the final layer, as it normalizes all output values between 0 and 1, hence providing a probabilistic output value. Softmax is given by Equation (4.4), and has a similar shape as the Sigmoid function. They differ in that the sum of all outputs from Softmax will be equal to 1, whereas Sigmoid outputs are independent.

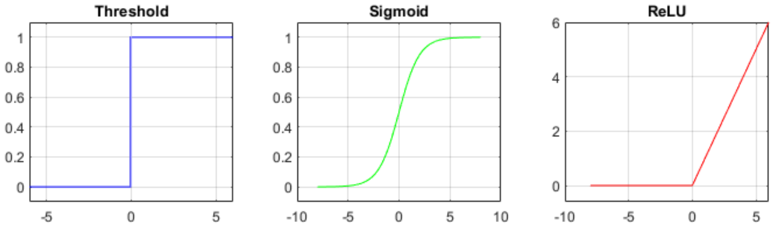


Figure 4.4: The first graph illustrates the threshold function (eq. 4.1, which outputs either 0 or 1. The middle graph is the Sigmoid function (eq. 4.2) which is a smoother version of the threshold, allowing all values between 0 and 1. The final graph is the ReLU function (eq. 4.3), which is 0 for all negative input values and linear for all others.

4.2.2 Convolutional Neural Networks

All ANNs consists of several neurons stacked in layers. The layers are characterized by feature maps, created by some mathematical operation performed on the input images. There are many types of ANNs, but for image classification the convolutional neural network (CNN) is the best suited as it can evaluate spatial information [5]. The performance is dependant on the dataset and the available computer engines [21]. The different layers and their abilities are discussed in detail in the following sections.

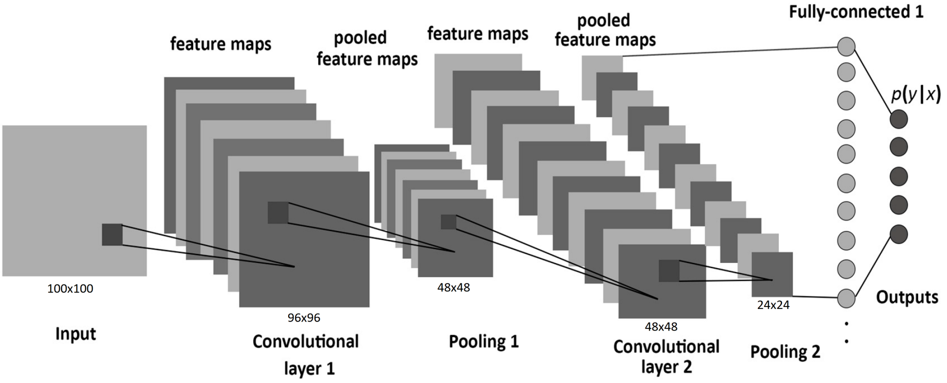


Figure 4.5: An example of how a CNN architecture can be designed. Each convolution layer creates several feature maps. The pooling layer follows the convolution layer, and down-samples the feature maps. The network ends with a fully connected layer that inputs information from all the neurons in the previous layer in order to create the output. Courtesy of [5]

Layers of a Convolutional Neural Network

A CNN typically contains at least one convolution layer followed by a pooling layer, before it ends with one or more fully connected layers with a Softmax activation to finalize the classification [5, 21, 23]. The convolution layer is where spatial information of the image is collected. In image processing, the convolution operation is associated with edge detection, sharpening and blurring of an image [6]. The convolution operation is performed by a matrix multiplication between an image and a particular kernel. The kernel is usually a 3x3 or 5x5 filter, which is convoluted with the image by sliding over each unique patch of the image. The operation is illustrated in figure 4.5, which demonstrates a typical CNN architecture with two convolution layers each followed by pooling operations and lastly fully connected layers.

The result of each convolution layer is stored in feature maps, where each spatial position is associated with the result of one convolution operation. The area of the input image that is connected to each spatial position is called receptive field, and will have the same size as the convolution filter. The number of feature maps in each convolution layer is determined by the number of different filters, where each filter will find different characteristics in the input image [5].

A classical example from image processing is the Sobel filter, which is an edge detector [6]. An example of how the kernel affects an image is shown in figure 4.6, which shows a sonar image of coral of size 100x100 pixels, and how its horizontal and vertical edges is detected by a Sobel filter. The horizontal Sobel filter is given in Equation (4.5), the vertical Sobel is the transposed of the horizontal Sobel.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.5}$$

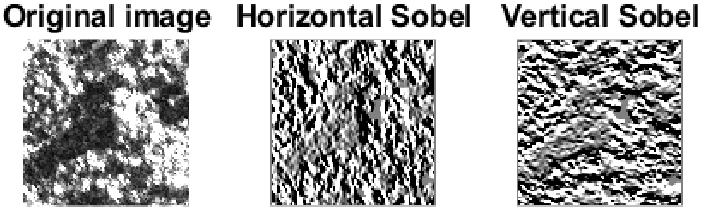


Figure 4.6: Illustration of how a horizontal and vertical sobel ([6]) will affect an image of a coral. The image is a 100x100 pixel HiSAS image collected in the Tautra area in 2012. The horizontal Sobel filter will detect the vertical edges of the coral, while the vertical Sobel filter detects the horisontal edges. The filter is convolved over the input image by performing a matrix operation with the filter over each unique section of the same size of the filter.

The filters are created randomly by the network itself and are a part of the learning parameters [21]. The size of the feature map is determined by the size of the input image (HxH), the size of the filter (FxF), image padding (P) and stride (S), i.e. how far the filter is moved between each convolution [5]. Usually, the padding is set to zero and the stride is 1. The feature map width (W) is then given by Equation (4.6). If the input image is 100x100 pixels and the filter is 5x5, the resulting feature map will have 96x96 spatial positions or neurons.

$$W = \frac{H - F}{S} + 1 \tag{4.6}$$

The number of neurons in the convolution layer is determined by the number of feature maps and their size. Each spatial position in the feature maps is a neuron connected to a certain receptive field of the former layer [43, 5]. Instead of each neuron in every feature map having its weights and biases, each feature maps shares weights. This concept is called parameter sharing, and it reduces the computational cost of the network [40]. This works because it is the filter type and what characteristics it activates on that matters, and not the specific receptive field it belongs to. Thus a convolution network is invariant to the rotation or place of a certain characteristic it activates on. The number of weights is greatly reduced, leading to a reduction of the computational cost which again increases the complexity of the network that can be trained [43].

The result of the convolution layer is that the neurons in each feature map will activate on certain characteristics of the input image. The number of neurons in the convolution layer in our example with figure 4.5 gives 7 feature maps, which equals $5 \times 5 \times 7 = 175$ neurons. Each neuron has $5 \times 5 = 25$ weights and 1 bias due to the filter size. This assumes grayscale images. Given color images, there would be three times as many weights. As a result, the first convolution layer in figure 4.5 has $175 + 7 = 182$ trainable parameters. In the next convolution layer, the channel size has increased to become the number of filters from the previous layer. Hence the number of parameters in the second convolution layer is $(5 \times 5 \times 7 + 1) \times 12 = 2112$.

After each convolution layer, a pooling layer is introduced to downsample the feature maps. The most used pooling operation is called max-pooling, where the maximum value of a predefined section of the image is outputted into the next layer [20]. The most common filter size of max-pooling is 2x2, which decreases feature map size by half. The number of feature maps remains constant, which allows for an increase in the number of filters without increasing the computational cost. The general concept is that adding more convolution layers will extract more complex features in the deeper layers [21]. Max-pooling is vital for this objective because it decreases the computational cost.

The final layer of a CNN is called a fully connected (FC) layer. They connect all the feature maps and neurons of the previous layer and identifies patterns [21]. It is common to have two FC layers in sequence, as this is a universal classifier, followed by the softmax activation in Equation 4.4. The neurons in FC layer aims to find the global features of the input. The number of neurons in the first FC layer defines the generalization capacity of the network [21]. This number has to be large enough to cover the necessary features that are needed to classify the images [43]. The final FC layer will have one neuron per class, and collect the class scores that is outputted from the softmax activation.

Epochs are the number of times the training data is run through the network model. A batch size is the number of images in a batch that is exposed to the network at the same time. One iteration is one run through of a batch.

Challenges for a Convolutional Neural Network

One of the most critical aspects of training a robust neural network is to have sufficient amounts of data [21]. Collecting data depends on the categories and availability of sources. If the available amount is limited, image augmentation can be considered as an option to artificially increase the data size.

Another challenge of creating a CNN is to ensure that there is no overfitting of the data. Overfitting means that the networks have been trained too specifically on the training data, and not generalized well [22]. If the accuracy of the validation is much lower than the

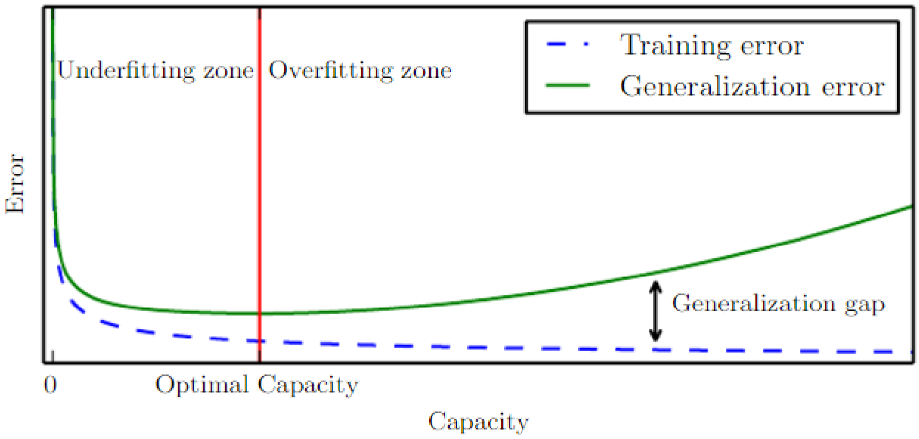


Figure 4.7: Relationship between capacity and generalization error. The curve shows where the model is prone to underfitting and overfitting. Courtesy of [7].

training accuracy, it is a sign that the network is overfitted. The problem can occur because of too many free and trainable parameters in the network [8, 22]. An illustration of how overfitting can be visualized is shown in Figure 4.7. The green line is the generalization error, or validation error, which grows during training if there is overtraining. Underfitting is another problem, occurring in the early stages of training. The generalization capacity of the network is poor in both conditions. The optimal network is created somewhere between the under- and over-fitting region.

A method for preventing overfitting is called batch normalization (BN). This technique re-normalizes the activations by a unit Gaussian distribution [22]. It de-correlates the data, and lead to centering and re-scaling of the image or feature maps of the previous layer. As the weights change during training, the distribution changes as well. The batch normalization makes the layer learn more efficiently because the data is normalized between each layer. BN acts as a regularizer of the network and contributes to its optimization [44]. The method also makes the network more robust against badly initialized values [43, 22]. BN is placed between the linear and nonlinear layers, i.e. between the convolution layer and the activation function (e.g. ReLU) [22].

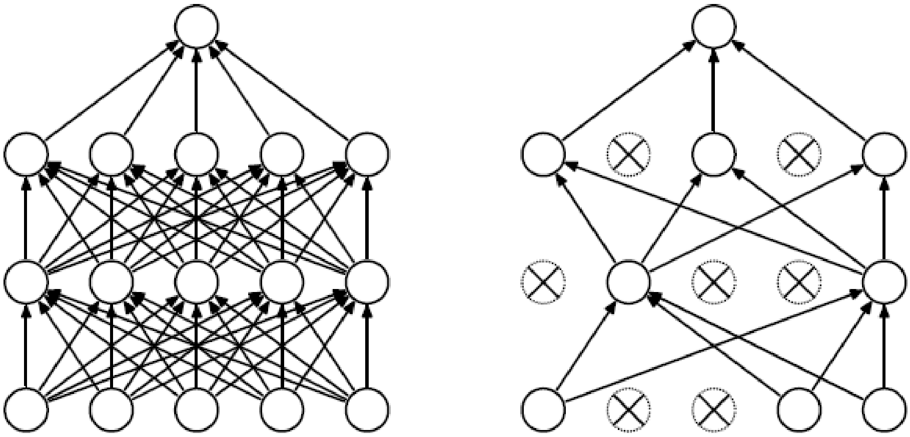


Figure 4.8: Dropout is when a portion of the neurons in a network are neutralized, i.e. they are removed from the training. This leads to more noise for the network, but it forces it to generalize better. Courtesy of [8].

Dropout is another technique for preventing overfitting, and its effect is sometimes confused with BN. Dropout ”kills” a portion of the neurons in each layer [8]. This is a form of adding noise to the network, thus forcing the network to generalize better [8, 44]. The difference between BN and Dropout is best explained through variance shift, as they affect the neural variances inconsistently when applied together. To benefit from both methods, dropout could be applied after all BN layers [44]. The dropout technique is illustrated in figure 4.8.

Input Data

To train a neural network, three sets of data are required. The original data containing all images and labels are split into a large training set, and two smaller datasets for validation and testing [40]. The training data is used by the neural network to learn, by updating weights and biases. After each iteration and epoch, validation accuracy and validation loss are calculated using the validation dataset. This information can provide useful information about the training process, and can be used to tune the parameters that are not trained by the network, e.g. network architecture. The third dataset is withheld from the network until the final stage of the training, where it is used to calculate the final accuracy. By convention, this dataset is called test data [40]. This ensures an unbiased accuracy and loss score of the neural network.

4.2.3 Learning and Optimization

In order to learn, the network needs an objective. The objective function is commonly called the loss function, which attempts to minimize the distance between the predicted and correct value [40]. The loss minimization function updates the weights after each iteration, according to the result of the loss function [39]. There are many types of loss functions, depending on what type of output the networks is to produce. Cross-entropy is a logarithmic loss function, and if there are more than two categories to be classified, categorical cross-entropy is used [39]. The function is stated in Equation (4.7), where the loss L_i is given by the prediction p_{ij} and the target t_{ij}. The categorical cross-entropy is used with the softmax activation in the final stage, of the CNN [39].

$$L_i = -\sum_j t_{ij} log(p_{ij}) \tag{4.7}$$

Mini-batch Gradient Descent and Learning Rate

Gradient descent is the technique used for optimization of the loss function, i.e. minimization of the distance between the predicted and real value. This is generally regarded as the best optimization method for neural networks [5, 7]. The gradient of the loss function is together with the learning rate used to update the weights [7]. If θ^t is the neuron weight at step t, ϵ is the learning rate and L is the loss function, the relation between them is given by Equation (4.8). The gradient is updated after each iteration. In the optimization algorithm, the most difficult part is to avoid local minimums that are not a global minimum. Saddle points can also cause difficulties. When the derivative reaches zero, the next step in the gradient descent is compromised, and the iteration stops [7]. Therefore it is important to ensure that it reaches the global minimum and not just a local minimum. This is controlled by the learning rate.

$$\theta^t = \theta^{t-1} - \epsilon^t \frac{\delta L}{\delta \theta} \tag{4.8}$$

If the learning rate is too high, the network can have problems converging [7]. The loss function should decrease with the number of epochs if the learning rate is well-defined. Often a learning rate decay is introduced to ensure that the loss function converges. If the learning rate is too low, the computation is slower, and convergence will take more time [7]. Several optimizing functions for CNNs are available, and the most popular ones are Adam, SGD, RMSprop and Adadelta [7].

Several other hyperparameters can be used and tuned for better learning. These include learning rate momentum, weight decay, structure-dependent learning rates, cyclic learning rates etc. [21, 45]. These will not be described in detail or used to optimize the CNN used in this project due to the complexity of the optimization.

Mini-batch gradient descent is a type of gradient descent where small batches of the total training data are presented to the CNN during training, and where the model and weights are updated after the batch is processes. This is a compromise between the Stochastic Gradient Descent (SGD) technique, which updates the weights and model after each example image is processed, and batch gradient descent, which updates after the entire training set is processed. The mini-batch technique utilizes the best of both worlds, by having lower computational time than batch descent, while at the same time reducing the risk of noisy gradient signals from the SGD.

Chapter 5

Method

The different aspects of the proposed method used in this thesis are presented in Figure 5.1. The first part is about collecting and pre-processing the sonar data into sonar imagery of high resolution. Then follows the description of how the labelling of training data was performed, before designing and training a CNN is described. After optimization of the neural network, the high resolution sonar imagery was classified using the trained CNN. A basic image processing technique called *closing* was also tested as a method for improving the spatial coherence of the classification. How the the results of both CNN classification and post-processing with *closing* will be presented. Finally there will be a description of how the calculation of total coral coverage area for all datasets. The stages will be described more in detail in the following sections.

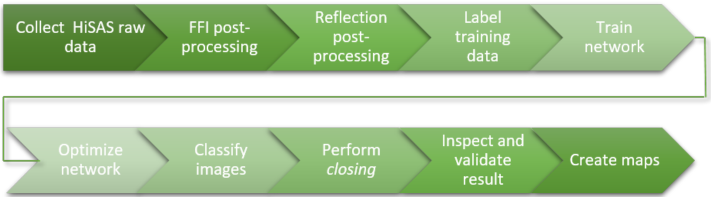


Figure 5.1: Work flow of the project, from collection of navigation data and sonar imagery from the AUV, through different stages of processing, training a network and at the end creating maps and images for inspections and validation of results.

5.1 Data Acquisition

Three surveys with their associated data were used to form the training and test data for the CNN. The area covered by the sonar is the Tautra Ridge in the Trondheim fjord. The area covered is approximately 2.7 km^2, illustrated in a 33x33cm resolution mosaic of the area in Figure 5.2. There were separate field trips in 2012, 2013 and 2017 that collected the sonar data from Tautra. The Kongsberg AUV Hugin 1000 was used with the HiSAS 1030 synthetic aperture sonar system. The trips were performed as a collaboration between NTNU and the Norwegian Defence Research Establishment (FFI) [13, 15]. The raw data was processed by FFI before it was provided to NTNU AUR-LAB (Applied Underwater Robotics Laboratory). As a part of this thesis, the data was processed in hte Kongsberg program Reflection, where geo-referenced sonar mosaics were made. The Tautra datasets were divided into training data and verification data.

A dataset of an area from Nordleksa was made available by NTNU AUR-LAB, which was used to validate the performance of the final classification model. The data from Nordleksa was not used for training, and is therefore considered to represent the actual performance of the model more accurately than the verification data from Tautra. Even though the latter was not used for training, the lawn mover pattern that the sonar data is collected in leads to overlapping imagery. Thus, there is a chance that the network has trained on images from the same coral mounds as the ones in the verification set. Therefore, the model results on the Nord-Leksa will be the most representative performance results.

The ocean basin outside the Nord-Leksa area is another well established and tested coral cite [1, 9]. The data was collected in 2016, with the same equipment and general setup as used at Tautra [46]. A mosaic with a resolution of 33x33 cm covering the entire area of 2.9 km^2 is presented in Figure 5.3.

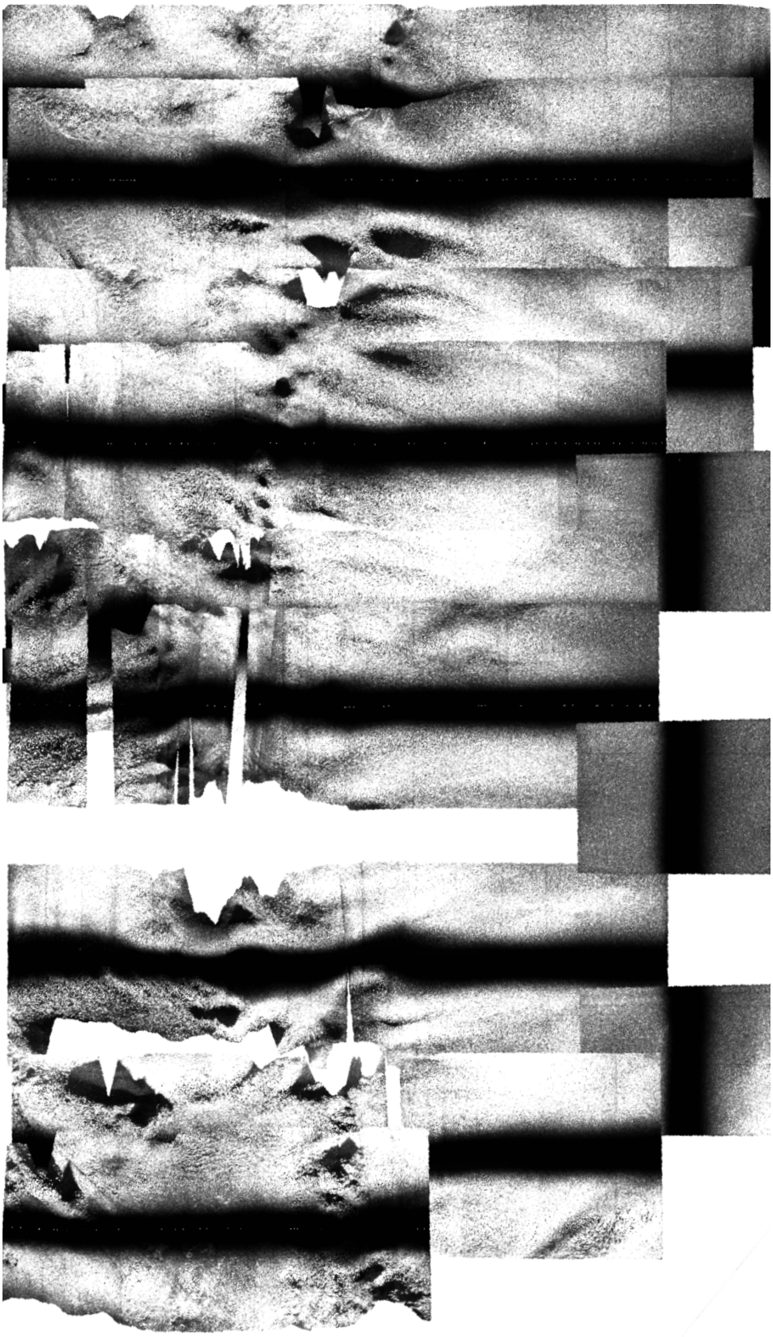


Figure 5.2: Sonar mosaic of the Tautra coral reef. The resolution is 33x33 cm, covering an area of 1260x2100 $m \approx 2.7\ km^2$. The image was created with Reflection, using the "last" mapping option, as this resulted in the best visual image. "Last" creates the mosaic by adding the files chronologically, without mixing the overlapping areas.

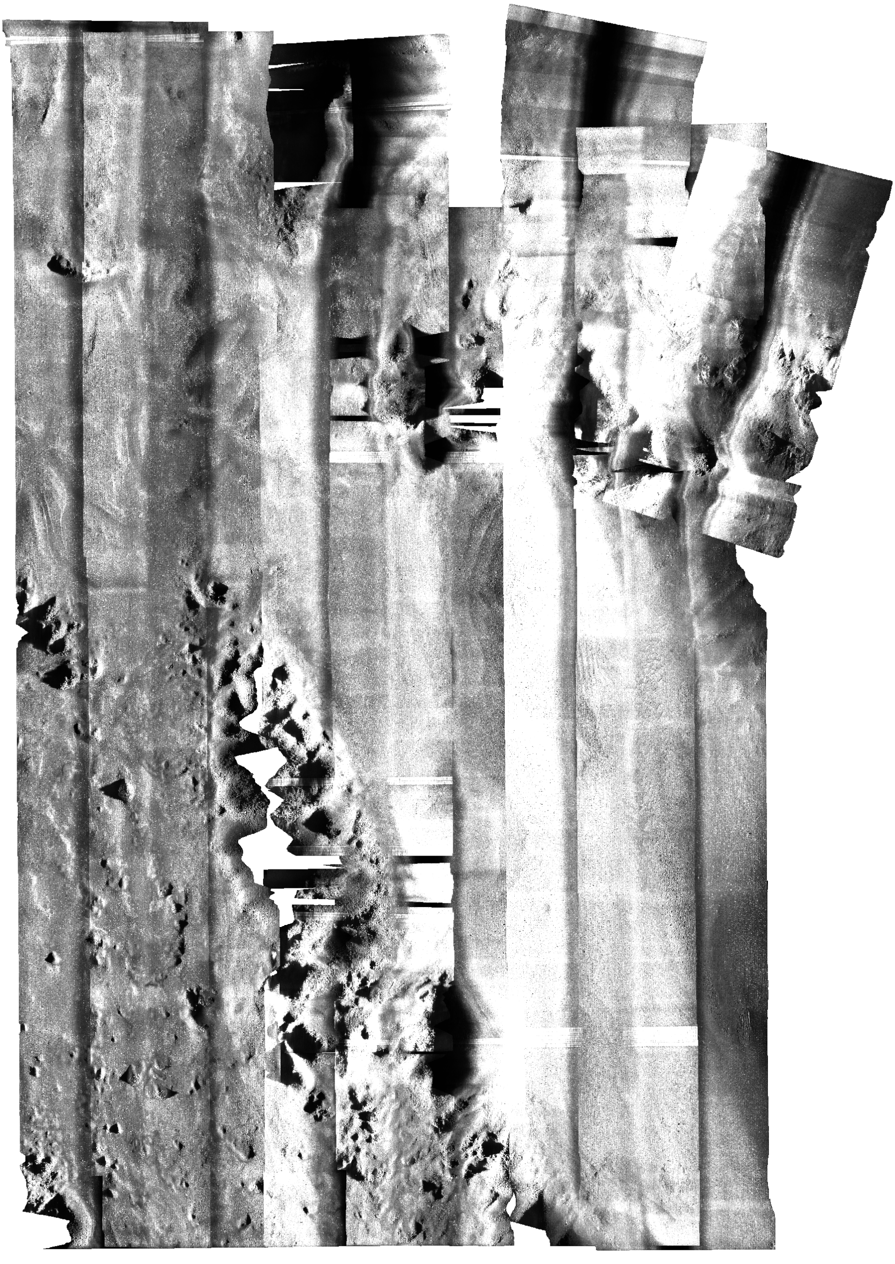
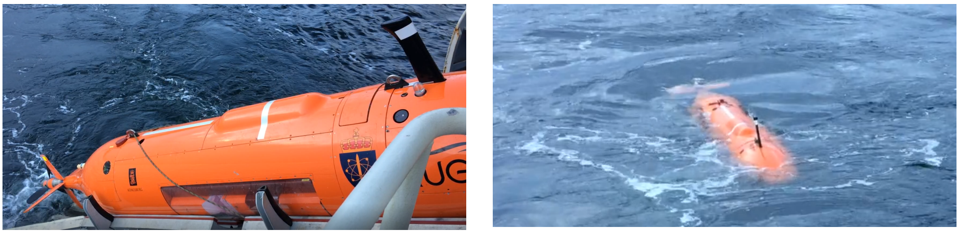


Figure 5.3: Sonar mosaic of the Nord-Leksa coral reef. The resolution is 33x33 cm, covering an area of 1325x2180 $m \approx 2.9\ km^2$. The image was created in Reflection, with the "average" mapping option. This option merges overlapping areas such that the nadir disappears and the complete area can be inspected. Some holes appear where shadows have prevented the sonar from collecting data. The image does not cover the entire dataset.



(a) Hugin right before deployment

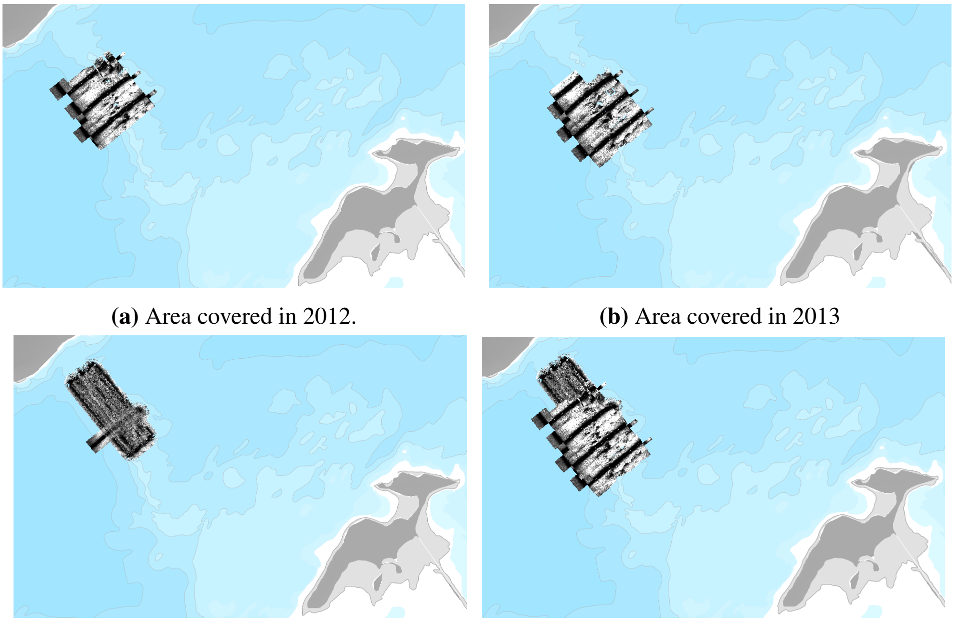
(b) Hugin directly after deployment.

Figure 5.4: Images of Hugin before and after deployment from the Hugin HUS on R/V Gunnerus for the 2017 survey of the Tautra coral reef.

The Hugin 1000 AUV from Kongsberg (Figure 5.4) is equipped with the HiSAS 1030 system. The HiSAS 1030 has a programmable center frequency, which is typically set to 100 kHz [27]. This was the case for the surveys in 2012, 2013 and 2017. The HiSAS 1030 has 32 elements in a 1.2 m receiver array, and has a range of 200 m given a $2m/s$ speed. In the surveys the range was approximately 170 m. The AUV can operate between 15 and 24 hours, depending on the payload. The sonar mosaics were created using Reflection, a Kongsberg software specifically developed for HiSAS image processing.

The Tautra ridge has depths ranging from approximately 60 metres to 30 metres, and the AUV had a mean altitude of 26 metres during each survey. It typically manages to stay between 25-27 m above the seabed throughout the surveys. The surveys did not cover the exact same area each time. This can be caused by minor variations in the navigational accuracy on Hugin, which is dependent on the quality of communication between transponders on Hugin and the surface vessel. The mosaics for each dataset is presented in QGIS in Figure 5.5.

The width and heights of the three datasets are 1230x1330m, 1230x1620m and 780x1990m for 2012, 2013 and 2017 respectively. The difference between 2012 and 2013 seems to be associated with a missing ”lap” in the lawn mover pattern, as each lap is approximately 340 m wide. The variation to the 2017 is however more distinguishing, however the largest portion of each dataset is overlapping the same area, as illustrated in Figure 5.5d.



(a) Area covered in 2012.

(b) Area covered in 2013

(c) Area covered in 2017

(d) All of the mosaics in the same image.

Figure 5.5: Area covered on the Tautra Ridge by the surveys in 2012, 2013 and 2017 respectively. The island to the left is Tautra.

5.2 Data Processing

The raw data collected from the HiSAS system was first processed by FFI before the sonar mosaics were created using the Kongsberg software Reflection. The dataset from 2017 differ from the others by higher intensities in the images. In sonar image processing, mosaicing refers to the merging of multiple swaths to form a single image of the seabed. In Reflection, ovrelapping segments can be chosen to be minimum, maximum, first, last or average. For the 2012 and 2013 datasets, maximum was used. The option makes uses the highest values in the merge between overlapping areas. In the 2017 dataset, this was a poor choice compared to the ”average” option, as it resulted in poorer contrast in the imagery. However, because a large dataset already was labelled for 2012 and 2013 data with the ”max” option, it was chosen for consistency in the training set. The mosaics including the whole dataset used for illustration of the coral coverage will use the ”last” option, as the maximum makes the image too bright and the ”average” makes distorted imagery. The ”last” option does not merge overlapping areas, but layers them with the latest addition on top of the former. Thus the nadir and some breaks in the dataset will cause the image to include a lot of black. Examples of these problems will be presented in Chapter 6.

Another mosaic option in Reflection is the resolution. Due to the high quality of the HiSAS system, it was possible to have a resolution of 3x3 cm. It was however chosen to be 4x4 cm, as a trade-off between image size and necessary resolution for detecting corals. The sonar mosaics were processed into maximum 178 000 000 pixels, as this is the memory limit in Python.

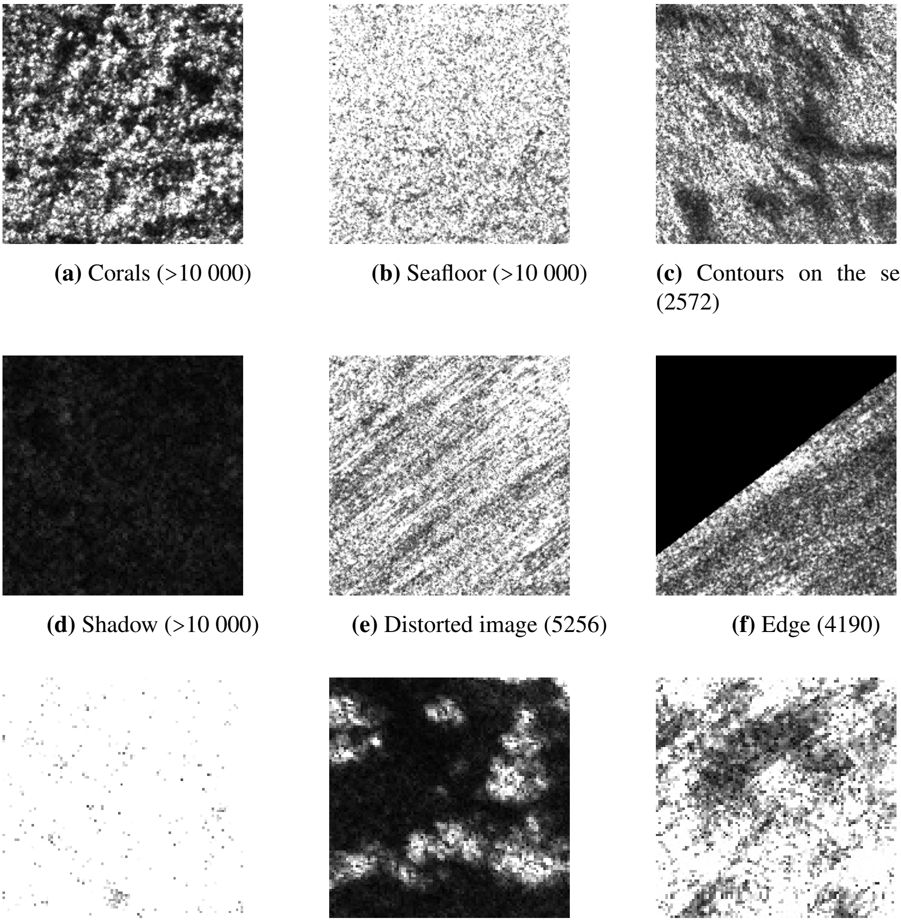
The sonar images were downloaded in TIF-format using Reflection, a software from Kongsberg. TIF, or Tagged Image Format, can handle storing high-quality imagery as well as other information. In this case, the TIFs are georeferenced, meaning that they store geographical position data. This is used to place the mosaics in geographical information system softwares like QGIS, which allocates the imagery to their correct geographical location. This allows for creation of large-scale geographical maps of corals.

5.3 Data Labelling

The training set for the CNN was designed using small sub-images of the sonar mosaics with a size of 100x100 pixels, each covering 4x4 m. In the project thesis, a CNN was trained with 200x200 pixels. This worked well, but with 100x100 the dataset would increase by a factor of four compared to 200x200 pixels, which is a very important aspect of training a neural network. 100x100 pixels seemed like a reasonable size to use, as *Lophelia Pertusa* skeleton clusters typically can grow up to 2 m in diameter [11]. Thus, corals should be visible in the 100x100 sub-images. A subset of nine coral images are shown in Appendix A.1

It has been proven that the amount of training data is very important for the resulting accuracy of the predictions [7]. Based on datasets such as MNIST and CIFAR [5], the target number of sample images for each category was at least 10 000 images. For some categories, this was not possible within the time at disposal for this part of the project.

Nine categories were identified in the dataset. An example from each category is shown in Figure 5.6, along with the number of samples that were labelled. The complexity of a neural network increases with number of classes, as well as training time and demand for large datasets. It was attempted to train a network with all the nine categories split into nine classes, which resulted in an accuracy of less than 30%.The number of images labelled for each class were limited by the number of images in the ”contours on the seafloor” category (Figure 5.6c), which only contained 2572 samples. Because of the statistical dependence of the training, one or more classes cannot include more images than the others in the training set, as this will increase the chance that one batch of images will not represent all of the classes. It was very time consuming to create sufficiently large datasets for nine classes, and therefore the classes had to be limited to a smaller number.



(a) Corals (>10 000)

(b) Seafloor (>10 000)

(c) Contours on the seafloor (2572)

(d) Shadow (>10 000)

(e) Distorted image (5256)

(f) Edge (4190)

(g) High intensity(>10 000)

(h) Shadowy coral (5445)

(i) Possible coral (976)

Figure 5.6: Examples of images from the training sets and how they are classified. Nine image categories are found in the SAS datasets from 2012, 2013 and 2017: **a)** typical example of corals, **b)** simple seafloor, **c)** contours on the seafloor, **d)** black spots, **e)** distorted imagery usually have clear lines, **f)** edges, **g)** high intensity, **h)** shadowy coral and **i)** possible coral. The images were grouped into two and three main categories for the purpose of training and optimizing a CNN.

Based on the high accuracy reached in the preliminary study in the project thesis [24], where only two classes were created, it was chosen to test for only two and three classes in this project. Three cases were made. Two of them containing two classes, and one containing three.

In the cases with two classes, the data was split into one class for coral and one for the rest called ”others”. The first ”two-class” case trained the coral class with only the high-quality coral category (Figure 5.6a). The second ”two-class” case included low-quality coral category, i.e. the shadowy coral category (Figure 5.6h), images in the coral class as well.

The three-class case split the high- and low-quality coral category into two separate classes, with the third and last class, named ”others”, containing the remaining categories.

The ”other” class included seafloor, shadows, edges, high intensities (almost white), formations, distortions and possible corals. The ”other” category contains the same amount of training samples as the coral category, evenly distributed between the remaining types of images.

The first case had 10 913 images in each class, and the coral class images were all of high quality sonar imagery. The second case included an additional 5445 images of shadowy corals, resulting in 16 358 training images per class.

The third case of training set separated the high quality image and low quality image into two classes, ”corals” and ”possible corals”, where the latter used the shadowy coral mixed with the small sized ”possible coral” category. The aim of this was to create a classification model that could provide a high certainty estimate, and a lower certainty estimate. The third dataset resulted in 6421 images per class, due to the limited number of shadowy coral and possible coral.

The distinction between poor quality corals and possible corals could be vague at times. There has been some discussion with geologist Terje Thorsnes from NGU on what is typical for corals in sonar imagery, but the labelling is highly subjective. By using ground-truthing, this could have been performed more accurately. Due to the large dataset, that was too time consuming for this project. A comparison was made with the ground-truthing made by Christian Malmquist on the 2012 sonar imagery from Tautra, see Figure 5.7. He compared ROV data to the SAS imagery along the straight lines in the image, which are the ROV tracks. The labelling of corals are based on this. They were were typically identified in clusters from the original image, and the round shapes of the coral mounds were used to distinguish the images from the other categories. Due to the size of the training dataset, a validation check for the entire dataset has not been performed.

This project aims to prove that high accuracy can be reached using CNN as a basis for a classification model. The results will be based on how accurately the model classifies according to what has been labelled as corals in the training dataset, and not necessarily what are corals in reality. In order to reach a higher certainty for the classification in reality, the entire training dataset has to be ground-truthed using optical imagery.

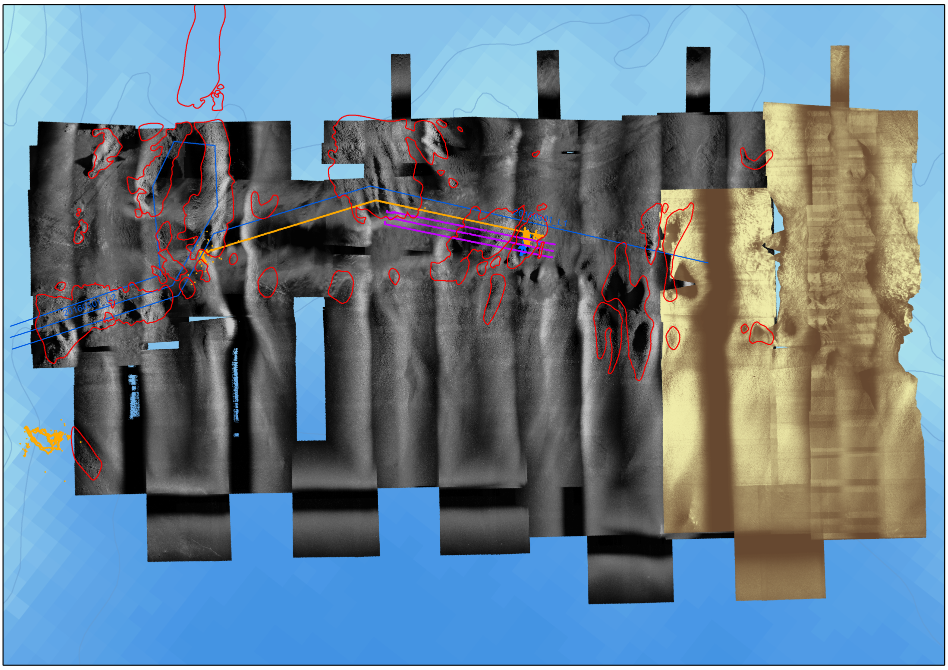


Figure 5.7: Sonar mosaic of the Tautra coral reef, the grey area is from 2012, while the bronze was added in 2013. The red contours represents corals. The lines in the middle of the image are the tracks for an ROV survey, showing where optical imagery were made. This was used for ground-truthing of the corals in the sonar imagery. The image was provided by NTNU AUR-LAB.

5.4 Programming Language and Back-ends

In the preliminary study, MATLAB and the Neural Network Toolbox were used to train a CNN [24]. The results were promising, but the personal computer used limited the functionality. The Institute of Marine Technology at NTNU could provide access to servers with powerful CPUs, but it was not possible to install the correct and necessary version of MATLAB on it. Python is another programming language with similar capabilities as MATLAB. It is open source, and dependant on programming libraries that are installed with it, much like the toolboxes in MATLAB. These could be installed on the NTNU servers, and therefore all previous code in MATLAB were converted to Python and the necessary Python libraries. After some time, a computer with a modern GPU was made available. This decreased computation time by a factor of 30, thus allowing more runs to optimize the network.

Keras is a programming library for Python containing the building blocks for neural network models. It can be supported by three different back-ends for low-level operations, like tensor manipulation. Keras provides a cleaner set up for training a CNN than using a back-end directly and is therefore often a preferred option. It is widely used and comes with a lot of documentation and sources for help in debugging. Due to this, it was chosen as the main programming library.

One goal is to create a program that can be user-friendly, and therefore it is essential to choose a software basis that is dependable and popular so that there is less chance of being quickly outdated. There are several large and popular deep learning frameworks. Two of the more popular ones being Theano and Tensorflow. Theano is Python based, with windows support. It is supposedly faster than Tensorflow, and more mature [47]. It was developed by the University of Montreal and is widely used for academic purposes. It has however not been adopted by the industry at the same rate as Tensorflow. Google published Tensorflow with an open source license in 2015, and it has since been a very popular machine learning toolkit worldwide [48]. Tensorflow also comes with a function called Tensorboard, which is a tool for visualization of the learning process. This is very helpful in debugging and optimization of the training process. Keras is compatible with both Theano and Tensorflow, but because of the capabilities of Tensorboard and the broader range of users that comes with Tensorflow, the chosen back-end for this project is Tensorflow.

To make use of GPU accelerated learning, the NVIDIA CUDA Deep Neural Network library (cuDNN) was installed. This makes use of the GPU to perform the matrix operations in the convolutional layers of the network. It decreases training time by a factor of 20-30, which is especially crucial during the optimization. The computer used for training is an Alienware Area-51 R5 with Intel(R) Core(TM) i7-7820X CPU with 8 cores and 16 logical processors. The GPU is a NVIDIA GeForce GTX 1080 Ti.

5.5 Establishing a Training Algorithm

To create and optimize a CNN, the guidelines presented by Goodfellow et al. 2016 in [7] were followed. They emphasize the importance of first deciding the goals and what error metrics to use, before establishing a functional end-to-end pipeline. The paper also includes some recommendations for what hyperparameters to use for a basic and well performing CNN. The recommended practice, combined with background knowledge as presented in section 4.2, was utilized to optimize the network.

The first step was to determine the goals. Based on the results of the project thesis, it seemed plausible to achieve an accuracy of 95-96% for the trained network, and above 85% recall for the classification. Additionally the validation loss should be as close to 0 as possible. The validation loss was monitored during the training, and early stopping was included to prevent overfitting.

Early stopping is a force quit on the training if the loss increases over a specified number of epochs [49]. *Patience* is the number of allowed increases of validation loss measured per epoch, and is included because there might be some fluctuations in validation loss over the first epochs that are not representative for decreasing performance. A patience of four epochs were used during optimization. By using a function called Model Checkpoint in Keras [49], the network weights are updated only if the validation loss improves.

Next, a working end-to-end pipeline was created in Python with Keras and the chosen back-ends. The accuracy and loss from the training were stored in text-files and Tensorboard logs. The weights and network architecture were stored in hd5f and JSON-format respectively. A pseudocode is presented in Listing 5.1.

Listing 5.1: Pseudocode for the script that trains a CNN.

```
import training data

split training, validation and test data

define hyper-parameters

create keras model
    define network architecture
    define metrics
activate callback function for Tensorboard, early stopping and
                                              model checkpoint

train CNN
evaluate CNN with test data
save network
```

A basic CNN was created based on other image classification problems and recommendations [5, 21, 7]. The architecture of the CNN is presented in Figure 5.8, and the associated default hyper-parameters as presented in Table 5.1. They are based on recommendations from Goodfellow et al. (2016) and Simard (2003), except from the learning rate which is the default value from Keras associated with the Adam optimizer.

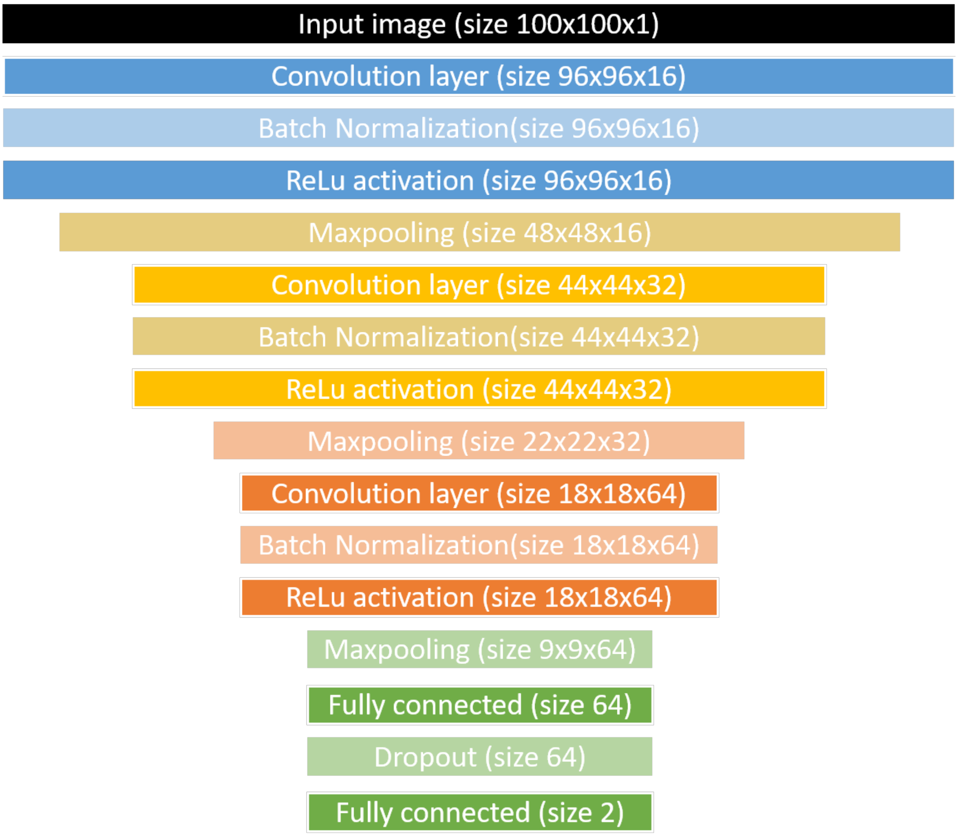

Input image (size 100x100x1)
Convolution layer (size 96x96x16)
Batch Normalization(size 96x96x16)
ReLu activation (size 96x96x16)
Maxpooling (size 48x48x16)
Convolution layer (size 44x44x32)
Batch Normalization(size 44x44x32)
ReLu activation (size 44x44x32)
Maxpooling (size 22x22x32)
Convolution layer (size 18x18x64)
Batch Normalization(size 18x18x64)
ReLu activation (size 18x18x64)
Maxpooling (size 9x9x64)
Fully connected (size 64)
Dropout (size 64)
Fully connected (size 2)

Figure 5.8: The basis design for network architecture used in the optimization.

Table 5.1: Optimization of CNN parameters

Parameter	Value
Optimization function	Adam
Batch size	64
Filter size	5x5
Number of filters	16 - 32 - 64
Learning rate	0.001
Number of epochs	20
Dropout rate	0.25
Units in FC layer	128

5.6 Inspection of Results

The metrics used for reporting the performance of the CNN training was the accuracy and loss, where accuracy is the percent of correct classification and loss is calculated using categorical cross-entropy. The final result of a trained CNN is the performance on the test dataset, which was split from the training data in the classification model prior to training (see Listing 5.1). Thus, the test data is completely unfamiliar to the network. The validation data is another subset of the training data that is not used for training, but for evaluation after each training epoch. These values were monitored using Tensorboard, a visualization tool compatible with Tensorflow, and were used to determine if the training process was performing as expected or not.

5.6.1 Confusion Matrix

Early in the optimization process, it was decided to only use two categories. Therefore, a binary classification performance model called confusion matrix was used as evaluation scheme. The definitions of the confusion matrix is presented in Table 5.2. It is a matrix that collects the true positives, true negatives, false positives and false negatives, measured relative to a target.

Several error metrics can be calculated based on the confusion matrix, i.e. precision, recall, mis-classification rate, specificity and overall classification accuracy. The different metrics reflect on different aspects of the classification performance. In some cases, it is critical that the number of failed predictions are as low as possible, for example in medical studies. For pattern recognition studies like the one in this thesis, *precision* (Equation 5.1) and *recall* (Equation 5.2) are the most common metrics. They result in an *F-score* that represents the total performance, as defined in Equation 5.3.

Table 5.2: Definition of a Confusion matrix

	Predicted Positive	**Predicted Negative**
Real Positive	True positive (TP)	False Negative (FN)
Real Negative	False Positive (FP)	True Negative (TN)

Corals were defined as the target for the classifier. Precision is the proportion of positive identifications that are correct, i.e. the number of correctly classified corals among all identified corals. The recall is the proportion of correctly classified corals among all corals, both the correctly classified and the unclassified corals. This can be summarized in Equations 5.1 and 5.2, using the definitions of Table 5.2.

$$p = \frac{TP}{TP + FP} \tag{5.1}$$

$$r = \frac{TP}{TP + FN} \tag{5.2}$$

$$F = \frac{2pr}{p + r} \tag{5.3}$$

5.7 Optimization Method

The optimization of the classification performance was evaluated using the reported test accuracy and loss, as well as performance on a verification dataset with an example image from each of datasets. These images had not been used to train or validate the network, to ensure that the results were representative for the classification performance. The images are presented in Appendix A.2.

The test accuracy and loss are not necessarily synonymous to the best performing CNN. Experience gained in the early stages of testing showed that if the difference between to CNNs are less than 5% accuracy or 0.5 in loss, it was better to evaluate the final classification performance subjectively on the verification images when determining the best performance.

Small differences in reported accuracy and loss might be caused by the random initialization of weights in the neurons, or the random shuffling of training data between each training epoch. This can cause a CNN to report quite similar accuracy, while at the same time performing classification relatively differently.

Grid Search and Random Search

Goodfellow et.al (2016) recommends to use grid search or random search when optimizing parameters. Grid search is a straight forward method where each hyper-parameter value is tested with conservative estimates within a range that is iteratively narrowed down until an optimal value is reached. This can be performed on as many hyper-parameters as needed, however Goodfellow et.al. recommends to limit it to three parameters.

Random search is similar to grid search, except that the initializing of variables are random within a specified ranges. Depending on the number of values tested within the range for each iteration, this method often converge faster towards an optimal solution than grid search [7].

It was chosen to use grid search for deciding number of classes to train for, which optimizer to use, the number of convolution layers and number of filters in the convolution layers. The remaining hyper-parameters were tested using random search, except from the dropout rate which was decided at the end of the optimization.

5.7.1 Defining Hyper-parameters

The first step of optimization was to determine the type of training set between the three versions established in Section 5.3. The next step was to determine which optimizer to use. Based on the optimizer recommended and used in other classification problems, Adam, SGD, RMSprop and Adadelta were tested. In accordance with the recommendations, the number of convolution layers were increased until the loss and accuracy ceased to improve. Finally the number of filters were decided. It was chosen to test 8, 16, 32 and 64 filters in the first convolution layer, and double the number for each convolution layer that follows the first.

It is stated in most of the literature that the learning rate is the most important hyper-parameter to optimize [7, 21, 50]. The remaining three parameters were more difficult to determine. A CNN is sensitive to batch size, which determines the number of examples the network is presented for in each iteration. The number of neurons in the FC layer has to be large enough to capture the learned features of the network, but will also increase training time due to the numerous connections for each neuron. The effect of the dropout rate had to be tested as well, as it regularizes the network training, and contributes to less overfitting.

Table 5.3: The ranges that the random search will use for optimization of learning rate, filter number, number of neurons in the first FC layer and dropout rate.

Parameter	**Range**
Learning rate	$\in [1e^{-6}, 0.01]$
Batch size	$\in [16, 512]$
Neurons in FC	$\in [10, 1000]$

Based on some initial testing with various dropout rates, it was decided that dropout could be determined after using random search to determine the other parameters. The random search was performed with four randomly initialized values in the ranges presented in Table 5.3 for each of the hyper-parameters. Two versions were selected for further testing with two different dropout rates, 0.25 and 0.5, before determining which network to use.

5.8 Optimization Results

The results of the basic grid search performed on each of the hyper-parameters number of categories, optimizer function, number of convolution layers and number of filters in each convolution layer are presented, followed by the results of the random grid search.

5.8.1 Results of the Grid Search

The results of the optimization process described in section 5.7 are presented in Table 5.4. If otherwise is not mentioned in the comment, the hyper-parameters are as defined in Table 5.1 and architecture as in Figure 5.8.

Table 5.4: Optimization results of CNN hyper-parameters

Parameter	Value	Comment	Accuracy	Loss
Nr. of classes	2	Only high quality corals	95.5%	0.2512
Nr. of classes	2	Shadowy corals included.	94.9%	0.2874
Nr. of classes	3	Shadowy coral, coral and others.	77.1%	1.1648
Optimizer	Adadelta	With learning rate: 0.01	94.7%	0.1370
Optimizer	SGD	With learning rate: 0.01	94.3%	0.1513
Optimizer	RMSprop	With learning rate: 0.001	90.7%	0.2736
Optimizer	Adam	With learning rate: 0.001	92.5%	0.1927
Nr. conv. layers	3	Continue with Adam (Lr = 0.001)	92.5%	0.1927
Nr. conv. layers	4		93.9%	0.2502
Nr. conv. layers	5		91.1%	0.3163
Filters in first layer	8	Continue with 4 layers	85.7%	0.4312
Filters in first layer	16	Doubles for each conv. layer	91.1%	0.3163
Filters in first layer	32		94.4%	0.1588
Filters in first layer	64		68.2%	1.5197

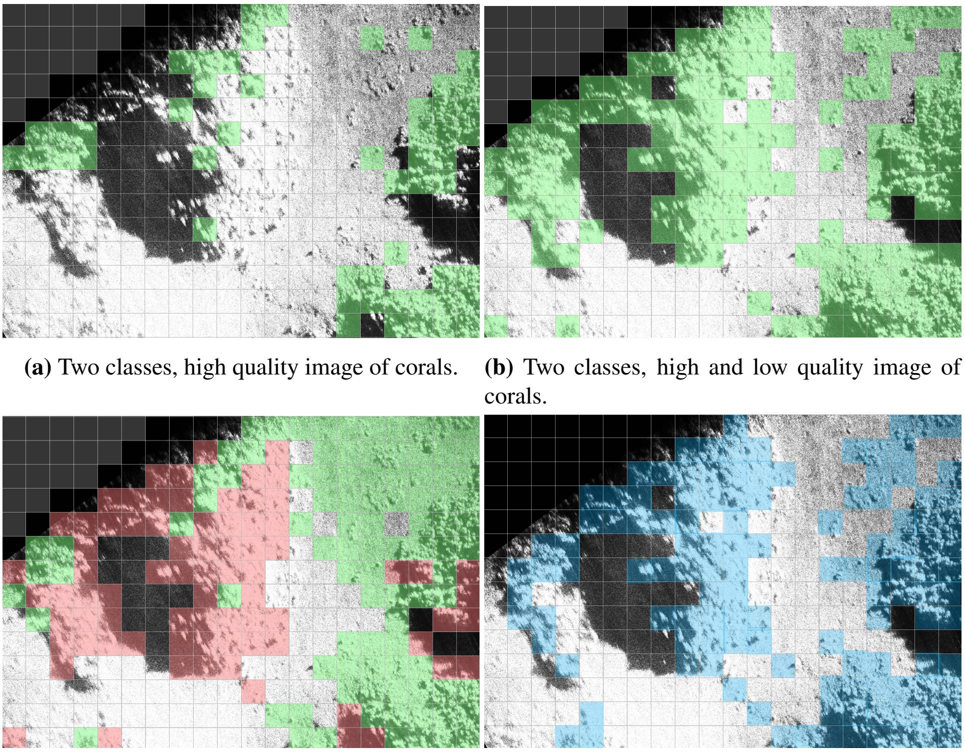
There are generally high accuracy and low losses for the CNNs throughout the optimization, as shown in Table 5.4. Therefore, the F-score had to be used as a more important measure of performance when deciding the hyper-parameters. For most of the stages in grid search, a comparison of the the classification results on the verification data was a sufficient way of determining which values of each parameter that performed best. Thus, the F-score did not have to be calculated for all the images in each step.

Number of Classes

An example is shown for the first step of the optimization if Figure 5.9. Each image illustrated the effect of the three different cases of training data on a section of the 2017 verification image. Figure 5.9a show the result of case one, which had two classes and high quality coral images in the coral class. Figure 5.9b show the result of case number two, where the shadowy-corals were included in the coral class of the two class case. The last figure show the third case representing three classes.

The selected part on the verification image in the Figure 5.9 was chosen because it shows a clear coral mound structure, situated in the middle of the image. There are also some additional coral towards the right of the image edge. Figure 5.9a show that a lot of the coral shapes of the mound are not classified as corals, while Figure 5.9b nearly covers the entire mound. Figure 5.9c shows that the three class classification works quite well, as the image on the mound is a bit distorted, while the corals towards the rights are of higher quality. However, a lot of false positives are visible towards the top right corner of the image.

The full scale version of Figure 5.9c show that there are in fact large areas of false positives for the three class case. Based on this, case number two were chosen for the proceeding work. Figure 5.9d illustrates a subjective opinion on how the image should have been classified based on the labelling for case number two. The confusion matrix of Figure 5.9b was calculated in comparison to Figure 5.9d, and the results are shown in Table 5.5. The precision was 82.6 % and the recall 91.3% , resulting in an F-score of 86.7%. This



(a) Two classes, high quality image of corals.

(b) Two classes, high and low quality image of corals.

(c) Three classes.

(d) Subjective proposition of true corals for case two.

Figure 5.9: Representative example from the classification results when optimizing the categories. The illustration shows the performance on an image from the 2017 dataset. These sub-images of size 2000x1400 pixels are smaller pieces of the images presented in Figures A.6 to A.8 in Appendix A.3.

only represents this small part of the actual 2017 verification image.

Table 5.5: Confusion matrix

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	116	11	127
Real Negative	24	129	153
Sum	140	140	280

By using case number two as training data, the dataset for each class included 17334 images. These were split into training, validation and test data, as mentioned in Listing

5.1. 20 % of the images were used for validation, and 20% of the remaining used for testing. This means that of the total 34668 images, 22187 were used for training, 5547 were used for validation and 6934 used for testing. The validation and training set are selected from the training data using a random seed, such that they are always the same for each test of training CNNs.

Similar comparisons with the verification data were made when deciding the optimizer, number of convolution layers and size of the filters.

Optimizer

Four different types of optimizers were tested based on recommendations by Goodfellow et.al and the Stanford CS321 Lecture notes [43], namely Adadelta, SGD, RMSprop and Adam. The mini-batch gradient descent technique. was used with each of them, and the learning rate was set to the default values in the Keras documentation [49]. They generally performed similarly on the reported accuracy and loss, as shown in Table 5.4, but the results of the classified verification images varied. Adam had less false positives and false negative than the others, and was therefore chosen as optimizer.

Adam is an acronym for adaptive moment estimation, and it was inspired by the adaptive gradient algorithm (Adagrad) and root mean square propagation (RMSprop) optimizers. Adam computes individual adaptive learning rates based on first and second moments of the gradient of the loss function [51].

Number of Convolution Layers

The number of convolution layers were increased until the accuracy and loss stopped improving, as recommended by Goodfellow et. al. The results in Table 5.4 show that using four convolution layers had similar results to three layers, and that increasing to five did not increase the accuracy. Also here a visual inspection of the classifications on the verification data was made, and it was decided to proceed using four convolution layers.

Number of Filters

The final hyper-parameter optimized using a grid search is the number of filters in the convolution layer. The number of filters are doubled between each convolution layer, due to the maxpooling layers ability to decrease the size of the former layer. As shown in the results in Table 5.4, using 32 filters in the first layer resulted in highest accuracy and lowest loss.

The result of a training will have some level of randomness associated with it. The initial weights of the network are random, in addition to the random shuffle of the training set. The latter will result in different sets of batches being presented to the network. It is possible to turn off the shuffle on the training dataset, and define the initial weights. However, this was not discovered in time to retrain and reproduce the results. The effect of these random states are unclear, but it might be that the result will converge to similar states anyhow.

5.8.2 Results of Random Search

Several attempts was performed with random search, in order to test a wide range of random values within the ranges defined in Table 5.3.

Table 5.6: The random values used for random search of hyper-parameters batch size, units in FC layer and learning rate

Parameter	Values			
Batch size	118	414	458	494
Learning rate	0.0001	0.0011	0.0037	0.0042
Units in FC layer	349	363	507	899

The runs with the best results are presented in Table 5.7. The attempt with four values of each parameter resulted in 64 different versions of a CNN, presented in Table 5.7. Run number 42 and 2 were tested with two different dropout rates. They were chosen because they have quite similar performance, but large difference in batch size. As dropout was excluded from the random search, two dropout rates were tested for the two selected CNN models. A dropout rate of 0.25 resulted in accuracy of 94.94% and 93.99% with loss of 0.1443 and 0.1734 for run 42 and 2 respectively. Dropout rate if 0.5 resulted in accuracy 92.10% and 90.17% with loss of 0.1938 and 0.2252 respectively.

The results are quite close to one another, taking the randomness of initialized weights and batches into account.The four network models were tested on the verification dataset. By visual inspection, it was clear that the reported highest accuracy was the best performing network, which is expected. Therefore, the resulting CNN used for quantification of the coral reefs were created using a batch size of 414, learning rate of 0.0001 and 507 neurons in the first FC layer.

Table 5.7: The ten best results of the random search.

Number	Batch Size	FC units	Learning rate	Accuracy	Loss
42	**414**	**507**	**0.0001**	**0.1427**	**95.01%**
46	414	349	0.0001	0.1531	94.95%
53	458	899	0.0011	0.1419	94.91%
2	**118**	**363**	**0.0001**	**0.1574**	**94.71%**
18	494	363	0.0001	0.1674	94.68%
6	118	899	0.0001	0.1818	94.45%
49	458	363	0.0011	0.2180	93.99%
26	494	507	0.0001	0.1938	93.77%
21	494	899	0.0011	0.1761	93.70%
62	458	349	0.0001	0.1767	93.70%
30	494	349	0.0001	0.1645	93.57%

Four versions for the Nord-Leksa verification image are presented in Appendix A.4, showing the result of nr. 42 and 2 with dropout 0.25 and 0.5. By visual inspection, it is evident that that nr. 42 with a dropout of 0.25 performs better than the others CNN versions, according to the labelling. This is also expected based on the resulting accuracy's and losses.

The suggested result CNN to be used to classify has 363 neurons in the first FC layer, a batch size of 118, dropout rate 0.25 and learning rate 0.0001. A summary of all the resulting hyper-parameters is presented in Table 5.8.

Table 5.8: Result of optimization of CNN parameters

Parameter	**Value**
Optimization function	Adam
Batch size	414
Filter size	5x5
Number of filters in first conv. layer	32
Learning rate	0.0001
Number of epochs	100
Dropout rate	0.25
Neurons in FC layer	507

5.9 Visualizing the Filters

Quiver is a Python library that can be used to visualize how the network filters affect the input imagery. Two sample images, one coral and one sandy seafloor, are presented here as a comparison. Four filters of each convolution layer is shown in Figure 5.10 and 5.11 for the coral and sand sample.

The filters in the first layers typically search for edges or smooth the input image in order to search for possible characteristic features. The texture is visibly different for the coral sample in comparison to the seafloor sample. It is not easy to see how the final layer differs for the two samples, or if the layer is important at all. How the fully connected layers finds global features is part of the ”black box” of the CNN. The visualization of the filters are however important for understanding how the network learns, and an important tool during debugging. Noisy patterns in the convolution layers can indicate that the network has not been trained long enough [43].

A symptom of too high learning rate can be shown in the visualization of activation layers if there are many ”dead” filters, showing only black images. However, this can also happen if dropout is added as a regularizer in the network. Figure 5.12 illustrates how the dropout affects the neural network. The completely black filter is a ”victim” of the dropout function, which ”kills” the activation for that specific filter.

Another effect that can be visualized in the activation layers are that the filters become sparser, and that the activations are more local. This is evident already in the second activation layer, illustrated with the coral sample in Figure 5.13. Notice how some of the filters are only small dots, which might be caused by activation on the round shape of a coral from the original image.

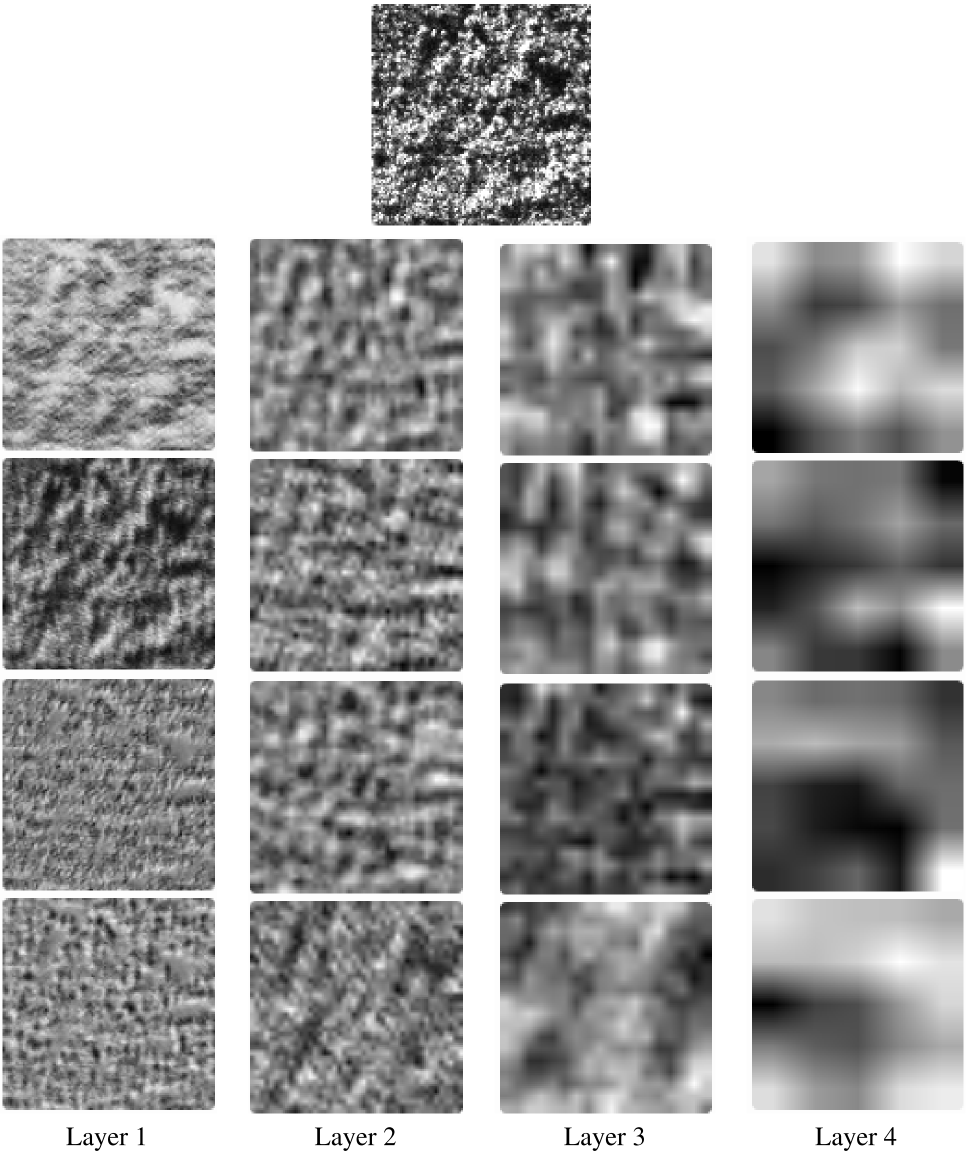


Figure 5.10: Illustration of how the filters changes for each convolution layer using a sample of coral image. A subset of four filters were collected from each layer using the python package Quiver.

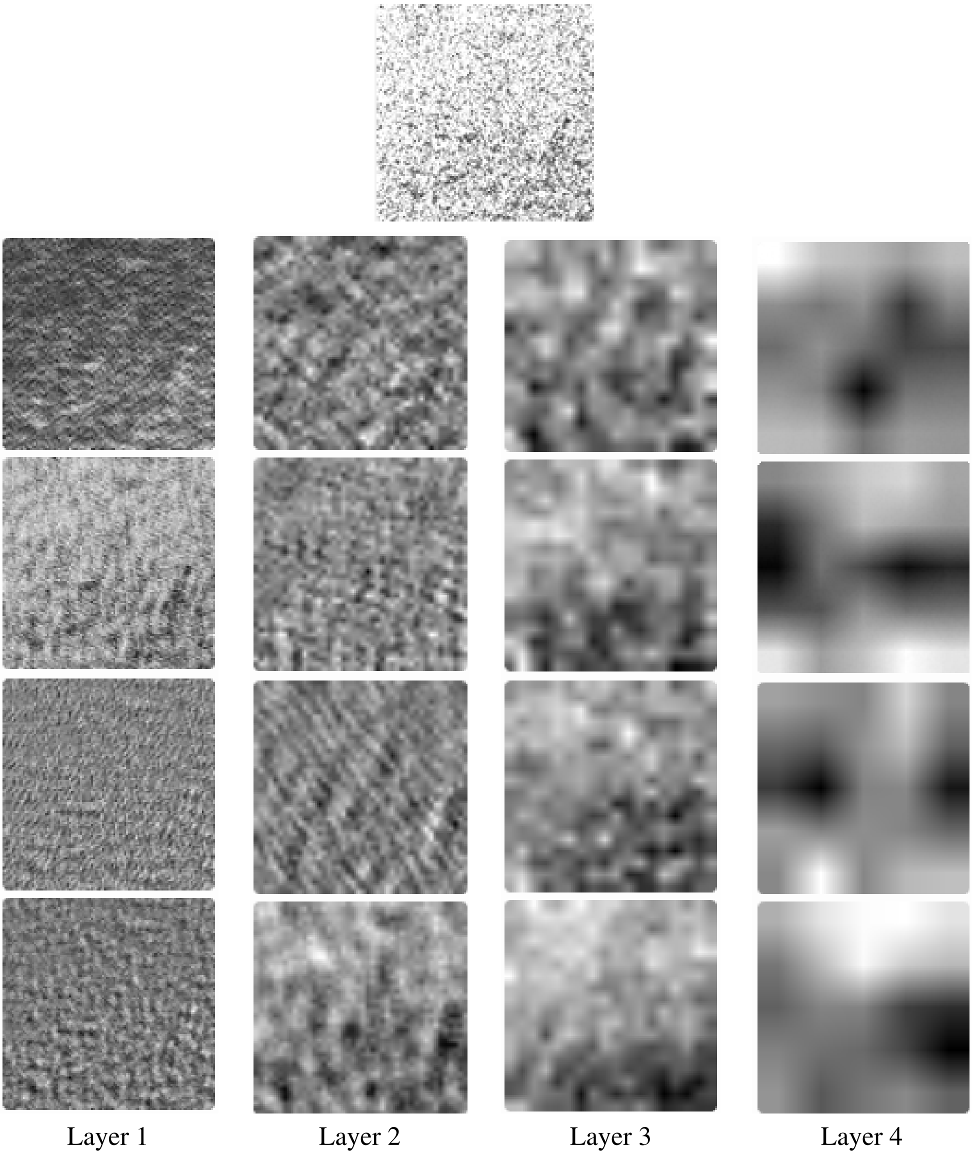


Figure 5.11: Illustration of how the filters changes for each convolution layer, using a sample of the seafloor category. A subset of four filters were collected from each layer using the python package Quiver.

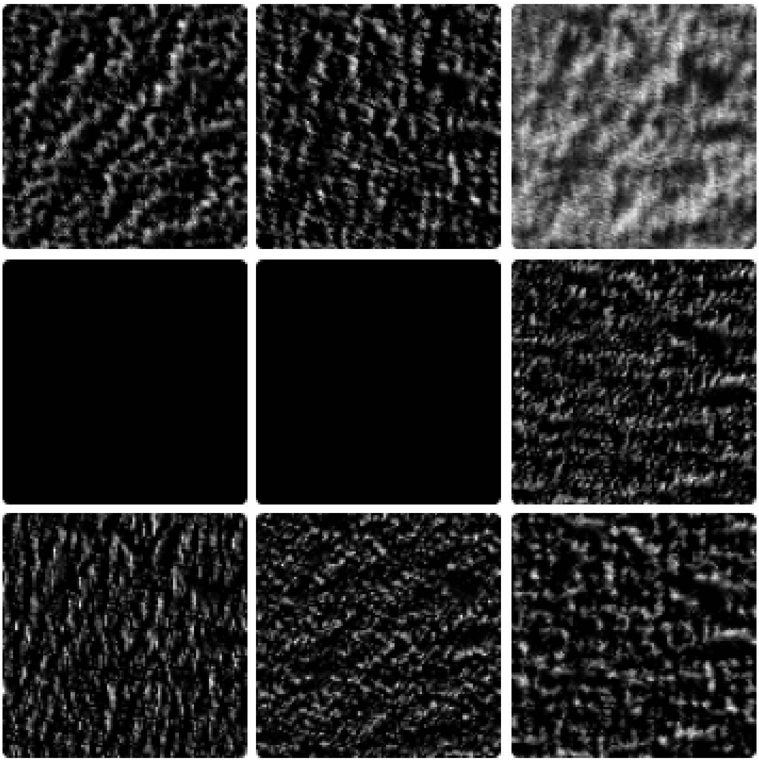


Figure 5.12: The dropout effect is illustrated here, where two filters are shown as black in the visualization tool Quiver. The filters are collected from the activation layer after the first convolution layer, and illustrated with the same sample of coral as in Figure 5.10.

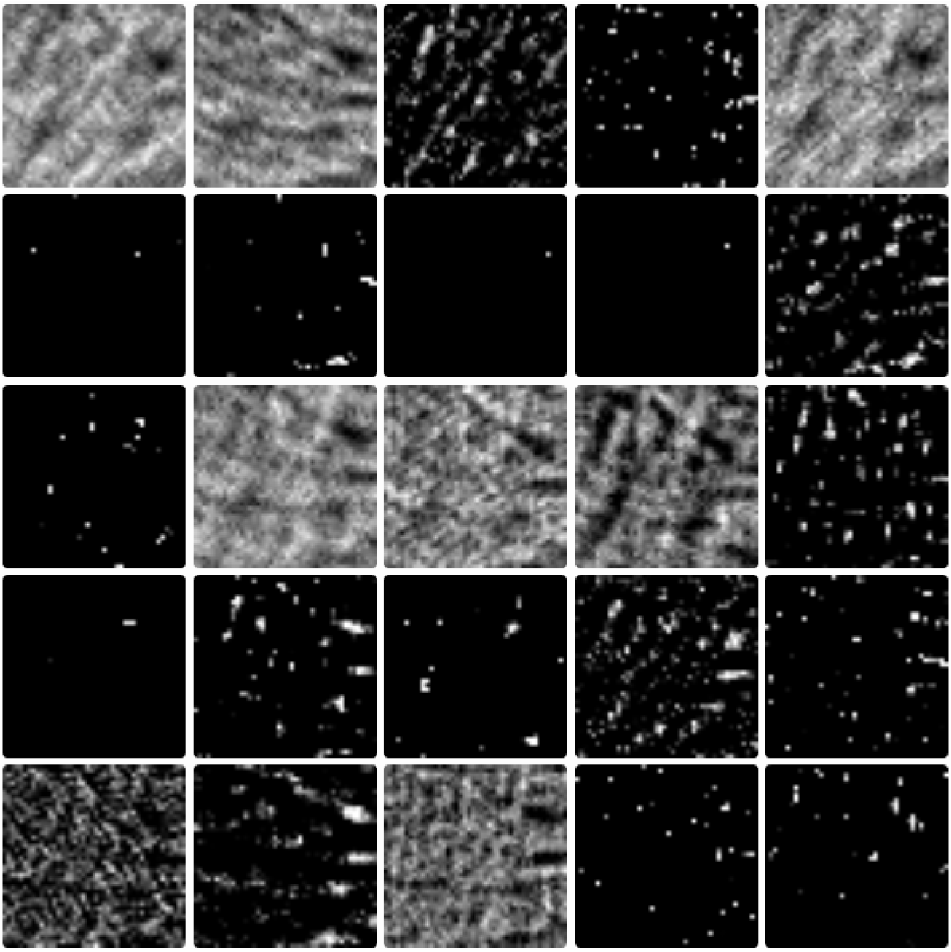


Figure 5.13: The figure show a subset of the filters in the second activation layer, which follows the second convolution layer. and illustrated with the same sample of coral as in Figure 5.10.

5.10 Classification

After an optimized network was chosen, the program was tested on the verification mosaics (see Appendix A.2). They were selected based on their characteristics, in such that the network was tested on both high and low quality imagery of corals. The result of classified mosaics are layered images where an RGB-image with the classified areas and a grid is merged with the original sonar mosaic.

A pseudocode for the classification algorithm is presented in Listing 5.2. The actual Python code is uploaded in a separate file along with the thesis. The classification is performed by first loading the trained model and mosaics to be classified, before determining how many classes the model is trained for. After that, the mosaics are split into 100x100 sub-images that each is classified by the CNN. The number of sub-images that are labelled corals, and shadowy corals for the three-class case, are counted in order to determine the coral coverage. The classifier outputs the layer RGB-image that is partly transparent green over the areas that are classified as corals, or red for shadowy corals. Additionally, a grid of size 100x100 pixels is added at the end as a guide for visual inspection of each sub-image.

Listing 5.2: Pseudocode for the script that classifies mosaics using the trained CNN.

```
import CNN model
import images

specify if there are two or three categories
if num_categories == 3
    split image into sub-images of 100x100 pixels
    classify the sub-images
    create a layer of color coded classification result
    count coral-classifications
    count shadowy coral - classifications
else
    split image into sub-images of 100x100 pixels
    classify the sub-images
    create a partly transparent layer of color coded classification result
    count coral-classifications

make image with the classification layer on top of the original image
create a grid with size 100x100 on top of the image
save layered image
```

The training data were made with the ”maximum” option for mapping in Reflection. When the 2017 sonar imagery was made available from NTNU AUR-LAB and FFI, it showed that the mosaics were much brighter than the ones in from 2012 and 2013. Due to the labelling that had already been performed on the 2012 and 2013 data in the preliminary study, it was chosen to continue using ”maximum” mapping.

A comparison of coral coverage between 2012, 2013 and 2017 data will be made, based on a small region on the seafloor containing a coral mound that was identified in each dataset.

5.11 Improving Accuracy with Closing

In order to improve the accuracy, the image processing technique *closing* was performed on the classification layers of the mosaics. This removes classified sub-images that stand alone, i.e. has no neighbours with same predicted class. It was tested on the verification set, with a quadratic structuring element of size 110x110. It is slightly larger than one sub-image (100x100), which ensures that only singles are eroded away.

5.12 Creating Maps

The final product of the classification model presented in this thesis, are classified mosaics covering the entire dataset from each year at Tautra, as well as the one from Nord-Leksa. The purpose is to visualize the coral coverage in a map, showing both the coral coverage and the sonar imagery.

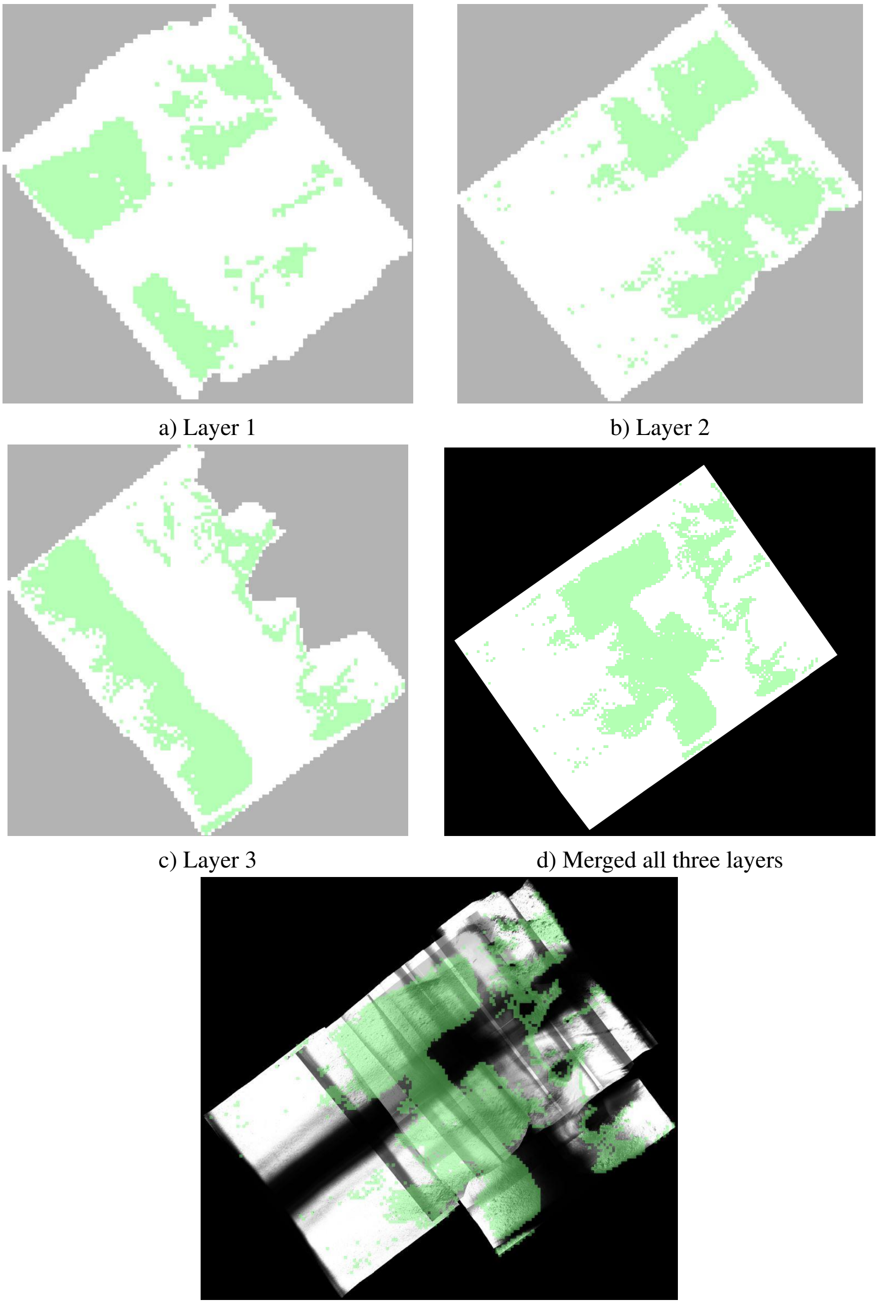
The Python library GDAL was attempted for reading the geo-reference on the sonar mosaics. However, even though the installation was successful, the library did not work in Python. Therefore, this part of the model had to be partly designed in MATLAB, using the Mapping Toolbox.

Listing 5.3: Pseudocode for the script that calculates total coral coverage of an area, using several sonar mosaics that might overlap.

```
import sonar mosaics
import classified layers for the mosaics

for image in all mosaics
    store corner coordinates
    store pixels rows and columns
    store image width and height in meters

% create bounding box by defining the rectangle bottom left
% and top right corner
bottom_left_x = minimum of bottom left x coordinates among the images
bottom_left_y = minimum of bottom left y coordinates among the images
top_right_x = maximum of top right x coordinates among the images
top_right_y = maximum of top right y coordinates among the images

width_box = top_right x - bottom_left_x
height_box = top_right y - bottom_left_y

% read in classification layers and allocate them to correct location
% in bounding box

for image in all mosaics
    collect top left corner coordinate
    assign location in bounding box
    allocate the layer image inside the bounding box

for all pixels in bounding box
    count coral pixels
    coral_coverage = coral_count*0.04*0.04 % coral coverage in m^2

scale bounding box by 4/33

save bounding box image
save coral count and coral coverage
```

A pseudocode for the algorithm is presented in Listing 5.3. The algorithm first creates a bounding box of the area covered by the 4x4cm resolution sonar mosaics. This is performed by reading the geo-reference of each corner of the mosaics. The layers containing the information on what part of the sonar mosaics are classified as corals are used to allocate the same information at the correct location within the bounding box. The algorithm assumes that all regions that are marked corals in each layer are correct, hence two overlapping areas where only one is marked as coral will result in coral inside the bounding box. The process is illustrated in Figure 5.14, where three overlapping mosaic layers are merged into one bounding box. The location of each layer is allocated based on the geo-reference of the associated sonar mosaic.

Some offset between the RGB-layer and sonar mosaic would occur due to the inaccuracies of navigation data that were used to position the imagery in the algorithm, as well as when the layer resolution was changed to fit a 33x33 resolution mosaic, i.e. scaled by $4/33$. An test was performed using three overlapping mosaics with 4x4cm resolution, to check if the accuracy of positioning and overlapping coral counts was made correctly.



a) Layer 1

b) Layer 2

c) Layer 3

d) Merged all three layers

e) Merged the bounding box and a sonar mosaic containing the overlapping areas.

Figure 5.14: Illustration of how the bounding box technique performs on three overlapping layers.

Chapter 6

Results

This chapter is split into three sections. The first section is a comparison of the F-scores on the verification data set before and after closing. The second presents the results of the quantification of corals in each of the three Tautra datasets followed by the result for Nord-Leksa. Finally, a small region of the same area covered in each Tautra data set is compared.

6.1 F-Scores on the Verification Mosaics

2012 Before Closing

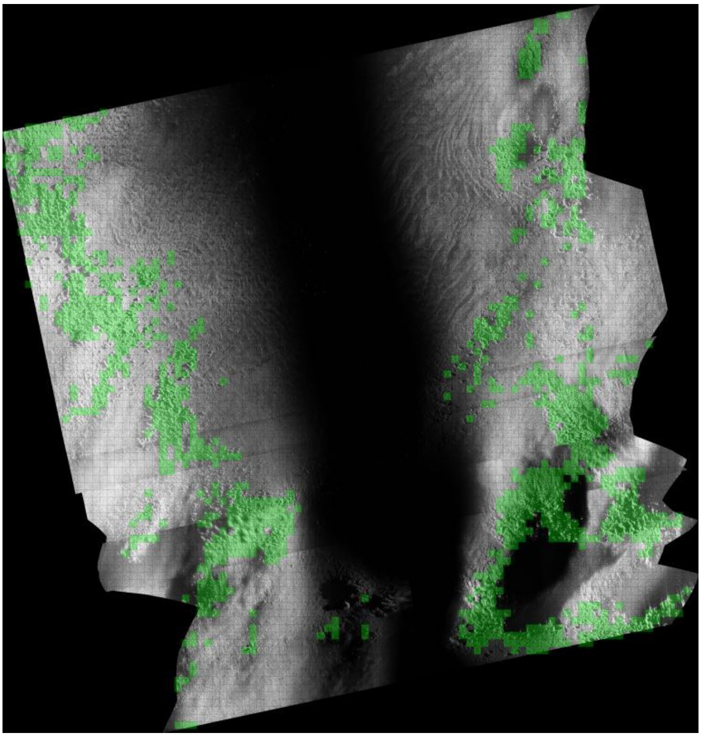


Figure 6.1: Verification image from the 2012 Tautra data set. The mosaic covers an area of 0.107 km^2.

Figure 6.1 show the result of classification on the 2012 verification image using the optimized CNN. The confusion matrix for this image is presented in Table 6.1, which results in a precision of 85.2% and a recall of 80.4%. The F-score is 82.72%.

Table 6.1: Confusion matrix for 2012 verification image from Tautra, calculated with the optimized network.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	1187	289	1476
Real Negative	207	4885	5092
Sum	1394	5174	6568

2012 After Closing

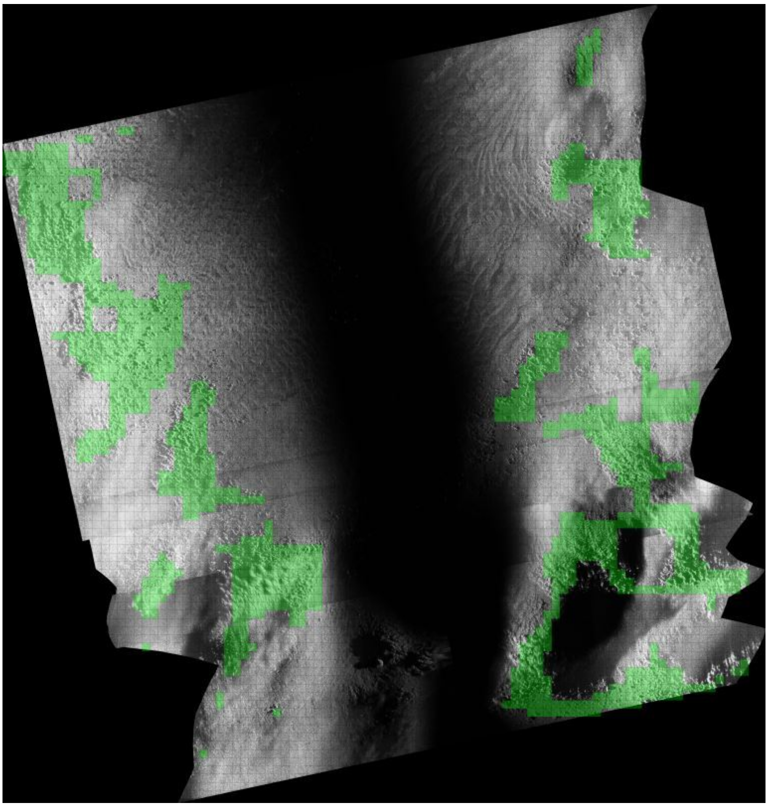


Figure 6.2: Verification image from the 2012 Tautra data set after *closing* is performed on the layer with the classification results.

Figure 6.2 show the result of the closed version of the 2012 verification image. The confusion matrix for this image is presented in Table 6.2, which results in a precision of 89.9% and a recall of 64.5%. The F-score is 75.11%.

Table 6.2: Confusion matrix for the 2012 verification mosaic from Tautra, after closing.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	952	524	1476
Real Negative	107	4985	5092
Sum	1059	5509	6568

2013 Before Closing

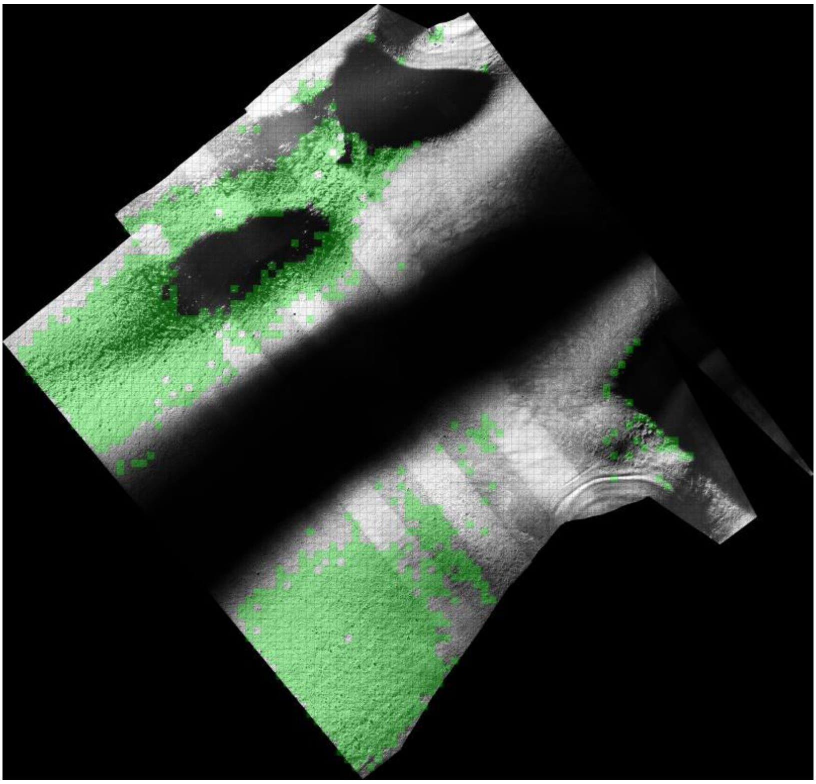


Figure 6.3: Verification image from the 2013 Tautra data set. The mosaic covers an area of 0.087 km^2.

Figure 6.3 show the result of classification on the 2013 verification image using the optimized CNN. The confusion matrix for this image is presented in Table 6.3, which results in a precision of 90.8% and a recall of 92.6%. The F-score is 91.7%.

Table 6.3: Confusion matrix for 2013 verification mosaic from Tautra, calculated with the optimized network.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	1540	123	1663
Real Negative	156	3751	3907
Sum	1696	3874	5570

2013 After Closing

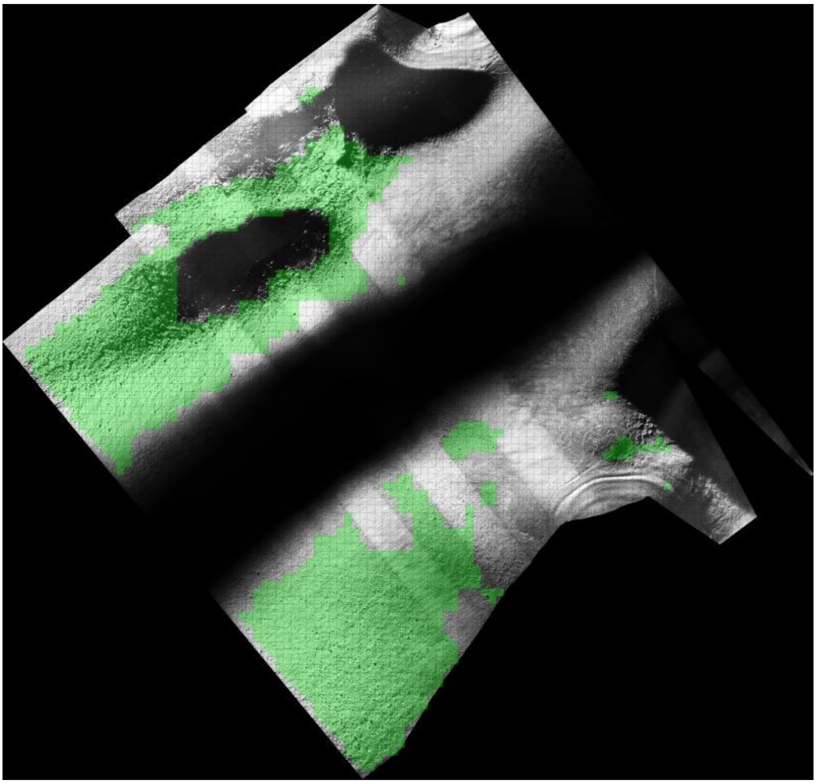


Figure 6.4: Verification image from the 2013 Tautra data set after *closing* is performed on the layer with the classification results

Figure 6.4 shows the result of classification on the 2013 verification image after closing. The confusion matrix is presented in Table 6.4, which results in a precision of 97.1% and a recall of 85.0%. The F-score is 90.6%.

Table 6.4: Confusion matrix for the 2013 verification mosaic, after closing.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	1414	249	1663
Real Negative	42	3865	3907
Sum	1456	4114	5570

2017 Before Closing

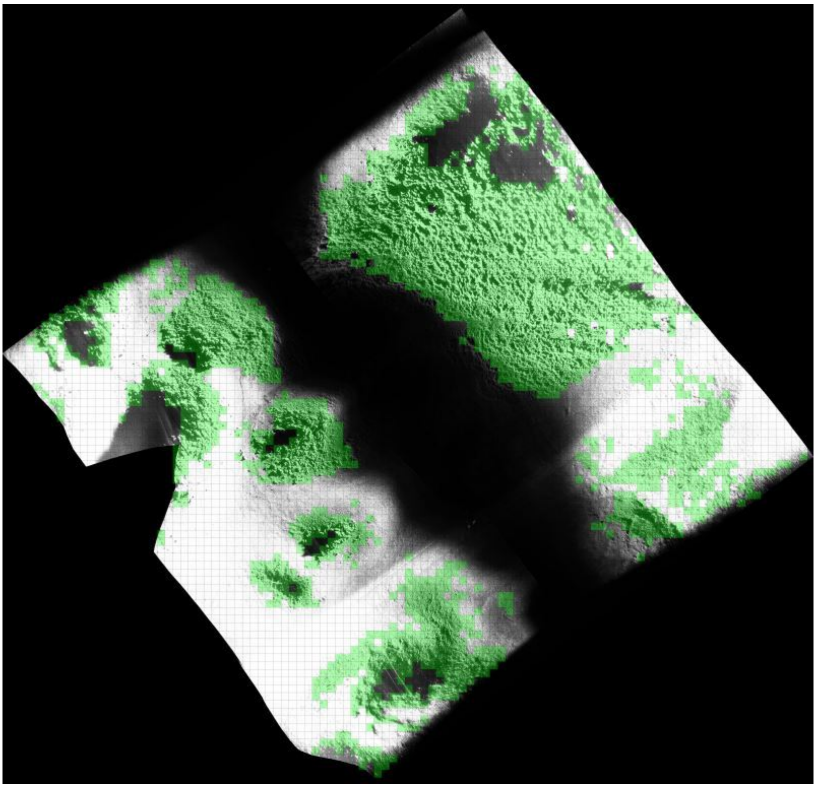


Figure 6.5: Verification image from the 2017 Tautra data set. The mosaic covers an area of 0.088 km^2.

Figure 6.5 show the result of classification on the 2017 verification image using the optimized CNN. The confusion matrix for this image is presented in Table 6.5, which results in a precision of 94.9%, recall of 94.9% and F-score 94.9%.

Table 6.5: Confusion matrix for 2017 verification mosaic from Tautra, calculated with the optimized network.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	1896	101	1997
Real Negative	102	3501	3603
Sum	1998	3602	5600

2017 After Closing

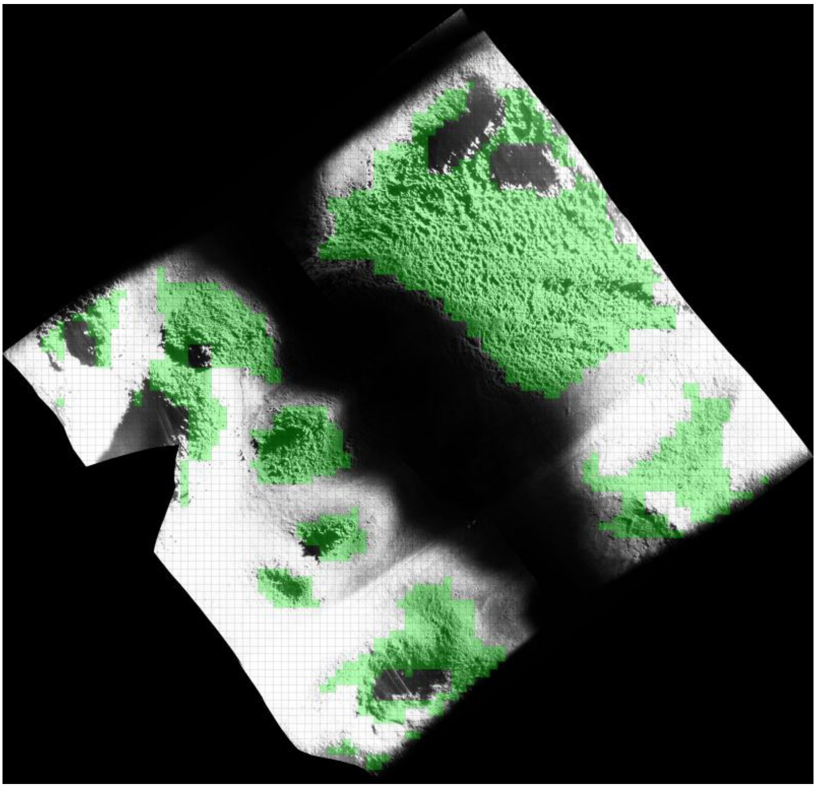


Figure 6.6: Verification image from the 2017 Tautra data set after *closing* is performed on the layer with the classification results

Figure 6.6 show the result of classification on the 2017 verification image after closing. The confusion matrix for this image is presented in Table 6.6, resulting in a precision of 96.5%, recall of 80.7% and F-score 87.9%.

Table 6.6: Confusion matrix for the 2017 verification mosaic, after closing.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	1612	385	1997
Real Negative	68	3523	3603
Sum	1670	3930	5600

Nord-Leksa Before Closing

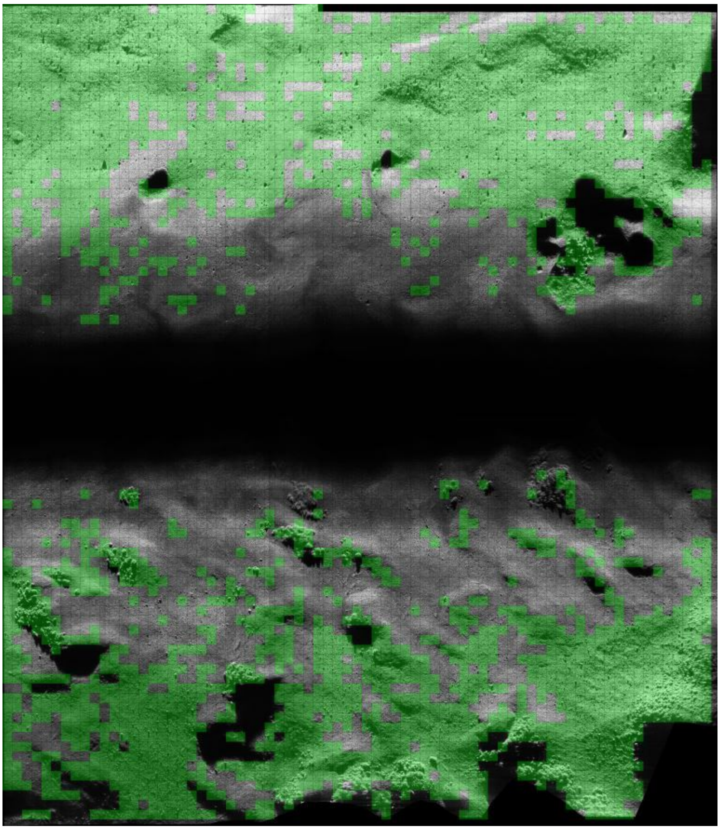


Figure 6.7: Verification image from the Nord-Leksa set. The mosaic covers an area of 0.097 km^2.

Figure 6.8 show the result of classification on the Nord-Leksa verification image after closing. The confusion matrix for this image is presented in Table 6.7, resulting in a precision of 88.7%, recall of 88.7% and F-score 88.7%.

Table 6.7: Confusion matrix for Nord-Leksa, calculated with the optimized network.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	2277	289	2566
Real Negative	290	3199	3489
Sum	2567	3488	6055

Nord-Leksa After Closing

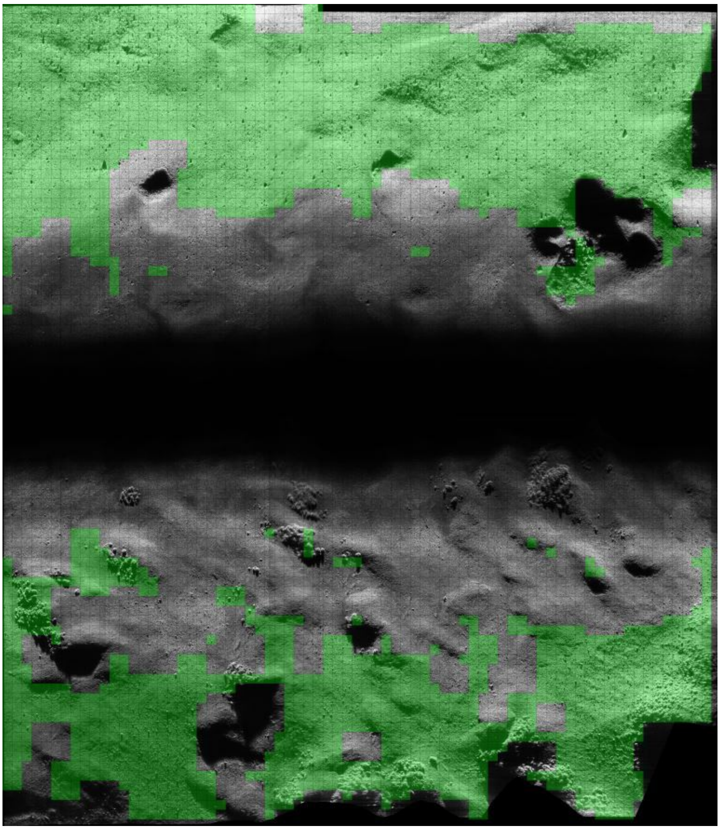


Figure 6.8: Verification image from the Nord-Leksa set after *closing* is performed on the layer with the classification results.

Figure 6.8 show the result of classification on the Nord-Leksa verification image after closing. The confusion matrix for this image is presented in Table 6.8, resulting in a precision of 93.3%, recall of 86.7% and F-score 89.9%.

Table 6.8: Confusion matrix for the Nord-Leksa verification image, after closing.

	Predicted Positive	**Predicted Negative**	Sum
Real Positive	2226	340	2566
Real Negative	161	3328	3489
Sum	2387	3668	6055

6.2 Quantification of Complete Data Sets

The results of coral quantification are presented in Table 6.9. The area coverage differs for each of the Tautra data sets. The coral coverage in 2012 is approximately $0.34km^2$ larger than in the 2013 data set, even though the 2013 data covers $0.2km^2$ less area. The results are presented in Figure 6.10 and 6.11. The 2017 mosaic in Figure 6.12 has a narrower and longer coverage than the two others, and reports a coral coverage of $0.66km^2$ on the $1.3km^2$ area covered by the sonar. Figure 6.9 show what parts of the Tautra area that includes confirmed coral reefs, as reported by the MAREANO project [1]. The orange dots represent the confirmed coral reefs, which can be compared to the green areas representing the classified corals in Figures 6.10 to 6.12.

Table 6.9: Results of coral quantification of each data set.

Data set	**Coral coverage [km^2]**	**Total area [km^2]**	**Percent of total area**
Tautra 2012	0.7694	1.7	45.3%
Tautra 2013	0.4325	1.9	22.7%
Tautra 2017	0.6618	1.3	50.9%
Nord-Leksa	2.2742	4.6	49.4%

The Nord-Leksa mosaic in Figure 6.13 shows a $4.6km^2$ large area, where 49.4% of it is covered by coral according to the classification scheme. Figure 6.14 show the verified coral reefs from the MAREANO projects, as a comparison to Figure 6.13.

The images of the sonar mosaics with the coral coverage are shown more in detail in Appendix A.5.

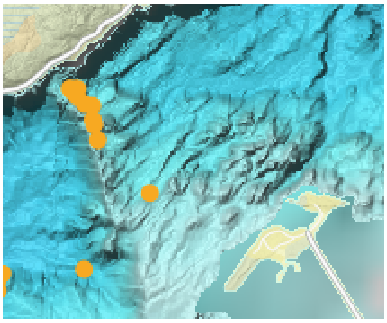


Figure 6.9: Illustration of the Tautra ridge and the confirmed coral reefs marked by orange dots. The image is collected from the MAREANO project websites [1].

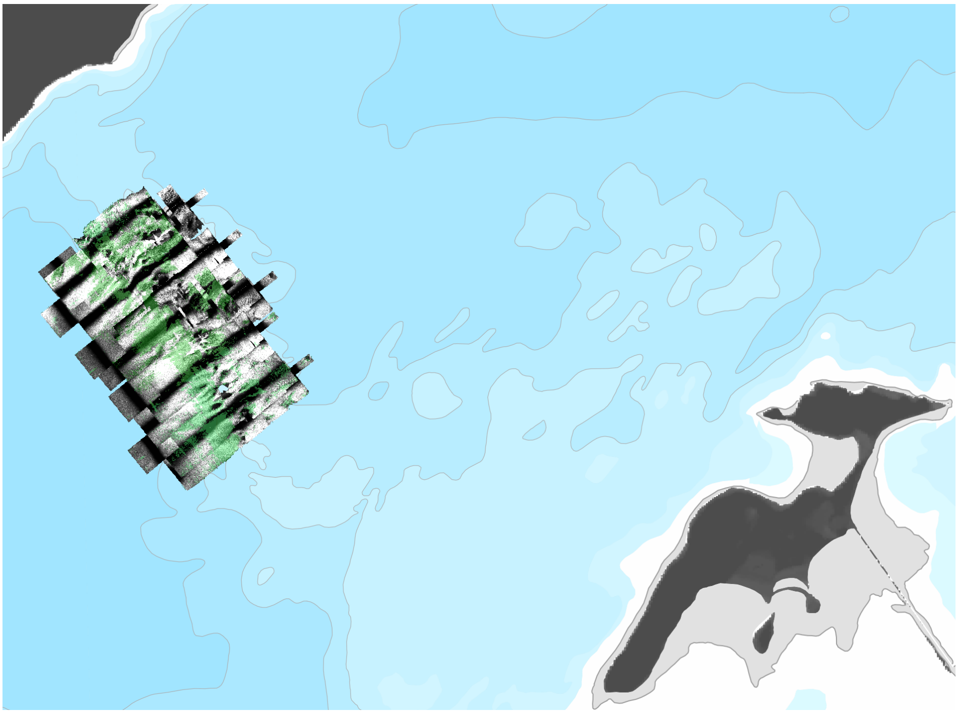


Figure 6.10: Representation of the coral coverage from the 2012 Tautra data set in QGIS. The area covered by the sonar is 1.7 km^2, and 45.3% of it is covered by corals according to the CNN classification.

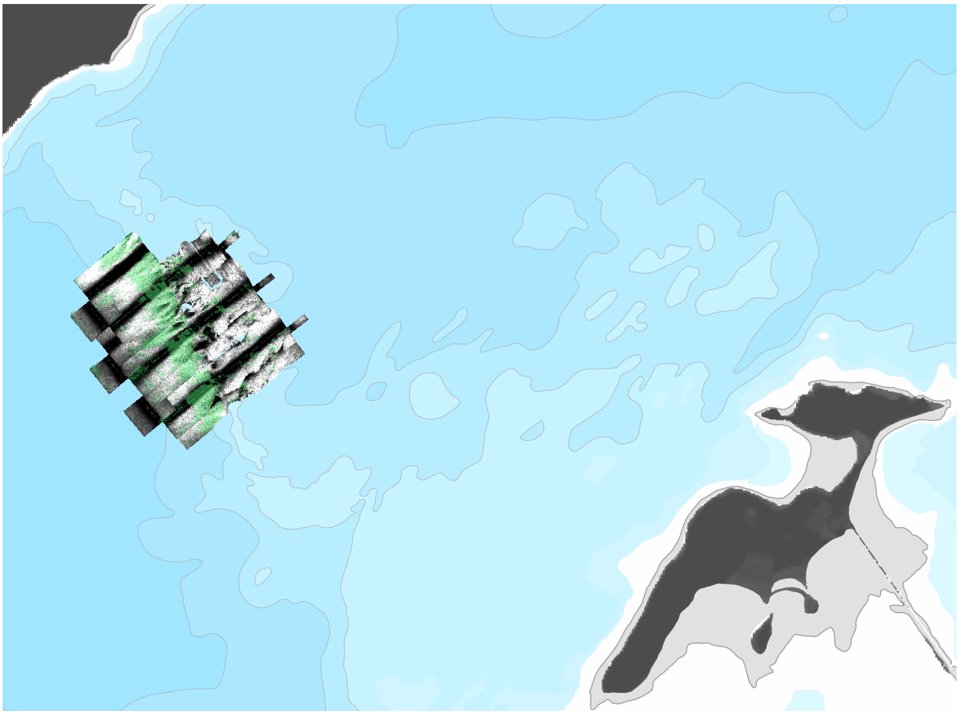


Figure 6.11: Representation of the coral coverage from the 2013 Tautra data set in QGIS. The area covered by the sonar is 1.9 km^2, and 22.8% of it is covered by corals according to the CNN classification.

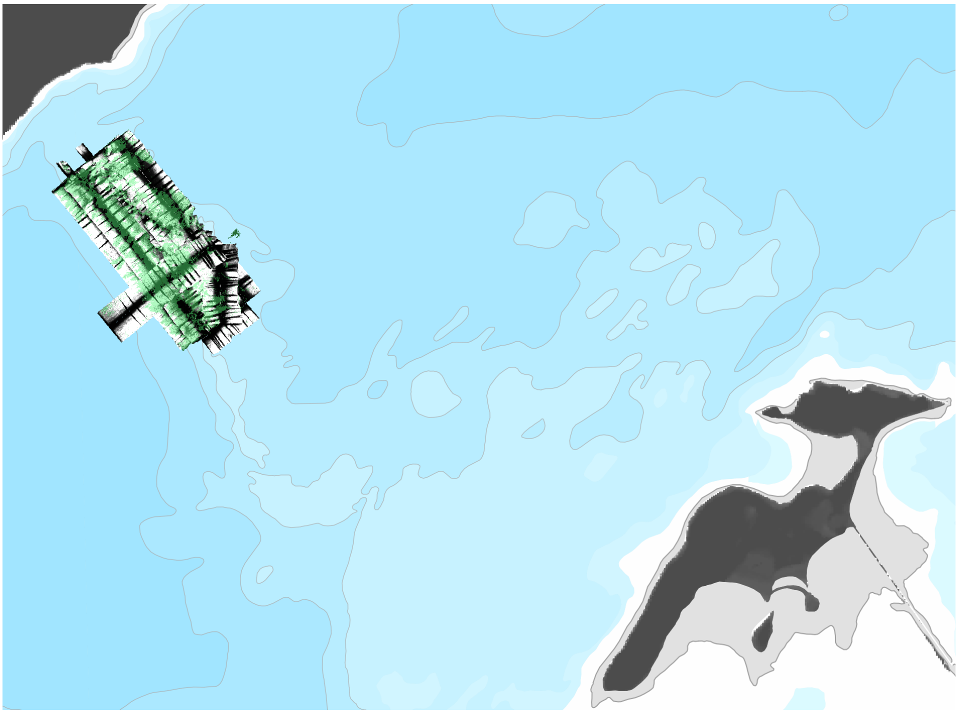


Figure 6.12: Representation of the coral coverage from the 2017 Tautra data set in QGIS. The area covered by the sonar is 1.3 km^2, and 50.9% of it is covered by corals according to the CNN classification.

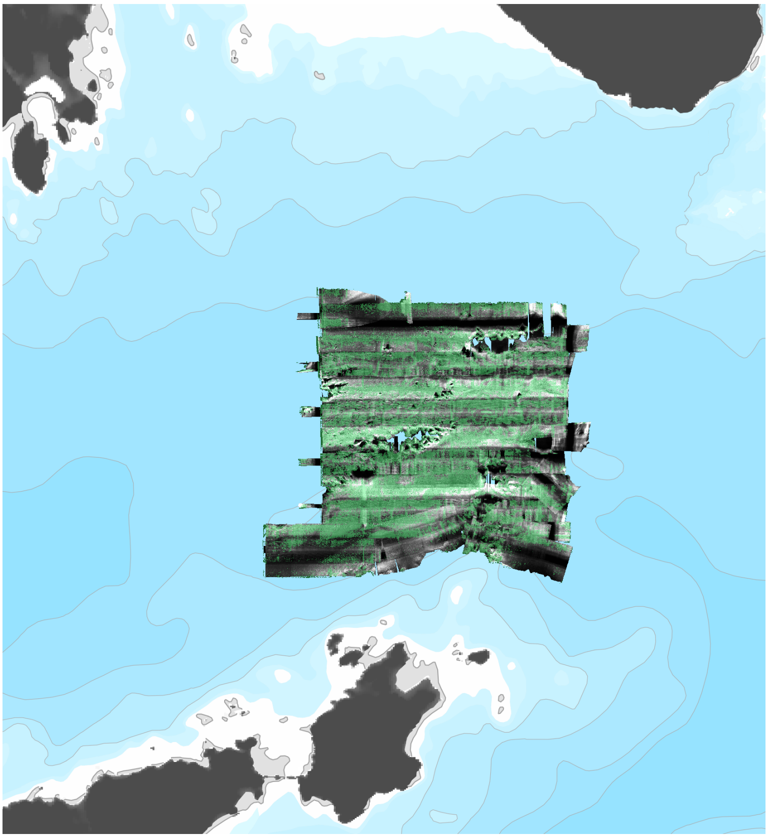


Figure 6.13: Representation of the coral coverage from the Nord-Leksa data set in QGIS. The area covered by the sonar is 4.6 km^2, and 49.4% of it is covered by corals according to the CNN classification.

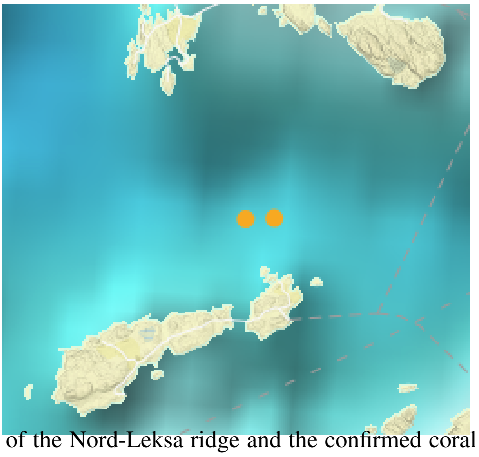


Figure 6.14: Illustration of the Nord-Leksa ridge and the confirmed coral reefs marked by orange dots. The image is collected from the MAREANO project websites [1].

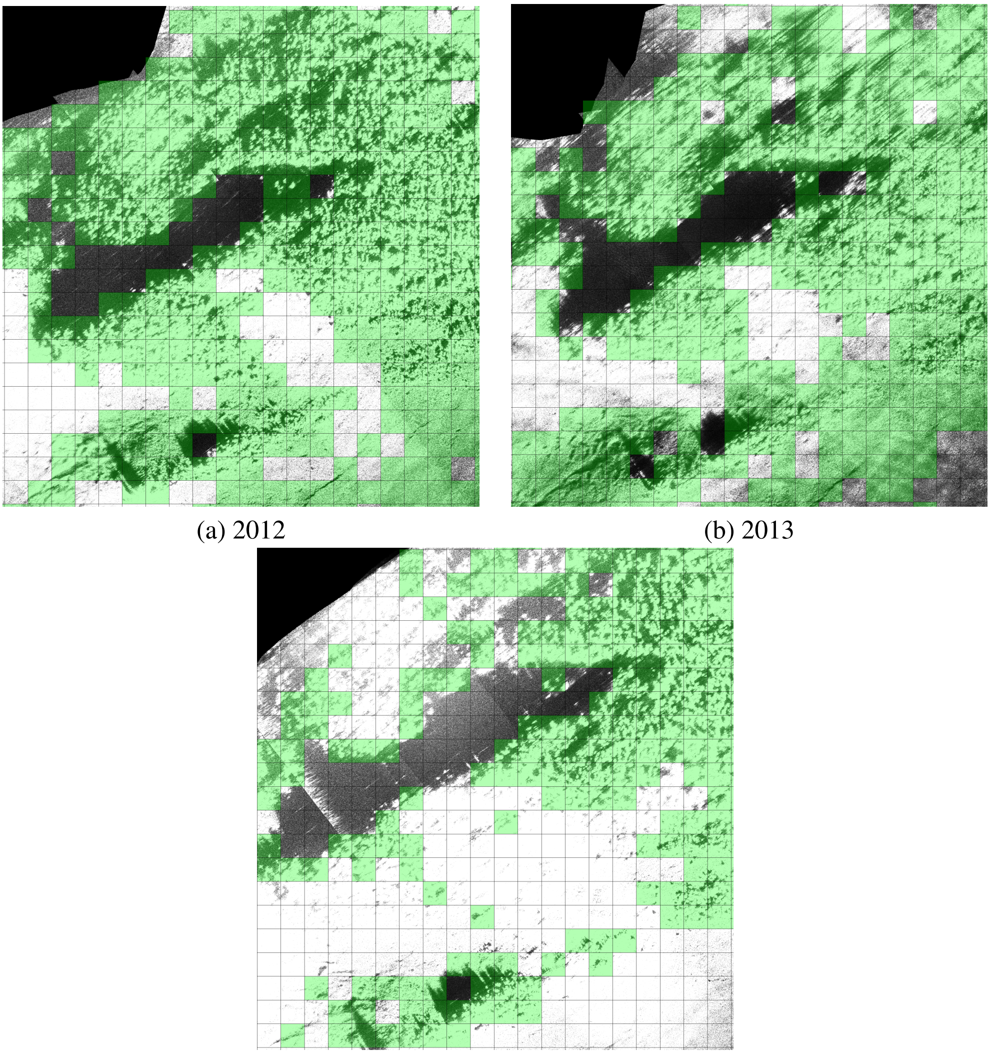
6.3 A Comparison of a region covered in 2012, 2013 and 2017

As the data sets from Tautra did not cover the exact same area, a small region that was covered by each data set was selected for comparison. The region covers approximately $6720m^2$, but it was found in the processing that the 2017 mosaic had a different resolution than the others. When comparing the mosaics, it is evident that the length of the coral mound in the image differ from one another. The 2012 and 2013 versions are however almost identical. Figure 6.15 illustrates the region, showing the same coral mound. By close inspection, it can be noted that the location of the shadow of the mound is not at the same place in the 2017 version as the two others. Notice how far away the large shadow is from the left edge of the image, and how that differs from the 2012 and 2013 images.

The coral coverage of the region is relatively similarly classified in the 2012 and 2013 version of the region, while it differs relatively much on the 2017 version, as shown in both Figure 6.15 and by the results in Table 6.10. According to the image size, they should represent the same 6720 m^2. Some of this area is not representative, as it includes the edge of the image, which varies for all images. A comparison of the length of the coral mound indicate a 20% increase in the 2017 data resolution.

Table 6.10: A comparison of the coral coverage of a $6720m^2$ large region covered in 2012, 2013 and 2017.

Image	**Coral Coverage** [m^2]	**Percent of Total Area**
2012	5024	74.7%
2013	4576	68.1%
2017	2880	42.9%



(a) 2012

(b) 2013

(c) 2017

Figure 6.15: The images show a certain coral mound covered by the sonar for each of the Tautra data sets. Figure (a) is from 2012, b) from 2013 and c) from 2017. Some local variations occur in the images, especially between 2017 and the two others. The reported coral coverage are presented in Table 6.10.

Chapter 7

Discussion

The discussion of this thesis is presented in three parts. The first section is a discussion of the method, including the approach for the optimization of the CNN and the labelling of data. The second is a discussion of the resulting F-scores of the verification images, before and after *closing*. Finally, the quantification of complete data sets is discussed, including the local variations of the Tautra data.

7.1 Discussion of the Method

The labelling of images used for training the CNN was mostly based on the subjective opinion of the author, with some influence from previously performed ground-truthing on the 2012 dataset and some discussion with a geologist. To increase the scientific quality of the results, the labelling should be performed in close cooperation with either an expert on corals in sonar imagery or by ground-truthing with optical imagery of the seafloor.

Most of the labelling was performed while viewing the adjacent sub-images in the data set. As a result, sub-images including only one coral head shape was included in the coral class. A sample of this kind of image can be seen in the top right corner of Figure A.1 in Appendix A.1. After studying the classification performance on the verification images, it becomes evident that these kinds of sample images could have similar characteristics as a rock or some forms of waste in sonar imagery. This might be the cause of many anomalous classifications, where no adjacent sub-images are classified as corals. It is important to note that this uncertainty is connected to the labelling, and not the CNN performance.

Handling data in the sizes represented in this thesis has been challenging in several ways. Processing and structuring the data in numerous stages increases the possibility of losing data, mislabel data or in some way damaging it. The difference in total area between 2012 and 2013 coverage seems to be caused by some missing data, or that the area coverage was increased in 2013 compared to 2012. The discovery was made when it was too late the reproduce the results.

The difference in saturation of the 2017 data compared to 2012 and 2013 is another aspect that might have affected the results, as shown in Figure 6.15. The over-saturation makes textural analysis difficult because of the decreased contrast. It can have contributed to the significant variation of reported coral coverage. The problem was identified early on, and could have been partly mitigated by using the ”average” mapping option in Reflection instead of ”maximum”. This was however not used, because inconsistency in the training data would lead to difficulties when deciding which mosaic option to use on other datasets, e.g. the Nord-Leksa imagery. Therefore, consistent settings were used. It was attempted to include a representative amount of images from 2017 in the training data, to make it better suited for a broader range of sonar image artifacts.

Another option would be to normalize the data using a data augmentation technique available for Keras. It was attempted, but caused the GPU capability of the CNN training to fail, leading to a 20-30% increase in training time. Therefore it was chosen to proceed without the normalization.

Visualization of the filters was used to analyze what the model was learning. The textural analysis performed by the CNN could be visualized by inspecting the filters. This was an important stage of the method because it might explain some of the misclassifications. The example images used in the comparison of section 5.9 show how an image of a coral and one of sandy seafloor would change by the some of the filters in the convolution layers. The coral image has some distinctive round shapes that stand out, while the seafloor image is quite homogeneous except for some small features towards the lower right corner. Those types of features could result in misclassifications if they are in the border-line between coral size and very textured seafloor.

The visualization of the convolution layers in Figures 5.10 and 5.11 showed what type of textures that were identified by the CNN. In the first convolution layer, the texture is coarser for the coral image than the seafloor image. Typically the first layer woould find edges in the image, which is especially clear in the lowest image of the coral example in Figure 5.10. The image would blur more and more for the deeper layers, which is expected because of the maxpooling and activation in between. Even though it is possible to visualize the filters, it is more difficult to determine which filters the network use to classify the images.The visualizations of the activation layers might tell more about what features the network activate on. The visualization of the network was an important tool for understanding how the CNN process the data. Many false positive classifications are typically rough textured seafloor images, which is easier to understand after a visual inspection of the filters.

The combination of using grid search and random search to find a suitable CNN architecture and its hyper-parameters was successful according to the goals. The target was to reach a reported accuracy of 95% on the test data, and $94.9 \approx 95\%$ was reported. However, due to the random initialization of network weights and random shuffling of training data before each epoch, some uncertainty has to be taken into account when reporting the final accuracy. A variation of ± 2-3% is expected in the reported CNN accuracy, based on the experience gained while optimizing the CNN.

The reported accuracy of the verification images indicates that the method is of high quality, and has a large potential for further use.

7.2 Accuracy on the Verification Images

The F-scores were calculated based on how the labelling was performed, thus calculated based on the subjective opinion of the author of this thesis. The level of uncertainty is difficult to determine, but due to the numerous hours spent on labelling, the accuracy of counting false positives and false negatives are considered to be relatively high.

It is questionable if using one image from each of the 40, 30 and 60 mosaics for the 2012, 2013 and 2017 makes representative estimates of the classification performance of the method. The manual work of classifying images containing over 5500 sub-images of 100x100 pixels was time-consuming, and adding more images to verify the performance on each dataset was not feasible within the given time frame of this thesis. It is however considered that all of the images were created using the same sonar and same settings, and should therefore represent similar characteristics of performance. They all report F-scores that are relatively high considering the complexity of sonar imagery. All but the 2012 example reach above the target of 85% in F-score.

The verification image from Nord-Leksa is the most representative image on the actual performance of the classification method, as no image from that area was used to form the training data. The F-score was calculated to 88.7%, which is above the stated target of 85%.

Manual interpretation of sonar imagery is tedious and time-consuming work illustrating the need of a method such as the one presented in this thesis. A more extensive selection of verification images would have increased the certainty of the actual performance, but the results reported on the verification images in this thesis gives an estimate of the performance of the classification method in general.

7.2.1 Effect of Closing

As a result of using *closing*, the precision increased while the recall decreased for all of the verification images. This happened in various degrees, the most affected case being the 2012 verification image, and least affected being the Nord-Leksa image. Generally, the number of false negatives increased, which entails that fewer coral images was identified. The recall of the 2012 verification image decreased by 16% after closing, while the Nord-Leksa recall decreased by 2 %. On the other hand, the precision increased by 4.7% and 4.6% for the 2012 and Nord-Leksa images respectively. Because of the large variation of the effect on the recall, *closing* was not included for the quantification of the entire datasets.

As a side note, *closing* might be an efficient tool to use for visualization of trends of coral reefs, as the precision is maintained or somewhat increased. The large decrease in recall might have a connection to the labelling as well, as many examples of sub-images with only one significant coral-like shape in them are causing many solitary coral classifications. As these are labelled corals in the training set, they have to be marked as real corals in the verification images. If the labelling of corals had not included this form of imagery, the effect of *closing* on the recall might have been smaller.

7.3 Quantification of Complete Data Sets

The F-scores from Tautra show that even though the data set from Tautra in 2013 covers a larger area than the data set from 2012, the reported coral coverage is lower. The difference in area coverage makes it difficult to compare the results numerically. However, by visual interpretation of the coral coverage in Figure 6.10 and 6.11, it is possible to see that the trending areas of coral follow a wide line across the sonar image tracks. In the 2012 data, it is also possible to see that the end of the mosaic furthest away from the Tautra island includes a lot of coral classifications. This can indicate that the differences might not be that much between 2012 and 2013, as much of the coral in the 2012 data appear in areas not covered by 2013.

Even though the 2017 data differ in width and length from the 2012 and 2013 areas, the same trending line of coral coverage is visible in Figure 6.12 as well. It corresponds well with the two other data sets, as well as the reports from MAREANO in Figure 6.9, which is the only validating comparison available in this thesis.

A quantitative comparison of the Tautra datasets was not performed due to the variations in total coverage. It would have been useful to create a comparable selection of the 2012 data set covering the same area as the available data from 2013. This was not acknowledged in time to produce the results. The 2017 dataset, has proved to differ in both resolution and coverage, making it unsuitable for a quantitative comparison with the two other datasets.

7.3.1 Comparison of 2012, 2013 and 2017 Data from Tautra

The comparison of the small region made in Section 6.3 illustrates some image artifacts that can cause differences in the classification of coral in the different data sets. Distorted and over-saturated imagery are examples of image artifacts, which is illustrated in Figure 6.15. Sub-figure (b) has a much higher level of distorted imagery toward the upper right part of the image than in (a), causing fewer coral sub-images to be classified in the region. Figure 6.15 (c) is over-saturated, resulting in a low contrast that makes textural analysis difficult. The difference between the 2012 and 2013 version of Figure 6.15 is only 6.6%, while the difference between 2012 and 2017 is 31.8%. These results are most likely caused by the image artifacts. Notice that the image is close to the edge of the swath. Although SAS, in theory, has range independent along-track and across-track resolution, there is a tendency in the datasets and the example shown in Figure 6.15, that the amount of image artifacts are connected to the distance from a mosaic edge.

Areas of coral reefs typically have large mound-like structures that might cause temporary and sudden variations in heave and pitch as the AUV attempts to maintain a constant altitude above the seabed. These sudden movements are likely to affect the sonar imagery. By comparison to Nord-Leksa, there are significantly more disturbances in the Tautra imagery than Nord-Leksa. By studying the reported depths of the area covered in both locations, it is evident that the Nord-Leksa bathymetry is more uniform than at Tautra. The uneven bathymetry at Tautra might be one of the main causes a higher level of distorted imagery.

Chapter 8

Conclusion

The combination of synthetic aperture sonar imagery and convolutional neural networks were used to create a classification model. The reported accuracy of the trained model was 95%, while the accuracy measured by F-score on four verification images varied between 83 % and 95%.

The corals in the sonar imagery were subjectively labelled. This affected how the false negatives and positives were identified, leading to possible inaccuracies of the reported performance.

The results illustrate that the classification method has potential for further use, except for that the level of quality of the labelling should be increased. It is suggested that the labelling should be performed by or in close cooperation with an expert on corals in sonar imagery or by ground-truthing to create a data set to use for training a CNN. The classification method successfully calculated the coral coverage of the sonar mosaics, and maps were made for visualization.

A combination of a grid search and a random search was successfully used to optimize the hyperparameters for the CNN. The optimization showed that it was possible to reach the goal of classification accuracy of 95% based on the test data set. Two sources of error on the reported test accuracy was discovered, namely the random initialization of weights and random shuffling of training data between each epoch.

The F-score on the verification images from Tautra differed between 83% and 95%, while the Nord-Leksa verification image had an accuracy of 89%. Generally, the precision was higher than the recall, implying that there are more false negatives than false positive classifications of corals. This will result in an overestimate of coral coverage, relative to the labelling performed for the training data.

The use of the image processing technique *closing* increased the precision on each verification image and decreased the recall considerably. Therefore, the use of this technique was abandoned before calculating the total coral coverage in each data set. A quantitative comparison of the Tautra data sets was not performed because of the variations of spatial coverage.

The accuracy of coral coverage with respect to the real presence of coral reefs could not be defined, as ground-truthing or other verification methods were too time-consuming to performed as a part of this thesis. However, the MAREANO project has produced maps with accurate locations of verified coral reefs. The results of quantification and presentation of coral coverage for all of the data sets show trends that dense areas of coral classifications have comparable locations with the coral reefs reported by MAREANO.

The methodology presented in this thesis is considered adaptable for further work on similar problems, with some minor modifications in the optimization part that resolves the issue of random initialization of weights and shuffling of training data. Additionally, the labelling of training data has to be performed with a higher level of certainty.

Chapter 9

Further Work

An important benefit of the high resolution of SAS imagery is that it can be used for change detection. The navigational data from the AUV is not accurate enough to do this, but by co-registration of features in SAS images taken in the same area, it is possible to eliminate the positioning error between the images [52]. The classification method presented in this thesis could form a basis for a change detection method.

A method for increasing the training data sizris to implement data augmentation to perform affine transformations on the imagery. Additionally it can be used to normalize the input data. This was attempted but failed to be compatible with the GPU accelerated training. By resolving this issue, it would make it possible to use more classes in the training of a CNN because of the increased dataset size. Especially separation of high and low-quality coral should be further investigated. Another aspect that would provide useful information would have been to collect the output of the softmax activation of the final layer, which outputs the statistical value of the prediction of the CNN. Another aspect to investigate further is to normalize the mean of the images, in an attempt at making the CNN generalize images better.

A lot of coral area was not quantified due to shadow caused by coral mounds. A method for estimation of the corals in the shadow should be included if the quantification of coral coverage is to be more accurate.

The research within artificial neural network is continuous, and new methods are still developed [20]. There might be other networks than CNN that could work for classification of corals in sonar imagery.

Bibliography

[1] Pål Buhl-Mortensen. Korallrev. http://www.mareano.no/tema/koraller/korallrev.

[2] Xavier Lurton and Geoffroy Lamarche. Backscatter Measurements by Seafloor-Mapping Sonars. GeoHab Working Group, May 2015.

[3] Alexandre C. G. Schimel, Terry R. Healy, David Johnson, and Dirk Immenga. Quantitative experimental comparison of single-beam, sidescan, and multibeam benthic habitat maps. ICES Journal of Marine Science, 67(8):1766–1779, 2010.

[4] Roy Edgar Hansen. Sonar Systems. Intech open, 2011. Introduction to Synthetic Aperture Sonar.

[5] Saleh Albelwi and Ausif Mahmood. A framework for designing the architectures of deep convolutional neural networks. Entropy, 19, 2017.

[6] Rafael C. Gonzalez and Richard E. Woods. Digital Image Processing. Pearson Educational International, 3rd edition, 2010.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

[8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan

Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.

[9] Pål B. Mortensen, T. Hovland, Jan Helge Fosså, and Dag M. Furevik. Distribution, abundance and size of lophelia pertusa coral reefs in mid-norway in relation to seabed characteristics. Journal of the Marine Biological Association of the United Kingdom, 81(4):581–597, 2001.

[10] J. Murray Roberts, Andrew J. Wheeler, and André Freiwald. Reefs of the deep: The biology and geology of cold-water coral ecosystems. Science, 312(5773):543–547, 2006.

[11] J.H. Fosså, P.B. Mortensen, and D.M. Furevik. The deep-water coral lophelia pertusa in norwegian waters: distribution and fishery impacts. Hydrobiologia, 471(1):1–12, Mar 2002.

[12] Sandra Brooke and Johanna Järnegren. Reproductive periodicity of the scleractinian coral lophelia pertusa from the trondheim fjord, norway. Marine Biology, 160(1):139–153, Jan 2013.

[13] Martin Ludvigsen, Geir Johnsen, Asgeir J. Sørensen, Petter A. Lågstad, and Øyvind Ødegå rd. Scientific operations combining rov and auv in the trondheim fjord. Marine Technology Society Journal, 48(2):59–71, 2014.

[14] Anette C. Elde, Ragnhild Pettersen, Per Bruheim, Johanna Järnegren, and Geir Johnsen. Pigmentation and spectral absorbance signatures in deep-water corals from the trondheimsfjord, norway. Marine Drugs, 10(6):1400–1411, 2012.

[15] Martin Ludvigsen and Asgeir J. Sørensen. Towards integrated autonomous underwater operations for ocean mapping and monitoring. Annual Reviews in Control, 42(Supplement C):145 – 157, 2016.

[16] Markus Diesing and Terje Thorsnes. Mapping of cold-water coral carbonate mounds based on geomorphometric features: An object-based approach. Geosciences, 8(2), 2018.

[17] Pål Buhl Mortensen, Hans Tore Rapp, and Ulf Båmstedt. Oxygen and carbon isotope ratios related to growth line patterns in skeletons of lophelia pertusa (l) (anthozoa, scleractinia): Implications for determination of linear extension rate. Sarsia, 83(5):433–446, 1998.

[18] Ivor Marsh and Colin Brown. Neural network classification of multibeam backscatter and bathymetry data from stanton bank (area iv). Applied Acoustics, 70(10):1269 – 1276, 2009. The Application of Underwater Acoustics for Seabed Habitat Mapping.

[19] V.A.I. Huvenne, Ph. Blondel, and J.-P. Henriet. Textural analyses of sidescan sonar imagery from two mound provinces in the porcupine seabight. Marine Geology, 189(3):323 – 341, 2002.

[20] I. Arel, D. C. Rose, and T. P. Karnowski. Deep machine learning - a new frontier in artificial intelligence research [research frontier]. IEEE Computational Intelligence Magazine, 5(4):13–18, Nov 2010.

[21] Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. Institute of Electrical and Electronics Engineers, Inc., August 2003.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[23] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In 2011 International Conference on Document Analysis and Recognition, pages 1135–1139, Sept 2011.

[24] Margrete Scheide. Classification of cold-water coral reefs in sonar imagery using convolutional neural networks. Project Thesis, December 2017.

[25] Jens M. Hovem. Marine Acoustics - The Physics of Sound in Underwater Environments. Peninsula Publishing, Los Altos Hills, California, 1 st ed. edition, 2012.

[26] Philippe Blondel. The handbook of sidescan sonar, 2009.

[27] Roy-Edgar Hansen. Synthetic aperture sonar technology review. Marine Technology Society Journal, 47(5):117–127, 2013.

[28] A. Bellettini and M. A. Pinto. Theoretical accuracy of synthetic aperture sonar micronavigation using a displaced phase-center antenna. IEEE Journal of Oceanic Engineering, 27(4):780–789, Oct 2002.

[29] Louis J. Cutrona. Comparison of sonar system performance achievable using synthetic-aperture techniques with the performance achievable by more conventional means. The Journal of the Acoustical Society of America, 58(2):336–348, 1975.

[30] B De Mol, P Van Rensbergen, S Pillen, K Van Herreweghe, D Van Rooij, A McDonnell, V Huvenne, M Ivanov, R Swennen, and J.P Henriet. Large deep-water coral banks in the porcupine basin, southwest of ireland. Marine Geology, 188(1):193 – 231, 2002.

[31] Thiago B. S. Correa, Mark Grasmueck, Gregor P. Eberli, Klaas Verwer, and Samuel J. Purkis. Deep Acoustic Applications, pages 253–282. Springer Netherlands, Dordrecht, 2013.

[32] Craig J. Brown and Philippe Blondel. Developments in the application of multibeam sonar backscatter for seafloor habitat mapping. Applied Acoustics, 70(10):1242 – 1247, 2009. The Application of Underwater Acoustics for Seabed Habitat Mapping.

[33] T. Kohonen. The self-organizing map. Proceedings of the IEEE, 78(9):1464–1480, Sep 1990.

[34] J. Chanussot, F. Maussang, and A. Hetet. Scalar image processing filters for speckle reduction on synthetic aperture sonar images. In OCEANS '02 MTS/IEEE, volume 4, pages 2294–2301 vol.4, Oct 2002.

[35] P. Blondel, J. C. Sempere, and V. Robigou. Textural analysis and structure-tracking for geological mapping: applications to sonar images from endeavour segment, juan de fuca ridge. In Proceedings of OCEANS '93, pages III/208–III/213 vol.3, Oct 1993.

[36] Philippe Blondel. Segmentation of the mid-atlantic ridge south of the azores, based on acoustic classification of tobi data. Geological Society, London, Special Publications, 118(1):17–28, 1996.

[37] P. Blondel and B.J. Murton. Handbook of seafloor sonar imagery. Wiley-Praxis series in remote sensing. Wiley published in association with Praxis Pub., 1997.

[38] A. B. Simmons and S. G. Chappell. Artificial intelligence-definition and practice. IEEE Journal of Oceanic Engineering, 13(2):14–42, Apr 1988.

[39] A. Gulli and S. Pal. Deep Learning with Keras. Packt Publishing, 2017.

[40] S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, third edition, 2010.

[41] Shan Suthaharan. Support Vector Machine, chapter 7-12, pages 145–306. Springer US, Boston, MA, 2016.

[42] Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85 – 117, 2015.

[43] Andrej Karpathy. Lecture notes in cs231n convolutional neural networks for visual recognition. `http://cs231n.github.io/`, 2018.

[44] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. CoRR, abs/1801.05134, 2018.

[45] Leslie N. Smith. No more pesky learning rate guessing games. CoRR, abs/1506.01186, 2015.

[46] M. Ludvigsen, S. M. Albrektsen, K. Cisek, T. A. Johansen, P. Norgren, R. Skjetne, A. Zolich, P. Sousa Dias, S. Ferreira, J. B. de Sousa, T. O. Fossum, Ø. Sture, T. Røbekk Krogstad, Ø. Midtgaard, V. Hovstein, and E. Vågsholm. Network of heterogeneous autonomous vehicles for marine research and management. In OCEANS 2016 MTS/IEEE Monterey, pages 1–7, Sept 2016.

[47] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688, May 2016.

[48] Martín Abadi, Ashish Agarwal, and Paul Barham et. al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[49] François Chollet et al. Keras. `https://keras.io`, 2015.

[50] Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[51] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[52] Roy Edgar Hansen, Torstein Sæbø, Ole Jacob Lorentzen, and Øivind Midtgaard. Change detection in topographic structures using interferometric synthetic aperture sonar. In In Proceedings of the 2st Underwater Acoustic Conference, Rhodes, Greece, 2014.

Appendix

A.1 Examples of Coral Labelling

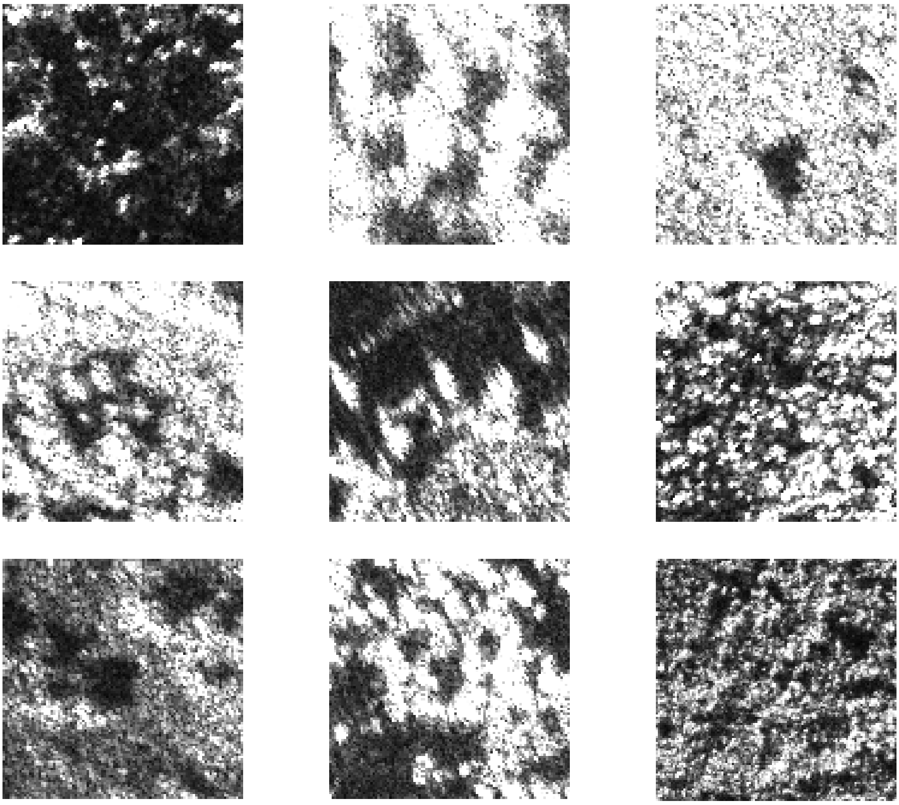


Figure A.1: A representative collection of what kind of sonar sub-images were labelled as corals. The samples presented has 100x100 pixels with 4x4cm resolution.

A.2 Test Set of Sonar Imagery Used in Optimization

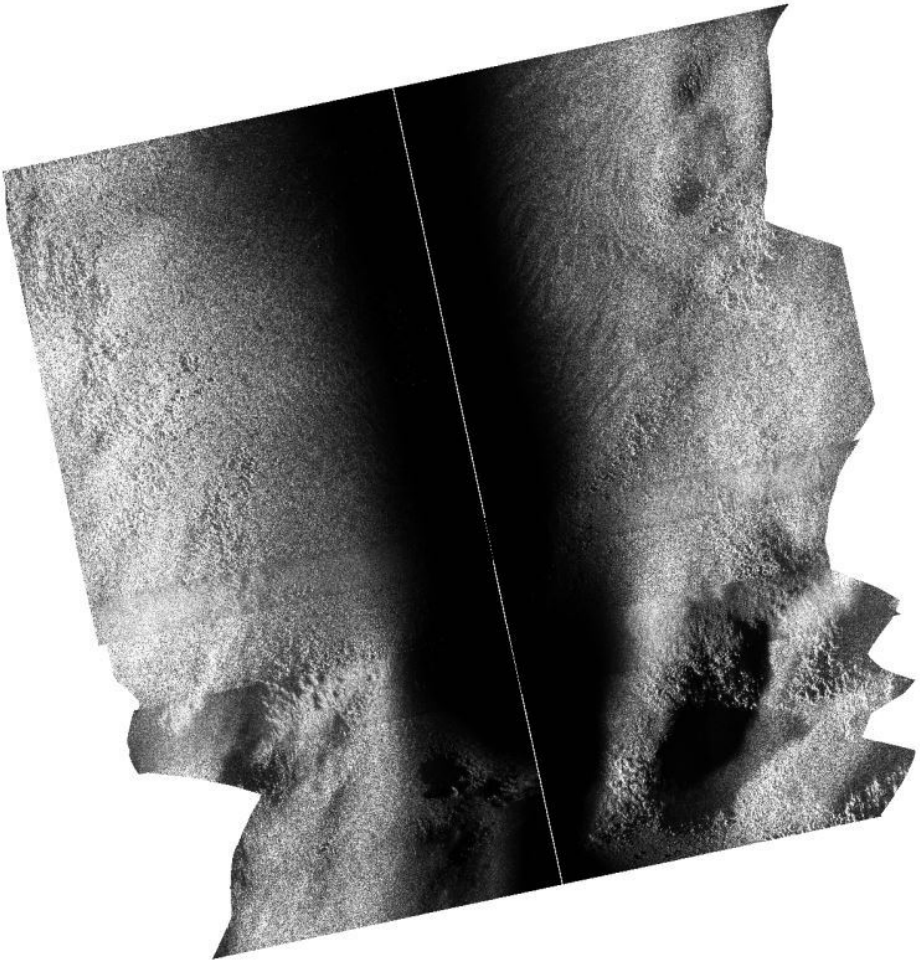


Figure A.2: Verification image from the 2012 data set collected at the Tautra Ridge. The resolution is 4x4 cm, and the image covers an area of 330x325m ≈ 0.1 km^2.

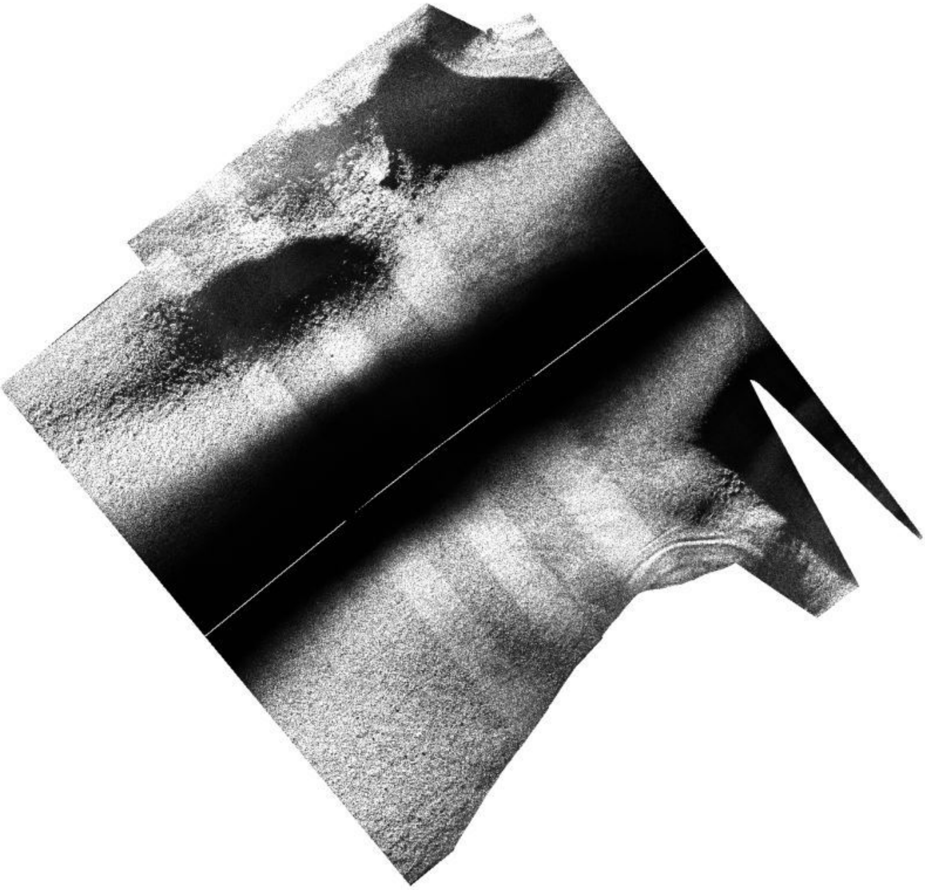


Figure A.3: Verification image from the 2013 data set collected at the Tautra Ridge. The resolution is 4x4 cm, and the image covers an area of 330x300m ≈ 0.1 km^2.

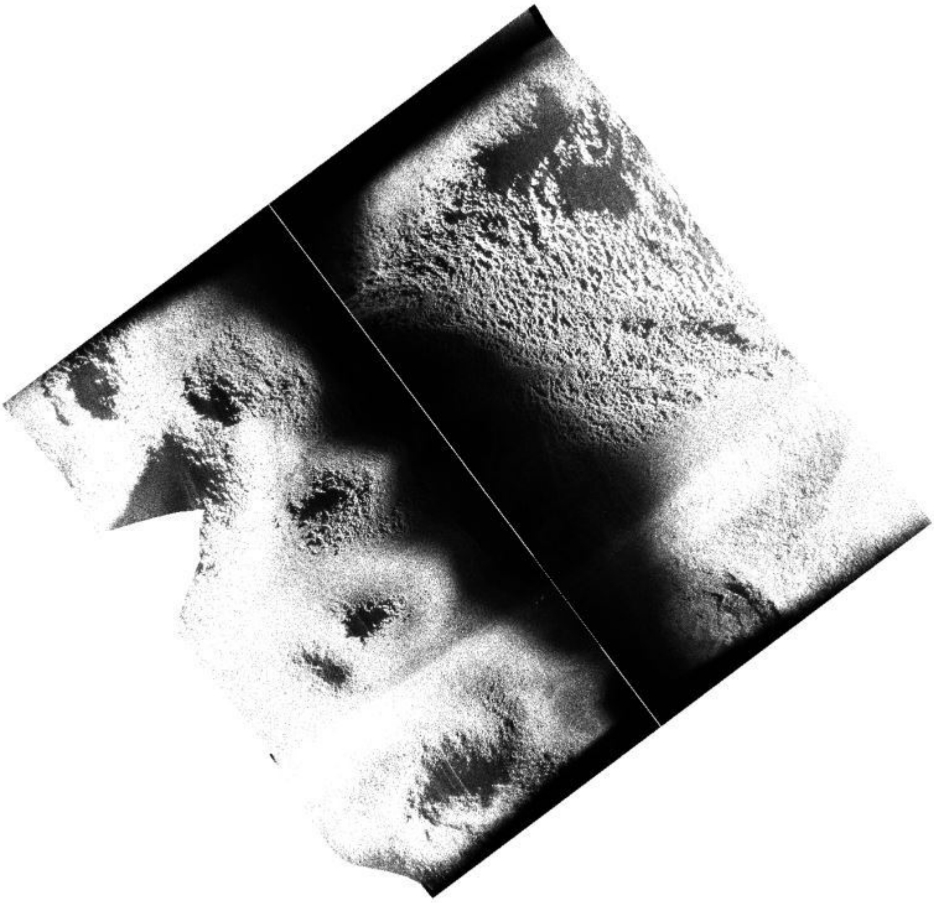


Figure A.4: Verification image from the 2017 data set collected at the Tautra Ridge. The resolution is 4x4 cm, and the image covers an area of 300x295m ≈ 0.09 km^2.

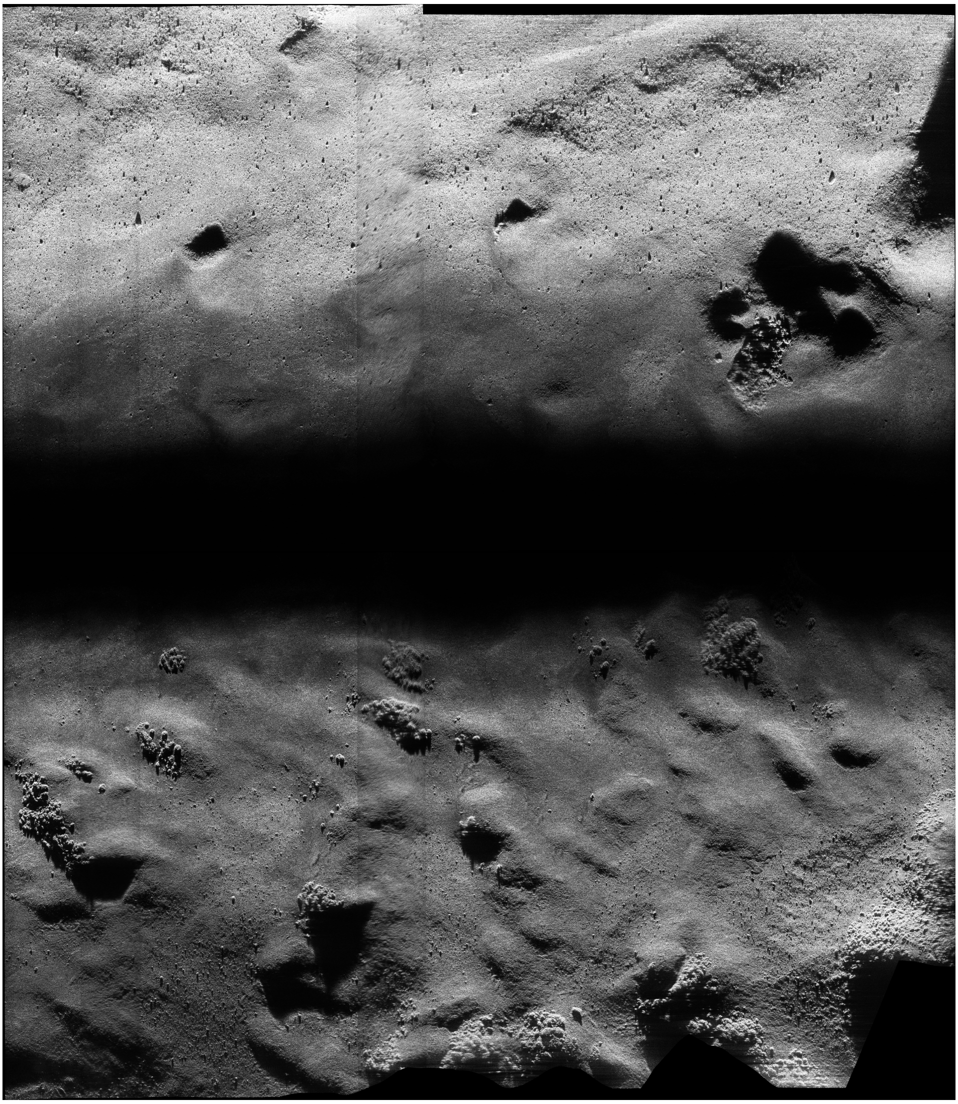


Figure A.5: Verification image from the Nord-Leksa data set. The resolution is 4x4 cm, and the image covers an area of 295x340m ≈ 0.1 km^2.

A.3 Optimizing Number of Categories

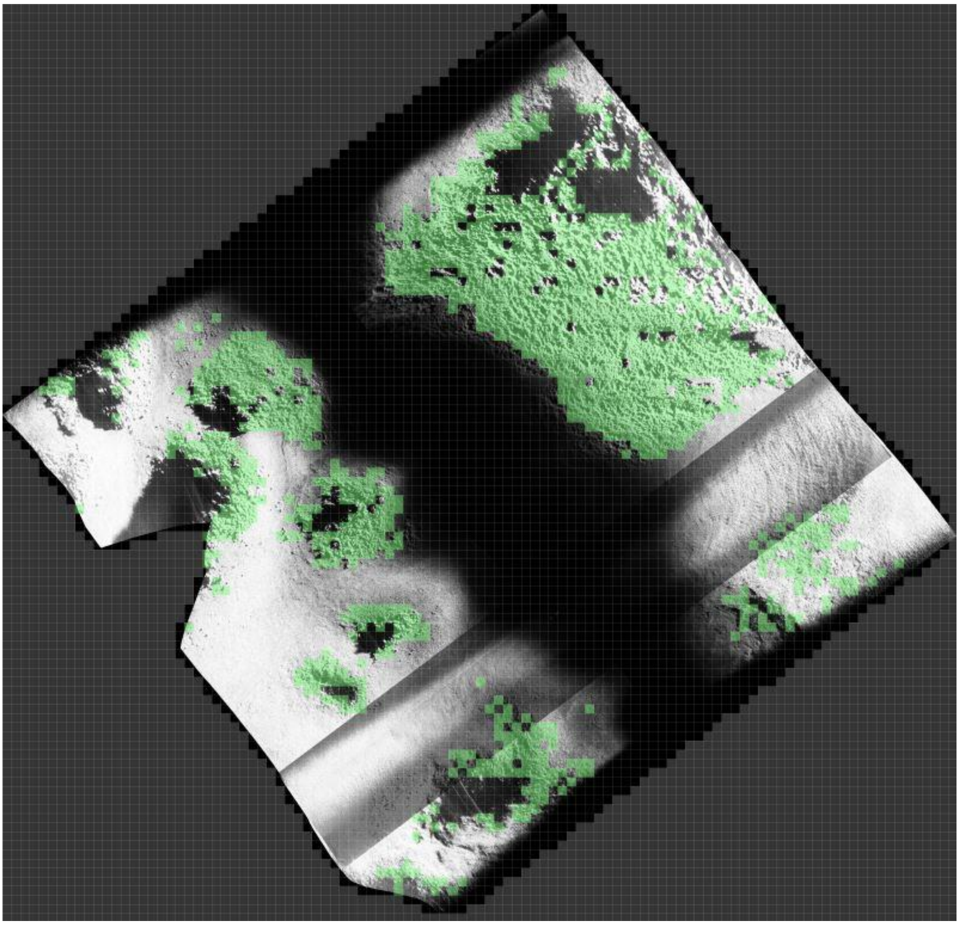


Figure A.6: Result of classification with a CNN trained on two categories, where the corals are only high quality. Test image is from 2017.

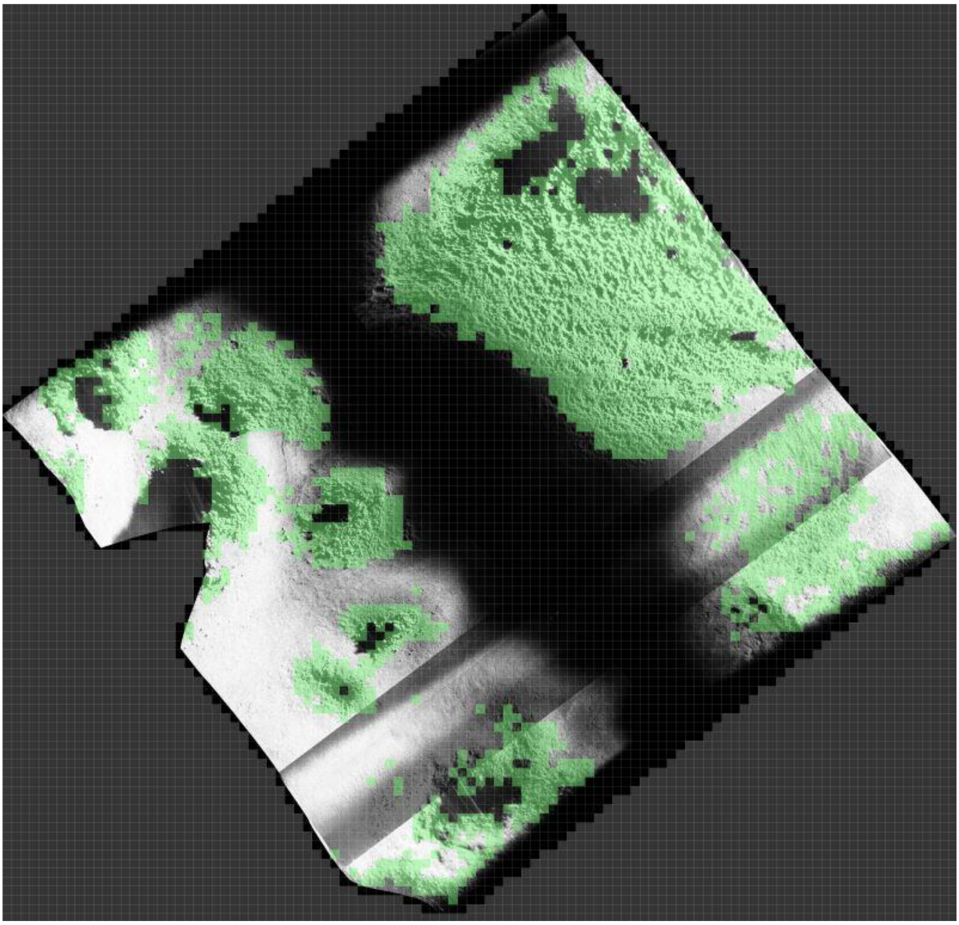


Figure A.7: Result of classification with a CNN trained on two categories, where the coral class include shadowy images of corals. Test image is from 2017.

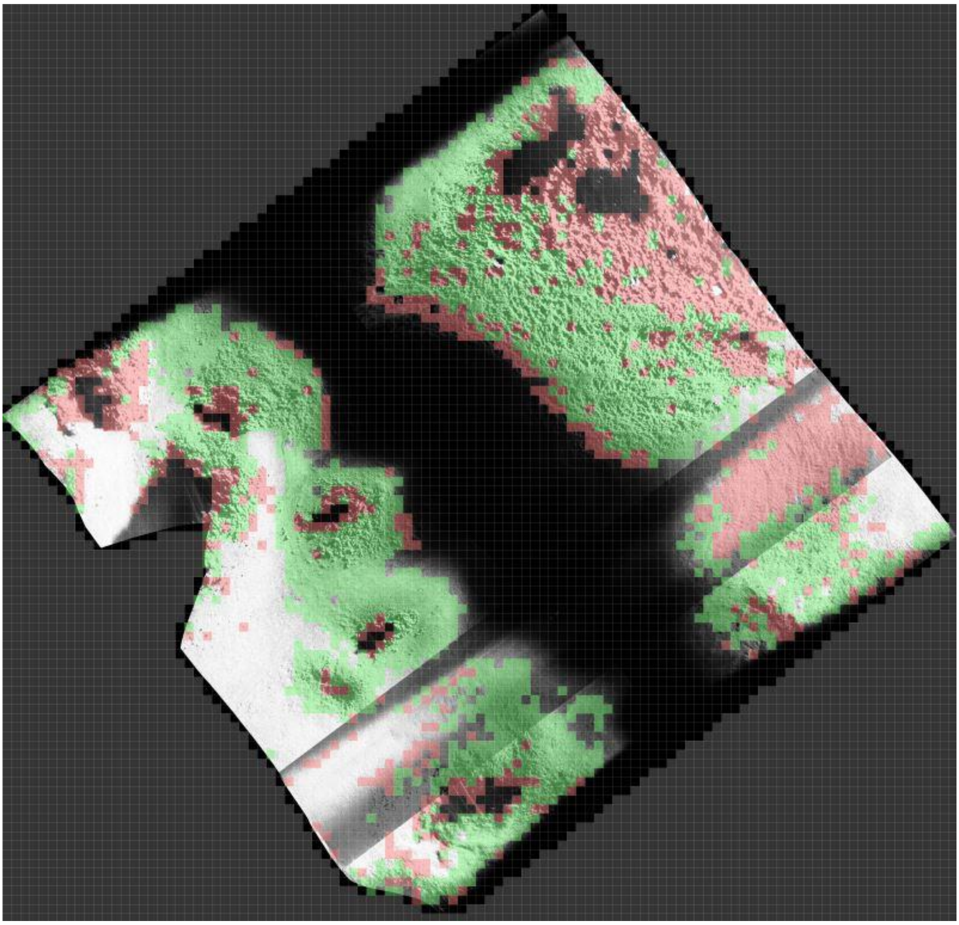


Figure A.8: Result of classification with a CNN trained on three categories. Test image is from 2017.

A.4 Results of Optimizing Dropout Rate

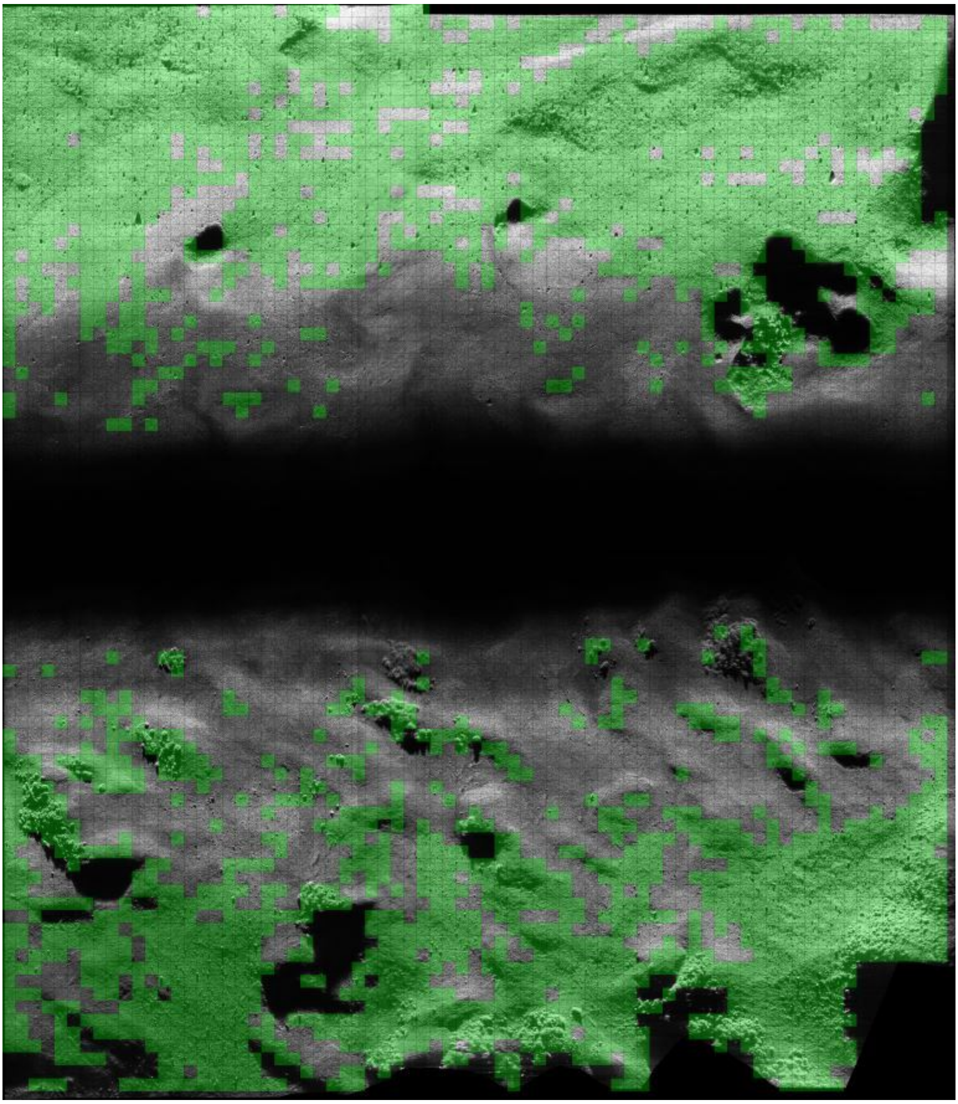


Figure A.9: The result of classification using CNN version Nr. 42 with a dropout ratio of 0.25. The rest of the hyper-parameters used are based the results from the grid search and random search (see Section 5.8).

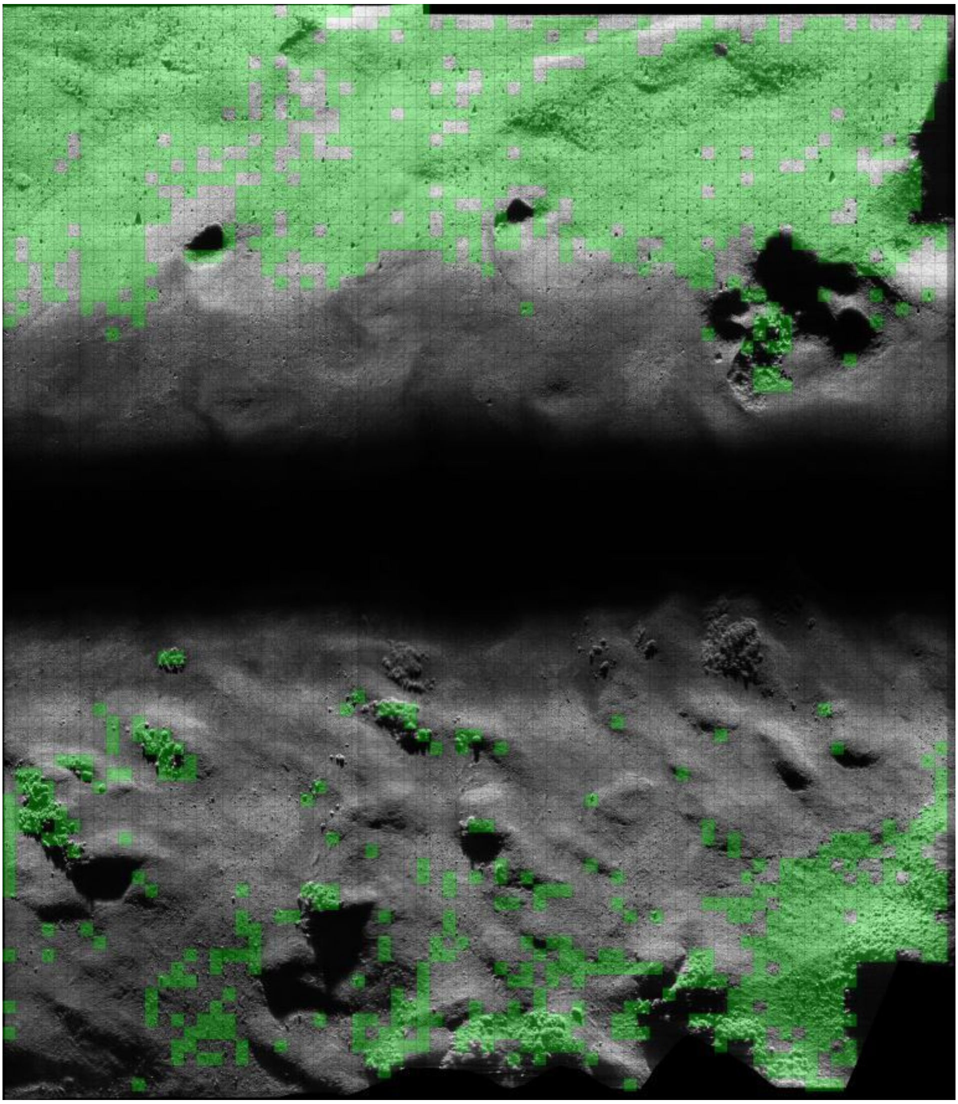


Figure A.10: The result of classification using CNN version Nr. 42 with a dropout ratio of 0.5. The rest of the hyper-parameters used are based on the results from the grid search and random search (see Section 5.8).

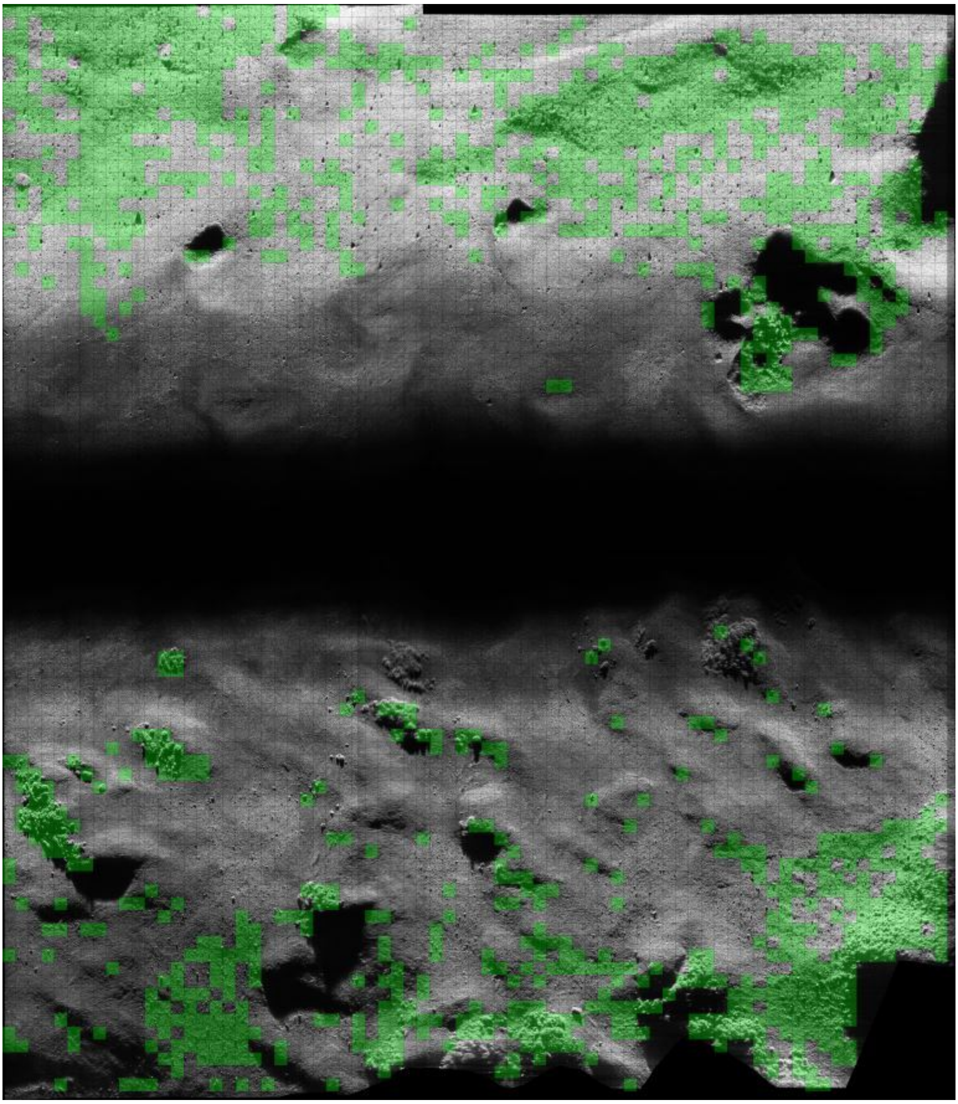


Figure A.11: The result of classification using CNN version Nr. 2 with a dropout ratio of 0.25. The rest of the hyper-parameters used are based on the results from the grid search and random search (see Section 5.8).

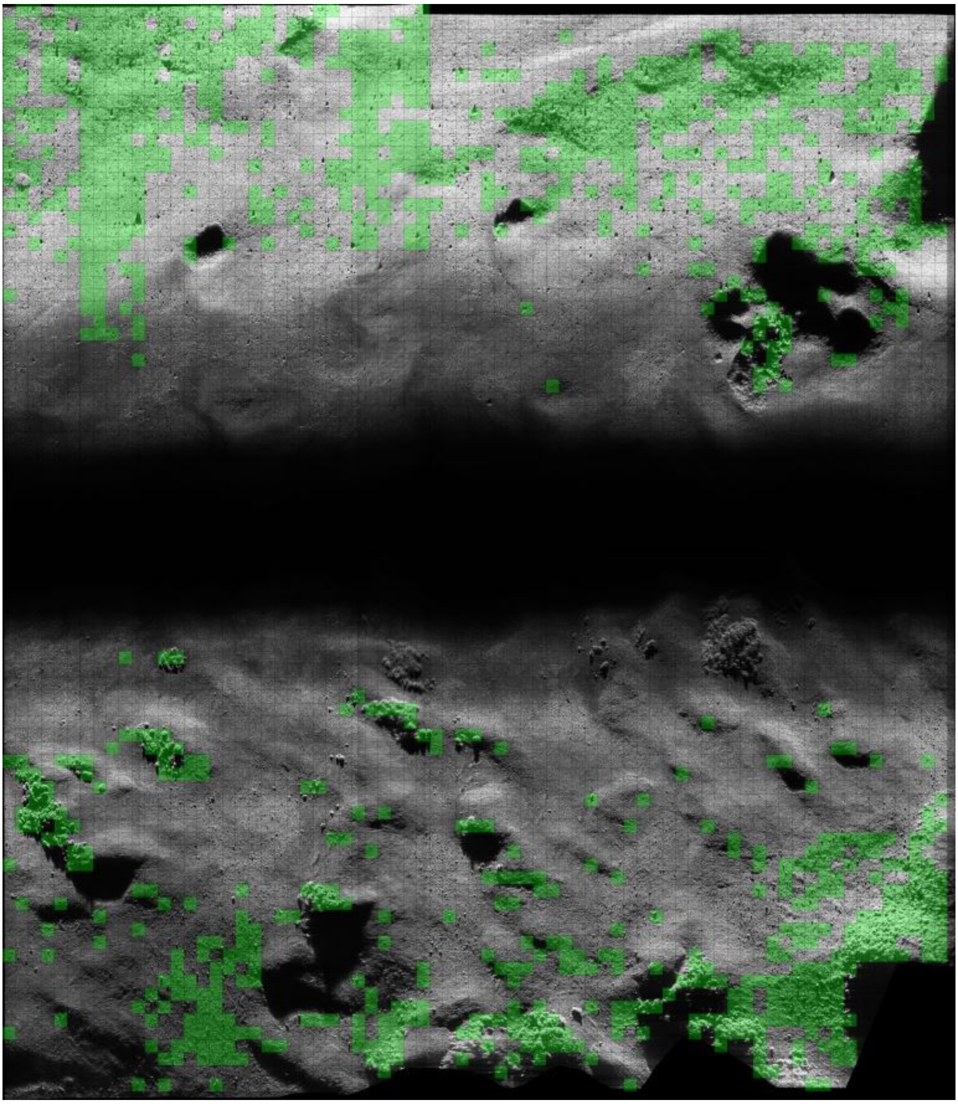


Figure A.12: The result of classification using CNN version Nr. 2 with a dropout ratio of 0.5. The rest of the hyper-parameters used are based on the results from the grid search and random search (see Section 5.8).

A.5 Results of Classifying Data Sets

A.5.1 The Tautra Data

2012

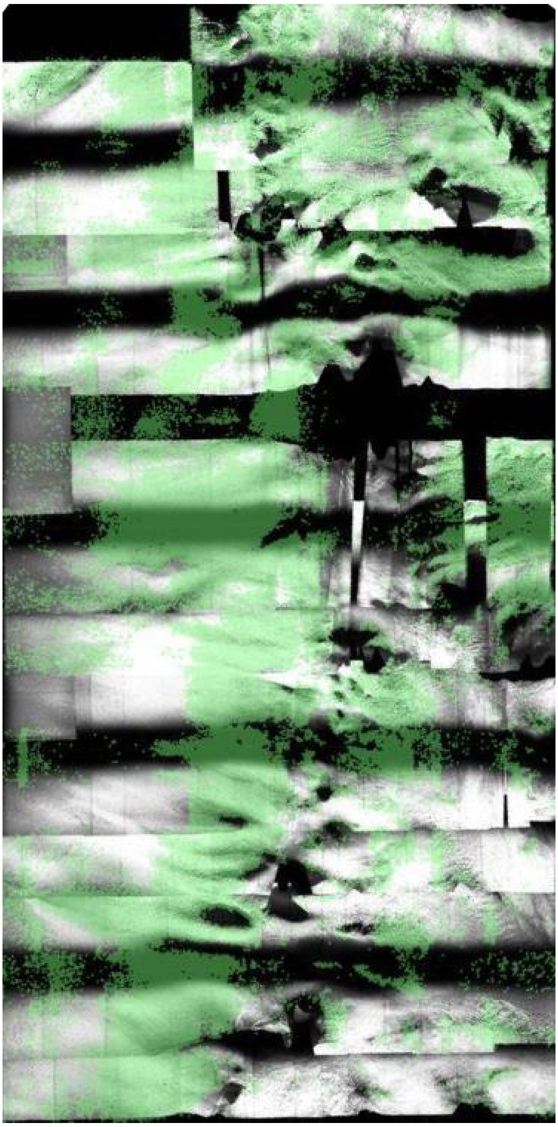


Figure A.13: Mosaic in 33x33 cm resolution with the ”average” option for mosaicing in Reflection, layered with the classification result created by the entire 4x4 cm resolution data set.

2013

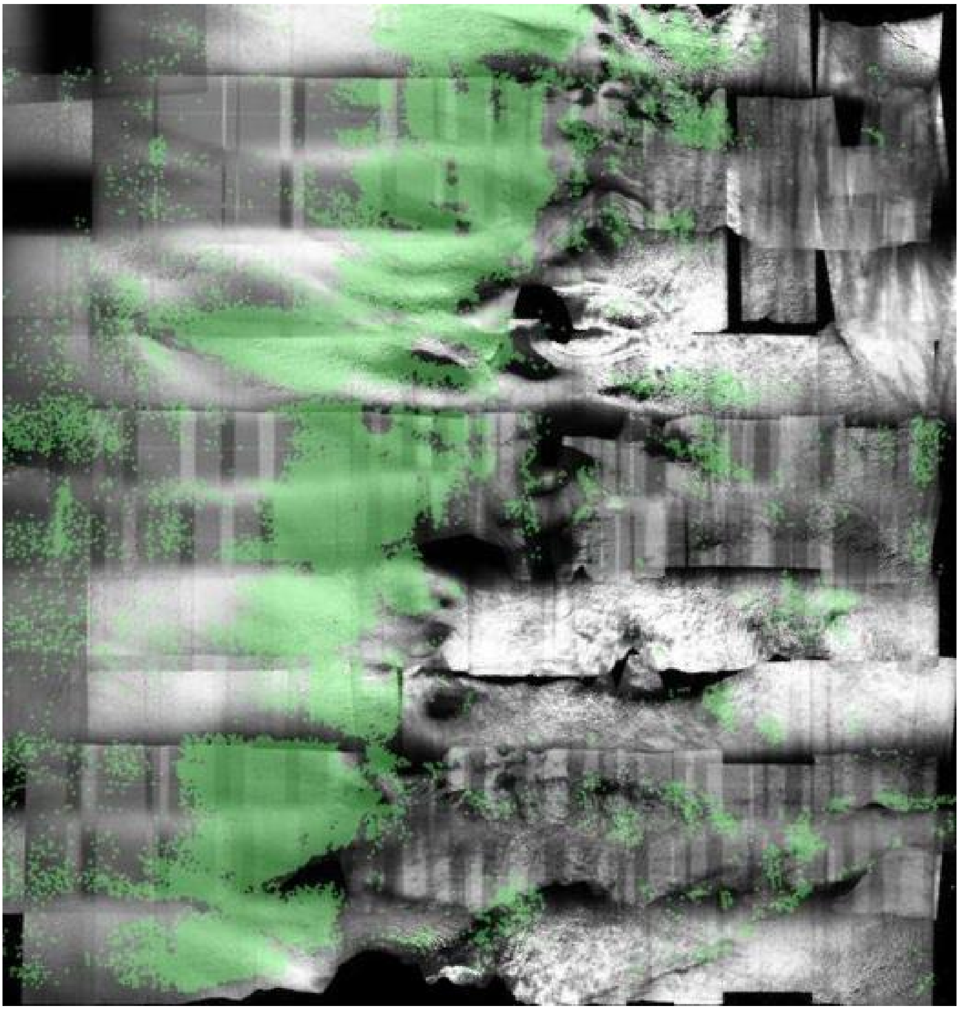


Figure A.14: Mosaic in 33x33 cm resolution of the Tautra ridge in 2013 with the ”average” option for mosaicing in Reflection, layered with the classification result created by the entire 4x4 cm resolution data set.

2017

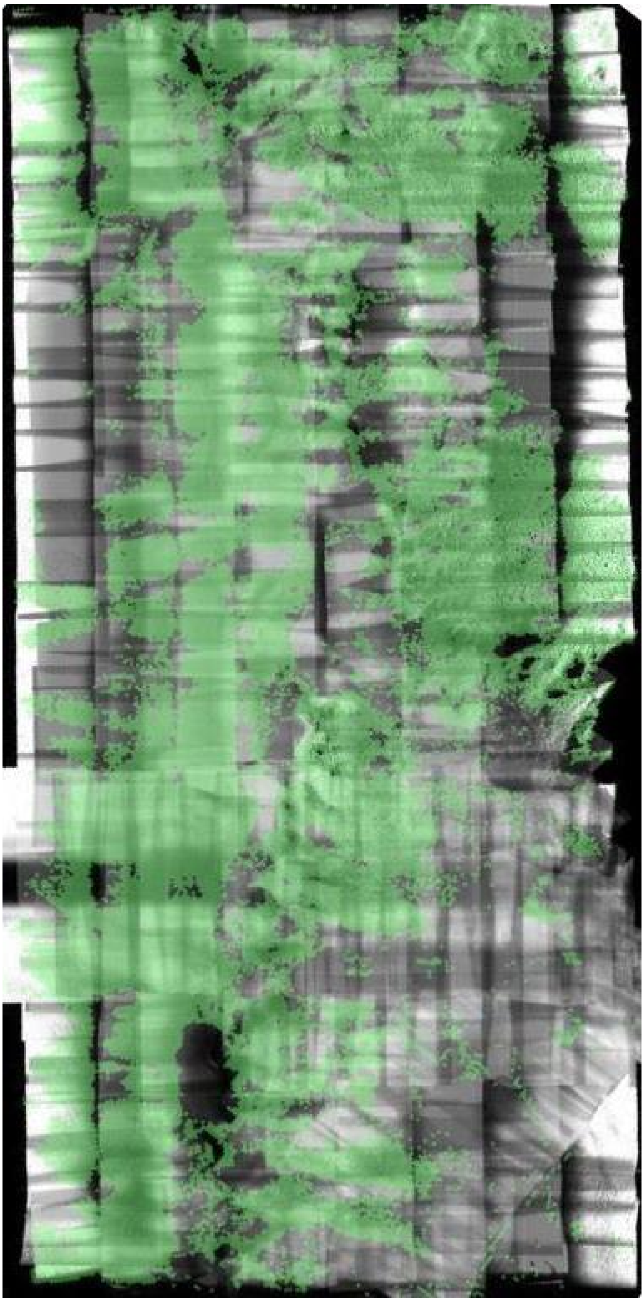


Figure A.15: Mosaic in 33x33 cm resolution of the Tautra ridge in 2017 with the ”average” option for mosaicing in Reflection, layered with the classification result created by the entire 4x4 cm resolution data set.

A.5.2 Nord-Leksa

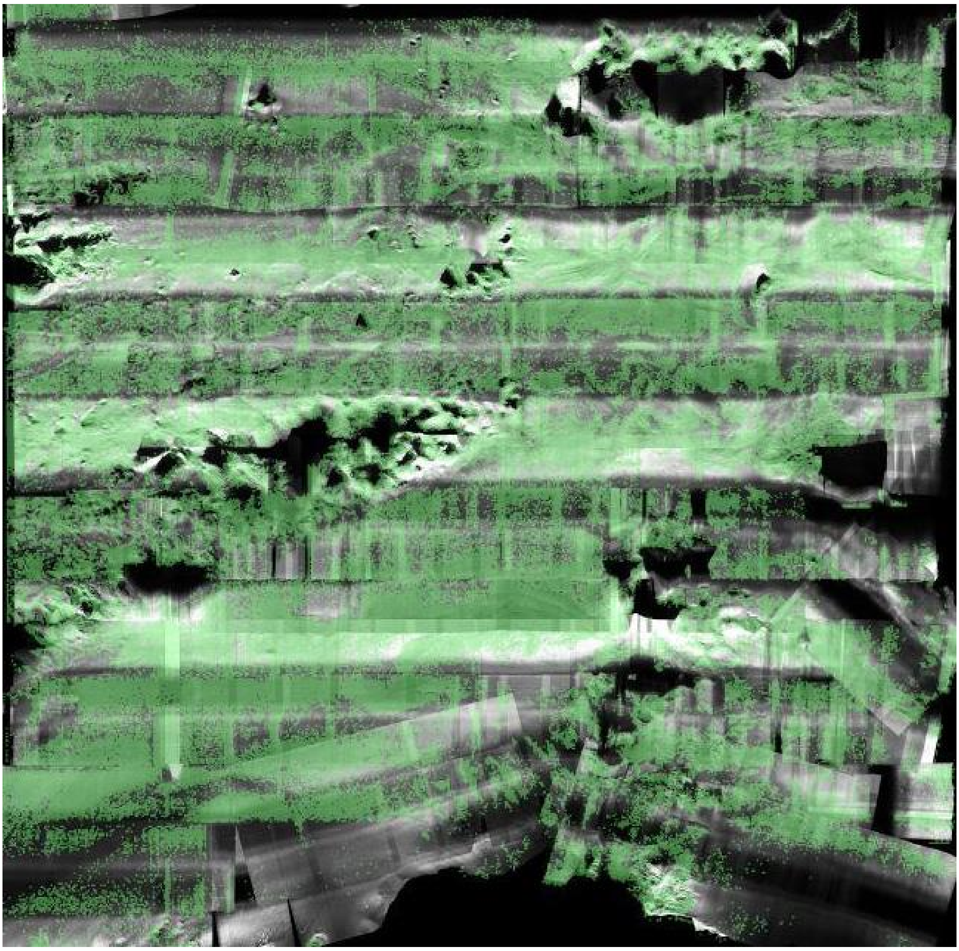


Figure A.16: Mosaic in 33x33 cm resolution of Nord-Lekssa with the ”average” option for mosaicing in Reflection, layered with the classification result created by the entire 4x4 cm resolution data set.