
Classifying coral shore regions, a deep learning approach

NIOZ / UTRECHT UNIVERSITY

MSc MARINE SCIENCES

Author
Nathalie DEES

Supervisors
Dr. Paolo STOCCHI
Dr. Adam CANDY
Dr. Francesca SANGIORGI

Abstract

Coral reef environments are important marine ecosystem and they are in decline due to climate change. Monitoring coral reefs is crucial but has its challenges since coral reefs are dynamic and data is sensitive to weather conditions. This often leads to difficulties for acquiring accurate data. Coral reef surveys are normally done during ideal conditions which are not always available when researchers are present. That is only one of several challenging aspects in coral reef classification. This study explores the possibilities of applying deep learning to classify drone imagery from coral reefs as a promising new approach for coral reef classification, but also as a new approach for producing geophysical output. This study also explores if it is possible to classify non ideal shore imagery. The possibility to work with non ideal imagery would provide more options for the analysis of coral reef surveys. The images are classified using a segmentation approach with a convolutional neural network. The binary classifications achieve high accuracies of 94% and 96% and IoU values of 77% and 71%. From the segmented output a coral density map is derived. The production of the coral density map is the one of the final steps towards producing promising geophysical output. The classification of non ideal imagery led to some difficulties, but it should be possible to fully correctly classify those as well, which can be discussed in further research.



Contents

1	Introduction	1
2	Background	3
2.1	Machine learning	3
2.1.1	Segmentation	4
2.2	Coral reef image analysis	5
2.2.1	Image analysis in remote sensing	5
2.2.2	Image analysis computer vision	6
2.3	Neural Networks	7
2.3.1	Artificial Neural Networks	7
2.3.2	Convolutional Neural Networks	9
2.3.3	Fully convolutional neural network	9
2.4	Model basics	10
2.4.1	Input data selection	11
2.4.2	Learning rate	11
2.4.3	Over- and underfitting	12
2.4.4	Loss function and regularization	12
2.5	Bathymetry	13
2.6	Previous work	14
3	Aim	15
4	Methods	16
4.1	Data	17
4.1.1	Moorea	17
4.1.2	Curaçao	17
4.2	Annotation	18
4.3	Data augmentation	19
4.4	Binary classification	20
4.4.1	Moorea	20
4.4.2	Curaçao	20
4.5	Bathymetry	21
5	Evaluation	22
5.1	Metrics	22
5.1.1	Training and validation error	22
5.1.2	Confusion Matrix	22
5.1.3	Accuracy	23
5.1.4	Intersection over Union	23
5.1.5	Assembled metrics	24
6	Results	25
6.1	Binary	25
6.1.1	Moorea	25
6.1.2	Curaçao	27
6.2	Coral Density	29

7 Discussion	31
7.1 Discussion on results	31
7.1.1 Moorea	31
7.1.2 Curaçao	31
7.1.3 Comparison	32
7.2 Problems which arose during classification	33
7.2.1 Edge artifacts	33
7.2.2 Augmentation effects	33
7.2.3 Validation set	33
7.2.4 Image enhancing	35
7.2.5 Fast.ai	35
7.3 Bathymetric map	35
7.4 Importance of this research	36
8 Conclusion	37
9 Acknowledgements	38
Appendices	43
A Extra background	43
A.1 Deep learning	43
A.1.1 Neuron	43
A.1.2 ResNet	43
A.2 Evaluation	44
A.2.1 Precision	44
A.2.2 Recall	44
B Methods	44
B.1 Binary Curaçao	44
B.2 Multi class classification	45
B.2.1 Five classes	45
B.2.2 Eight classes	46
C Results	46
C.1 Binary Curaçao	46
C.2 Multi class classification	47
C.2.1 Five classes	47
C.2.2 Eight classes	48

1 Introduction

It is becoming increasingly acknowledged that coral reefs are globally threatened (Gardner et al. 2003; Heron et al. 2016). Coral reefs are ecosystems with copious amounts of productivity because they are efficient at trapping nutrients, zooplankton, and possibly phytoplankton from the surrounding waters (Yahel et al. 1998; Monismith 2007). Our knowledge about the human impacts on coral reefs worldwide is patchy and composed largely of short-lived data collection efforts (Pendleton and Edwards 2017). Quantification of benthic community structure is central to understanding coral reef ecosystem function (Hochberg et al. 2003) and coral reef monitoring will play a crucial part in improving that understanding.

Thousands of images of coral reefs are being captured regularly by several types of photography (González-Rivero et al. 2016). Remote sensing has provided the opportunity to produce large data sets with fine spatial and temporal resolutions while decreasing physical or biological disturbance (Nagendra 2001; Kerr and Ostrovsky 2003; K. Wang et al. 2010; Gomes et al. 2018). There is a need to quantify reef community structure through time and over large areas, and remote sensing seems to be the most promising tool for the task (Hochberg et al. 2003). There is however, still a considerable disconnection between pixel resolution and the spatial scale of ecological-relevant features and processes (Philipson and Lindell 2003; Zharikov et al. 2005) associated with remote sensing techniques (Gomes et al. 2018).

Remote sensing includes any method of detection or measurement made without directly contacting the object of study (Shihavuddin et al. 2013). Firstly, there is large scale (satellite) imagery. It has a coarse resolution and is often subject to environmental conditions, which hinder a detailed spatial analysis (Bennett and Younes 2020). Secondly, there is high-resolution satellite imagery from commercial satellites. It has pixel sizes under 10 m and enables higher accuracy when classifying coral reefs over finer spatial scales (Saul and Purkis 2015). However, these commercially sourced images can be expensive to acquire (Bennett and Younes 2020). Thirdly, there are options using unmanned aerial vehicles (UAVs). Recent advances in remote sensing UAV provide advantages over satellite imagery, including their low costs (Parsons et al. 2018; Bennett and Younes 2020). Therefore, UAVs can act as relatively quick monitoring tools for shore detection, coral detection, surveillance (Parsons et al. 2018) and offers potential for regular routine use. Finally, there are 'traditional' reef surveys for mapping underwater taxa, which are performed in situ by scuba divers trained in marine ecology (King et al. 2018). While accurate, in situ surveys are time consuming, expensive and allow only limited coverage of the coral reef area. Large scale (satellite imagery) and in-situ reef surveys are the most commonly used forms for reef analysis. Because of its costs, the efficiency of high-resolution satellite imagery has not been explored a lot so far. UAV imagery provides an exiting opportunity to fill the gap between in situ reef surveys and large scale imagery.

In parallel, analysing the growing amount of remote sensing data is a bottleneck, which normally is manually done by experts. Until recently, remote sensing for shore detection has been combined with standard machine learning practises (Maxwell et al. 2018). The classification of underwater coral reef photography has been tackled by using deep learning in the field of computer science, such as by Beijbom et al. (2012), I. Alonso et al. (2019), and Gómez-Ríos et al. (2019). Deep learning is a branch of machine learning, which uses neural networks as standard algorithm. Machine learning refers to models that are able to automatically detect patterns in data (Raphael et al. 2020). Based on learned patterns, machine learning is able to recognize patterns in new data. In remote sensing however, deep learning approaches are sparsely explored. Remote sensing and computer science research both focus on different aspects of coral reef classification. Most remote sensing literature classifies shore regions (Casella et al. 2017; Ventura et al. 2018), while computer science literature classify underwater photography (Beijbom et al. 2012; I. Alonso et al. 2019; Gómez-Ríos et al. 2019).

In computer science, underwater coral classification is often done while comparing several deep learning tools. In older research, only the usage of standard machine learning practices is explored (Beijbom et al. 2012) or a comparison between standard machine learning and more advanced deep learning approaches is made (Shihavuddin et al. 2013). However, more recent research all focus on neural networks (González-Rivero et al. 2020)). Most computer science papers strive to reach the highest accuracy, or build the most efficient model, such as (I. Alonso et al. 2019; F. Liu and Fang 2020). Whilst most remote sensing papers strive for realistic, usable results (Collin et al. 2018). The two distinct research fields and their approaches are often not combined. However, exploring possibilities to combine the two fields might lead to a beneficial outcome. Analysing coral reef data from different points of views can help in the analysis of large amounts of reef data, which thereby can improve the understanding of coral reefs (González-Rivero et al. 2016).

Both research fields provide results for coral reef imagery in the form of classification, which means certain features in an image are classified. However, deep learning could be applied to provide a geophysical output, which in this case could be bathymetric data. Normally, bathymetric data can be obtained by either active or passive remote sensing (Jawak et al. 2015). Currently, if both active or passive bathymetry do not succeed, there is no alternative to obtaining bathymetric data. Applying deep learning to coral reef imagery, might provide an alternative to obtaining a bathymetric profile. This research will explore the possibilities to apply deep learning to transform imagery to bathymetric data

Data analysis of coral shore regions is normally done under ideal conditions. Shore regions are prone to influence by hydrodynamics which are heavily influenced by weather conditions such as tides, wind, waves and/or clouds. Poor weather conditions can cause for distortions in the imagery. Besides, data acquisition in coastal regions is prone to error, mainly due to two factors. Firstly, light absorption precludes the recognition of features below a critical depth threshold (depending on water turbulence and turbidity). Secondly, optical distortions and reflections at the air–water interface and within the water column introduce errors (Casella et al. 2017). When conditions are non ideal, the probability for acquiring poor data is high. Therefore, data which classifies as non ideal, would normally be unusable. For this study, two datasets have been used. A data set acquired under ideal conditions from Moorea, French Polynesia and one acquired during non ideal settings from Curacao in the Caribbean. The environmental setting per island differs per location. This research strives to answer essential questions in coral reef classification.

2 Background

2.1 Machine learning

Artificial intelligence (AI) is a branch of computer science concerned with building smart machines capable of performing tasks which normally require human intelligence. Machine learning (ML) is a subdivision of artificial intelligence. Machine learning provides computers the ability to learn without being explicitly programmed. ML learns the underlying behavior of a system from a set of input data. ML-based techniques do not need a prior knowledge about the nature of the relationships between the data (Lary et al. 2016). Machine learning is divided into two categories, namely supervised learning and unsupervised learning (Hastie et al. 2009). Supervised learning is when input and output variables are known and the algorithm maps the input to the output variables. Unsupervised learning is when the only known value is the input value, the algorithm needs to learn the output value by itself, without guidance of a human annotator. Supervised learning algorithms can be broken down into regression and classification. A regression model seeks to predict a continuous quantity, for example price or weight. A classification model seeks to predict class labels. The application of ML-based techniques in the geosciences and remote sensing area is fairly new and limited (Lary et al. 2016).

Random Forests (RF), Support Vector Machines (SVM) and Artificial Neural Network (ANN) are the most commonly explored ML techniques in remote sensing literature (Lary et al. 2016). Neural networks will be explained in section 2.3, a brief summary of SVM and RF will be given below. SVM can be used for both regression and classification. It is an algorithm which finds a hyperplane between data points to distribute them in different classes. As there are several hyperplanes possible to distribute the data points. The objective is to find a plane that has the maximum margin, so the maximum distance between data points of both classes as is illustrated in figure 1. The SVM technique improves the generalization ability of machine learning, and proved to be a baseline method for classification.

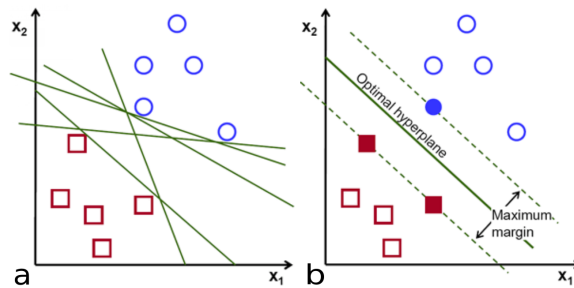


Figure 1: (a) shows the options for possible hyperplanes to group data points, whilst (b) shows the optimal hyperplane to group data points, found by the SVM algorithm, altered from Gandhi (2018).

A random forest model is build up of multiple random decision trees (fig 2). A decision tree is a basic supervised learning algorithm. The structure of a decision tree is like a tree with nodes (fig. 2). The branches depend on the number of criteria. It splits data into several branches, which can be defined by the user, till it achieves a threshold unit. In a RF model, two types of randomnesses are built into the trees. Firstly, each tree is built on a random sample from the original data. Secondly, at each tree node a subset of features are randomly selected to generate the best possibilities for splitting up the data. Finally, outputs of all trees are compared and the most likely classification is picked as final classification. RF is widely utilized in several fields and has advantages of high

efficiency and insensitivity to data outliers (Wan and Ma 2020).

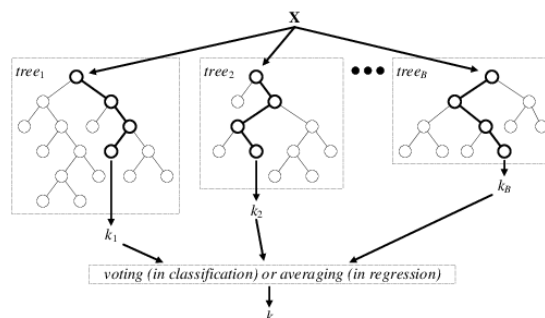


Figure 2: A simplified explanation of the random forest algorithm, the forest is built up of several decision trees. X is the input variable, and k is the output (Verikas et al. 2016).

The usage of machine learning and deep learning in remote sensing is explored in literature, but it is often not used for real life applications (Maxwell et al. 2018). For example, in analysis of 1651 articles that compared remote-sensing classifications methods, Le Yu et al. (2014) found that the maximum likelihood classifier was the most commonly used method, employed in 32 percent of the articles, even though machine-learning methods were routinely found to have notably higher accuracies than maximum likelihood (Maxwell et al. 2018; Wicaksono et al. 2019). The majority of published research focusing on the use of supervised classification of remote sensing data has been for the prediction of land cover or vegetation classes (Cracknell and Reading 2014).

Although basic machine learning practises, such as RF and SVM, have been shown to be effective, the data sets are often small and contain little or no noise and/or outliers. Their accuracy and efficiency are significantly challenged by problems involving big data and complex patterns, which can vary from spatial patterns to differences in shape (W. Li and Hsu 2020). Besides, RF and SVM rely on so-called “shallow structures” without comprehensively learning the complex nonlinear features, causing a dilemma with respect to the improvement of the classification accuracy (Wan and Ma 2020).

2.1.1 Segmentation

In image analysis and computer vision, deep learning applications can be categorized into three classes: classification, object detection, and segmentation (Badea et al. 2016; W. Li and Hsu 2020). Classification identifies what is inside an image. Object detection (fig. 3) goes one step further, as it not only predicts an object class but also identifies the location of an object (W. Li and Hsu 2020). Segmentation can be divided into semantic segmentation and instance segmentation, see figure 3. Instance segmentation differs from semantic segmentation as it gives a unique label to all particular objects in the image. Semantic segmentation has been explored more, as instance segmentation is a new phenomenon. Semantic segmentation assigns a specific label to each pixel in the image. Semantic segmentation represents a step forward in benthic image analysis as it not only allows detection of shapes and possibly specific species, but also explores morphological attributes such as size-frequency distribution of key groups across an image set with minimal labeling efforts (I. Alonso et al. 2019). As we want to label every pixel in the images, semantic segmentation is the most valid approach.

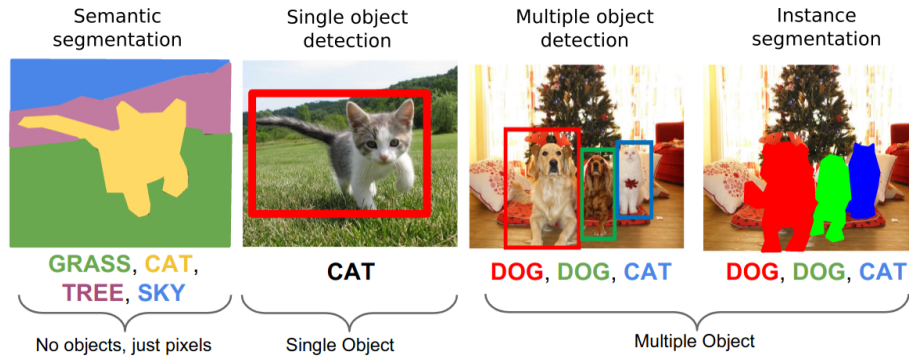


Figure 3: Examples of semantic segmentation, object detection and image segmentation shown respectively. The image is altered from F.-F. Li et al. (2017).

2.2 Coral reef image analysis

There are two distinct research fields in coral reef image analysis. A brief history is given for both research fields, as well some details about what makes coral reef image analysis a hard task. Shore regions in general, but coral reefs especially, are dynamic ecosystems that can vary in community structure over small spatial scales due to the mosaic of habitats created by biotic and abiotic influences (Airoldi 2003; Murfitt et al. 2017). Hydrodynamical and geochemical interactions occurring in corals reefs differ spatially (Monismith 2007). Therefore, depending on what is researched, coral reefs can be mapped in several scales (Phinn et al. 2012) (fig. 4). Challenges in coral reef image analysis are caused by extreme variations in the size, color, shape, and texture of each of the organism classes (i.e., taxa) and the organic and ambiguous nature of the class boundaries (Beijbom et al. 2012; King et al. 2018). Both underwater and shore data acquisition and analysis cause for several difficulties. Manual in-situ mapping performed underwater by human divers is time consuming, whereas aerial photography and satellite remote sensing are both limited by the fact that seawater absorbs light strongly, thereby limiting monitoring to only shallow marine ecosystems (King et al. 2018). Traditionally, overhead images of coral reef sections are manually annotated by domain experts. During the annotation process, experts are presented with randomly generated pixel positions in an image and are required to provide a classification label for each of these pixels. A shortcoming of this process is that it is labor intensive, which limits the scale and sampling frequency of coral ecosystem assessment (King et al. 2018).

2.2.1 Image analysis in remote sensing

Aerial photography is one of the simplest and oldest form of acquiring remote sensing data. It typically covers local to regional scales (from several km to 100s of km), sometimes with records stretching back to the 1930s (Hernández-Cruz et al. 2006; Goodman et al. 2013) p 9. As aerial photography consists of photos, the data consists of three bands, namely the red, green and blue bands. Satellite imagery consist of more bands, picking up a wider spectrum of data input. This is called spectral reflectance. Aerial photography cannot display a spectral-reflectance signature; however their simple format and long term collection worldwide make them a unique resource for coral reef mapping and monitoring over time (Goodman et al. 2013), which can lead to unique understandings about specific coral reefs (Bakker et al. 2017). Previous methods that were used for

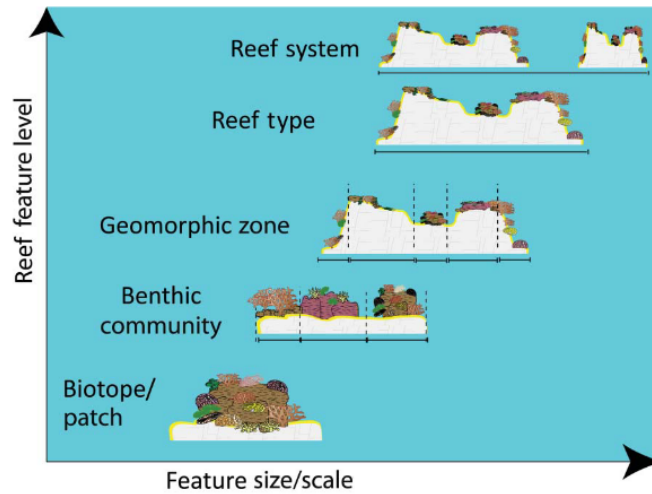


Figure 4: Spatial scale and levels of mapping detail for coral reefs (Phinn et al. 2012).

coral reef classification include both unsupervised classification methods and supervised classification methods such as support vector machine (SVM).

Most recent advances in image processing have seen significant development and application of Object Based Image Analysis (OBIA) approaches (Phinn et al. 2012). OBIA applies a segmentation approach and divides up the image into several segments (groups of pixels), which can have neighbouring or hierarchical relationships (Blaschke et al. 2014) (fig. 5). Object based analysis contains two steps: (1) segmentation of the image into specific segments and (2) the classification of the segments. The grouping process determines the size, shape and number of segments to which image pixels are grouped, which depends on the required spatial mapping detail. The groups of pixels are then assigned a class through membership rules based on colour, texture, shape, size, position and relation to the environmental properties map of the area (fig. 5). This is different from segmentation mentioned previously, as those assigns a pixel value to each pixel in an image, without generating separate objects.

While a commonly used approach in terrestrial environments, publications of the application of OBIA in coral reef environments are increasing in recent years (Hedley et al. 2016). The OBIA approach however, needs widely used commercial software to analyse images (Bennett and Younes 2020), which in some cases are not easily accessible to scientists and managers. The commercial software often requires expensive subscriptions, specialized knowledge, and capable computing hardware to perform analyses on large image collections (A. Alonso et al. 2016; Bennett and Younes 2020). Although, open source alternatives to these commercial software are on the rise (Phinn et al. 2012) and Google Earth Engine is starting to provide free alternatives as well (Bennett and Younes 2020). Phinn et al. (2012) found it challenging to classify several reefs with only one workflow in OBIA. They suggest room for improvement for advanced segmentation options in labelling and classifying, as manual adaptation was needed to use the same method for several reefs.

2.2.2 Image analysis computer vision

The main goal of coral classification in computer vision has been classifying underwater imagery (Wan and Ma 2020). This is because the most popular type of algorithm, CNNs, have proved to be

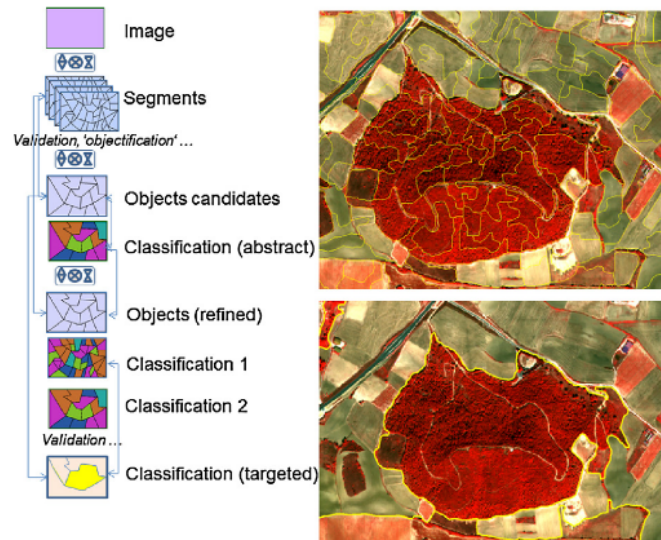


Figure 5: An example of the OBIA workflow (Blaschke et al. 2014).

beneficial in the field of underwater photography (Raphael et al. 2020). However, even with CNNs some challenges with coral reef detection arise. Difficulties are caused by variations in lighting due to the water column, or by the variance between images of the same class, or because that specific coral species tend to appear together (Gómez-Ríos et al. 2019). Another bottleneck of CNNs is that they normally need a large data set to achieve a good performance (Gómez-Ríos et al. 2019). A great collaboration by computer vision scientists was the invention of CoralNet by Beijbom et al. (2012). CoralNet enables users to upload and annotate coral reef images and share knowledge. It is also now starting to deploy deep neural networks which allow fully or semi-automated annotation of images. In an extensive study, Williams et al. (2019) showed that the automated annotations for CoralNet Beta produced benthic cover estimates similar to results derived by manual annotation. The goal of the CoralNet is to provide any user with the ability to take advantage of their automated analysis (Raphael et al. 2020). It's big strength is that several researchers can benefit from the already uploaded images to more easily train their networks on.

2.3 Neural Networks

Since our focus is to use neural networks, rather than improve the theoretical aspects of their design, we will not focus on the mathematical background describing the neural network in great detail. The focus will be on basic understanding of how the algorithms work, so they can easily be applied to earth science problems.

2.3.1 Artificial Neural Networks

The most simple framework for a neural network is an Artificial Neural Network (ANN), which can be seen in figure 6. Artificial neural networks take inspiration from models of the biological brain, and try to reproduce some of its functions by using simple but massively interconnected processing units, the neurons. See the appendix for more information about neurons.

A typical neural network architecture comprises several layers of neurons feeding one another, which is what is referred to as the "deep" attribute in deep learning (Castelluccio et al. 2015; Raphael et al. 2020). The ANN in figure 6a for example, only has one layer. An ANN has minimum input and output layers, with a neuron for each input variable, and a neuron for each output class. The key characteristic of an ANN is that all neurons in one layer are connected to all neurons in all neighbouring layers, and the connections with these layers have weights. The weights on the connections determine how input values are mapped to values on the output nodes. These weights are in combination an activation function that further modifies values at every neuron (Maxwell et al. 2018). An activation function decides whether a neuron should be activated or not by calculating weighted sum and further adds bias to the neuron. Normally, the activation function is non-linear. Increasing the number of neurons in the hidden layer, and adding more hidden layers, rapidly increases the potential for describing very complex decision boundaries. As they train on non-linear functions, they specialize in picking up non-linear boundaries. Neural networks are typically trained by randomly guessing values for the weights, and then iteratively adjusting those weights and observing the effect on the output nodes. Adjustments that improve the classification are kept and reinforced; adjustments that do not are discarded (Maxwell et al. 2018). The process of adjusting the weights during the training stage of the network is called backpropagation. This whole process happens automatically once the network is build, there is no need for manual adaptation. The model trains itself to find the most optimal solution. Neural networks can be built up from scratch, meaning the architecture mentioned needs to be manually built. However, several types of architectures are already pre-built, which do not need manual adaptation, which leaves to user with only a few manual steps.

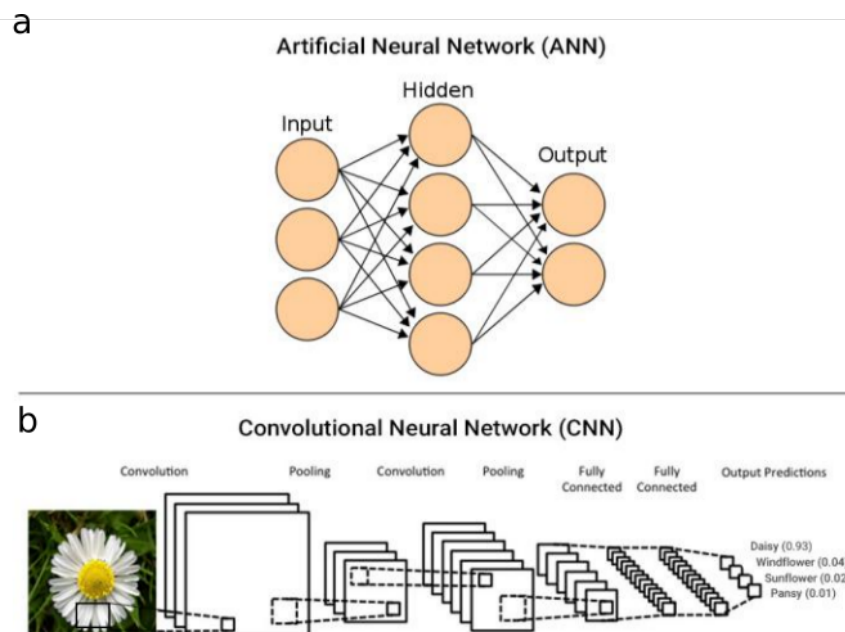


Figure 6: An example of the frameworks for (a) an Artificial Neural Network (ANN) and (b) a Convolutional Neural Network (CNN), altered from Gogul and Kumar (2017).

2.3.2 Convolutional Neural Networks

The advantage of using convolutions to ANNs is that ANNs are only able to work with 1D data, whilst convolutions can work with 2D data. For image analysis, it means that ANNs loses spatial features, whilst with convolutions, the spatial features stay intact. Convolutional Neural Networks (CNNs) automatically extract high-level semantics, such as object parts and borders, from low- to mid-level features, such as color, shape, or texture, in the learning process (W. Li and Hsu 2020)

After passing through the CNN, an image becomes abstracted into features or factors that are organised in a hierarchical way, where higher level factors (an object or a landscape) are learned from lower level or more basic layers (e.g., circles, square edges). Based on this concept, the CNN architecture uses a cascade of many layers to extract and transform features from an image. Each successive layer uses the output from the previous layer as input that construct more complex features that start resembling objects in the higher-level layers (González-Rivero et al. 2020). In this way the CNN is organised by layers forming a hierarchy from low-level to high-level features (González-Rivero et al. 2020; Raphael et al. 2020).

A neural network becomes a convolutional neural network when the hidden layers consist of convolutional layers, which are normally followed by the non-linear activation function and consisting of one or more fully connected layers. Layers where all the inputs from one layer are connected to every activation unit of the next layer are called fully connected layers. For example, the neural network in figure 6a is a fully connected network, even though it only consists of one layer. A convolution merges two sets of information, it refers to the mathematical combination of two functions to produce a third function. The convolution is performed on the input data with the use of filter to produce feature maps. This filter takes nearby pixels together which is important to extract positional relationship in the image data. The convolution is executed by moving the filter over the input data. The shape of the filter differs per input shape of the image and architecture of the network. At every location, matrix multiplication is performed and sums the result onto the feature map. The results collected by the multiplication are then summed and the local output is generated. Then this output pass-through activation function like Rectified Linear Unit (RELU), sigmoid function or tanh, RELU is the most commonly used one (Dertat 2017). A RELU function returns 0 if it receives any negative input, but for any positive value x , it returns that value back. Meaning it is linear for positive values and 0 for negative values. This process is still assessing weights to connections, as described above. Which means the data then moves on to the next layer, doing this process all over again if it is a convolutional layer.

2.3.3 Fully convolutional neural network

The summary given in the subsection neural networks brings us at the type of network which is used for this analysis, which is a Fully Convolutional Neural Network (FCNN). For image classification with CNNs, the model normally generates a single label classification for the whole image. However, in several visual tasks, the desired output should be a specific class label for each separate pixel in an image (Ronneberger et al. 2015). FCNNs are capable of doing object classification and semantic segmentation by generating a class prediction for each pixel in each image (King et al. 2018). This is because a FCNN transforms the height and width of the intermediate layer feature map back to the size of input image through a specific convolution layer (Zhang et al. 2020). The specific convolutional layer is carried out for upsampling, thus to generate an output feature map that has a spatial dimension greater than that of the input feature map. The fully connected part of the CNN network goes through its own backpropagation process to determine the most accurate weights. Each neuron receives weights that prioritize the most appropriate label. Finally, the neurons “vote” on each of the labels, and the winner of that vote is the classification decision.

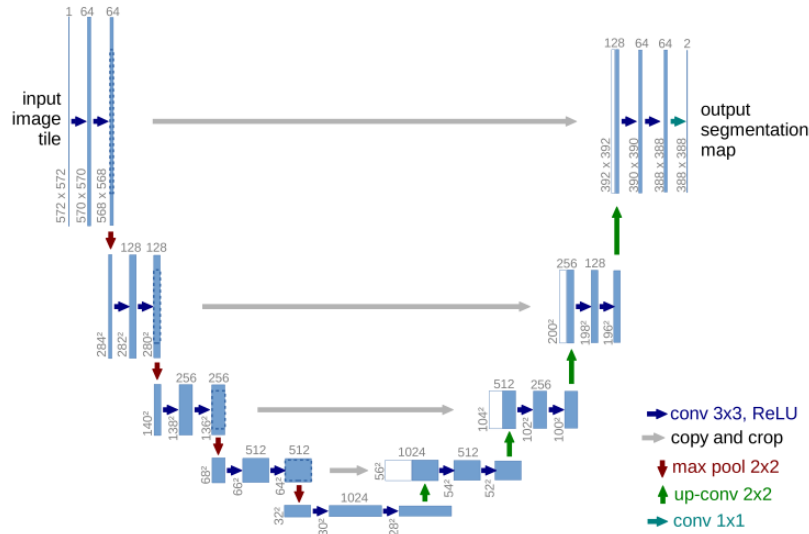


Figure 7: The Unet architecture (Ronneberger et al. 2015)

For this research a class label for each pixel is desired, therefore a FCNN is the only valid approach. One image can consist of several features which need to be localized, this is why class labels for the whole image would not suffice. There are several architectures to choose from, the most popular are Unet (Ronneberger et al. 2015) (fig. 7) and Deeplab V3+ (Chen et al. 2018). Unet was developed as a biomedical image segmentation algorithm, but due to its well build architecture, it is now used in a variety of different research fields. Deeplab V3+ is developed by Google and it is used in their facial detection algorithms and autonomous driving algorithm. For this research, we have chosen the Unet architecture. This is because it is already implemented in the fast.ai Library (Howard and Gugger 2020) build in Python (Van Rossum and Drake 2009) used for this analysis. Fastai is a modern deep learning library and is organized around two main design goals: to be approachable and rapidly productive, while also being deeply hackable and configurable (Howard and Gugger 2020). The built-in Unet function makes it a quick and easy process to work with, as there is no need to manually build a deep learning algorithm. However, manually building the preferred architecture is also possible in fast.ai.

2.4 Model basics

The data is divided into a training set, a validation set and a test set (fig. 8). The training set is what the model uses to train itself on. It contains manually annotated data. The validation set is used to evaluate the performance of the model. The validation set compares the human input with the output of the model. Ideally, it contains new but similar data input compared the training and validation set, to provide non-biased evaluation metrics. When satisfied with the model performance, the model can be tested using the test set. The test set contains new, similar data and ideally, it does not contain data samples which are also used in the validation and training set. After the optimal model has been trained, the model is used to generate new outputs for the test set. Normally, circa 20 or 30 percent of the input data is chosen as validation set, the other 80 or 70 percent is used as training set.

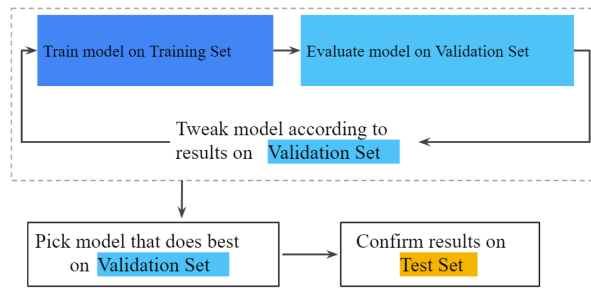


Figure 8: Basics of a machine learning model (link)

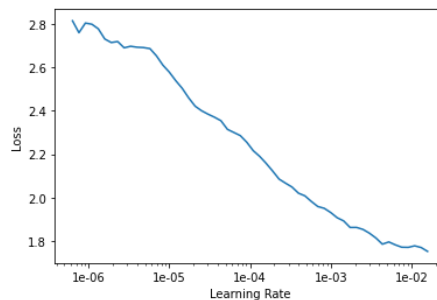
For a segmentation problem, data input consists of two parts, namely the raw image and an annotated mask. A mask is a separate image file, which is linked with the raw image during training. When a finalized model is applied to the test set, new masks are generated.

2.4.1 Input data selection

The training data must be representative for the data available in the test set. When the data in the test set varies, the network may fail to classify the test data accurately (Mas and Flores 2008). Therefore, with a small data set, it is crucial which images are a part of the training set and which are not. Another importance is the amount of classes used as labels. The proportion of the amount of categories per amount of data is crucial (Mas and Flores 2008). A network trained on a data set with unevenly distributed classes will bias its decision towards the more represented classes, as this allows the algorithm to lower the overall error, which is much more heavily influenced by the more common cases (Mas and Flores 2008).

2.4.2 Learning rate

The learning rate parameter determines how much an updating step influences the current value of the weights assigned to the neurons. Generally speaking, the learning rate decides how many epochs are needed optimally train the model. One **epoch** is one run using all the data. If learning rate is small, the training process will run slowly and it will take a lot of time to train. If learning rate is high, the training algorithm will learn too quick, meaning it takes large, illogical steps, to get to the end result.

Figure 9: Example of a learning rate finder graph, here the ideal learning rate would also be around 1×10^{-4}

To find the ideal learning rate in fast.ai, a graph like figure 9 is plotted. Visually, the best learning rate is chosen. Manually, a high and low learning rate are also tried, to see which algorithm provides the best results. Trying and testing several rates, has been proved to be the most consistent method (Smith 2017; Howard and Gugger 2020). Normally, the ideal learning rate is found where the gradient is the steepest, which can be detected automatically or visually. Howard and Gugger (2020) argues that most of the times the ideal learning rate is around 1×10^{-4} , but it depends on the graph shown.

2.4.3 Over- and underfitting

Ideally, a model is not underfitting but also not overfitting. The phenomenon of fitting our training data more closely than we fit the underlying distribution is called overfitting (Zhang et al. 2020; Mas and Flores 2008). Underfitting is when the model does not match the underlying distribution, this can be due to a lack of data. Examples of under and overfitting can be seen in figure 10. A model which can fit to data appropriately, can be applied to new data sets. However, a model which is overfitting, cannot be applied to a new data set, since it is only able to perform on that specific data set.

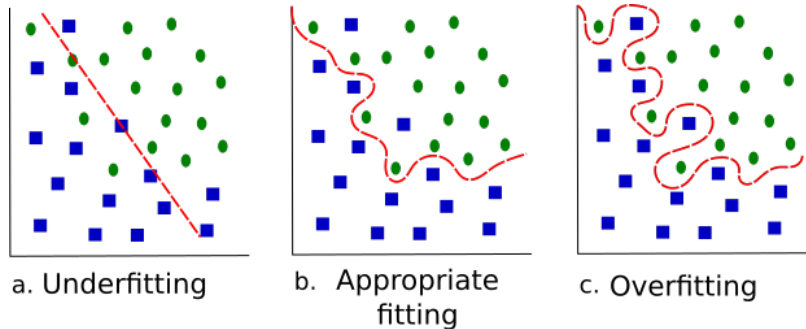


Figure 10: Graphical representation of a) underfitting, the model is too simple to example the variance, b) appropriate fitting, the model is missing some outliers but can also be applied to new data and c) overfitting, the model can only be applied to this specific data set, it cannot generalize.

2.4.4 Loss function and regularization

In mathematical optimisation, statistics, econometrics, decision theory, machine learning and computational neuroscience, a loss function or cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some “cost” associated with the event (Wu et al. 2019). In our case, the loss function is the left part of figure 11. For a sweet spot between over- and underfitting, the model needs to be simplistic, yet complex. A way to both generalise but also increase complexity is to add something to that loss function. This is called a regularization technique. The most popular regularization technique is L2-regularization, which is described here. This term adds something to the loss function, namely the squares of all parameters, or weights, see fig 11. This can result in high loss values, which could lead the model to set all parameters to 0. To prevent that from happening, the sum of squares is multiplied with another smaller number. This number is called weight decay, which is the right part of figure 11. **Weight decay** is the regularization method which will be used in this study. This method tries to estimate the mean of the data to find the sweet spot.

The completed loss function shows how the weights or parameters in the neural network need to be changed in order to find the desired output. This process happens during every epoch, in every neuron connection. Weights are adjusted accordingly when the model is trained during each epoch, until the model is able to produce the expected output. This is the way the neural network learns, which in terminology is called backpropagation. In summary, backpropagation is an algorithm which performs an efficient search for the most optimal weight values. It allows to bring the error functions to a minimum with low computational resources, even in large, realistic models.

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij} W_j)^2}_{\text{Loss function}} + \lambda \underbrace{\sum_{j=0}^M W_j^2}_{\text{Regularization Term}}$$

2

Figure 11: The mathematical equation for a standard loss function (left part) and the regularization term (right part). Figure is taken from (link).

2.5 Bathymetry

A depth map is important for coral reef environments, as it can provide understanding in types of habitats and it can function as a basis for hydrodynamical modelling experiments. Most common bathymetric scales would provide information of the reef system, reef type and at the smallest scale, geomorphic zone (fig. 4). Bathymetric data can be obtained in several ways, for example, it can be estimated from multispectral or hyperspectral imagery. In all cases, the highest quality bathymetric data is obtainable under the clearest water conditions and with minimal surface sun-glint (Hedley et al. 2016). Casella et al. (2017) proved it is also possible to generate coral reef bathymetric data with the structure from motion (SfM) approach, using only aerial photography. SfM solves camera pose and scene geometry simultaneously and automatically using several photos of the same features, but from different positions, see Westoby et al. (2012) for more information. When using the SfM approach, the maximum reliable depth estimates can be 20 m, but 10 m is a more realistic aim. In dynamic coastal areas, maximum reliable depth may be a metre or less: where the bottom cannot be seen the depth is unknown (Hedley et al. 2016). Digital Elevation Models (DEMs) produced by UAV imagery have previously been used to derive terrain roughness and measure coral reef bathymetry (Leon et al. 2015; Casella et al. 2017; Collin et al. 2018).

In LIDAR bathymetry, a laser transmitter/receiver mounted on an object transmits a pulse that travels to the air–water interface, and a portion of that energy reflects back to the receiver. The remaining energy propagates through the water column and reflects off the sea bottom (Klemas 2011), which is how the bathymetric map is made. In general, the accuracy of bathymetric maps generated from multispectral or hyperspectral imagery is considered low compared to the use of active remote sensing approaches such as sonar or LIDAR, on the contrary, they are very cost effective over large areas and adequate for some applications (Hedley et al. 2016). For cases where bathymetric data is not available from either LiDAR or the SfM methods, there is no alternative to obtain

bathymetric data. However, with the upcoming evolution in machine learning new possibilities should be explored, which is what will be done in this study.

2.6 Previous work

The amount of research which is exploring classification methods of the marine realm with drone imagery is limited. A short overview of literature which were exploring similar ideas is presented. However, none of them pursue the classification of their drone imagery with a deep learning segmentation approach, which states the importance for exploring this method.

Chabot et al. (2018) are recognizing aquatic invasive species in lakes while using drone imagery and the random forest algorithm. They find that varying spectral and textural characteristics at different depths provide challenges for machine learning classifiers. They find difficulties in distinction between over-segmentation of shallow, bright features and under-segmentation in deeper and darker features. They find high error rates for submerged classes, with the highest error in a vegetation class of 52 percent. However, they find high accuracies for above water features, between 89 and 92 percent. Ventura et al. (2018) are classifying sensitive marine habitats using drone imagery. They find overall accuracies of 84 percent, and class accuracies above 70 percent. They do not mention specific accuracy per class, but results are relatively high, which is due to decreasing distinctness of submerged vegetation with increasing depth. Wicaksono et al. (2019) are using high resolution satellite imagery combined with different machine learning classifiers (SVM and RF) to classify benthic habitats. They state that a suitable classification scheme can be as important as finding the right classifier. They find that benthic habitat maps with only four classes generally provide higher accuracies compared to other maps. The highest overall accuracy they find is 88 percent, while using the random forest classifier.

A pioneer in making bathymetric data from drone imagery in the marine realm is Collin et al. (2018), as they make a bathymetric map and a coral classification from the whole island of Moorea. However, they do have the advantage of possessing over LiDAR data. They make use of traditional drone imagery, LiDAR surface and LiDAR intensity, which they combine in an ANN to produce the bathymetric map and a classification with 5 classes. They make use of a relatively small validation set, 135 pixels compared to the test set of over a million pixels. They find class accuracies between 68 and 85 percent, which is high for classifying a whole island in 5 distinctive classes.

3 Aim

There is a considerable knowledge gap on several aspects. Firstly, there is a data gap on spatial scale, meaning that the usage of UAV imagery in shallow shore coral reef environments is sparsely explored. Secondly, the knowledge from computer science and geoscience is rarely combined. Both research fields mostly stay in their own fields. Finally, using deep learning segmentation methods for the production of a bathymetric map from only UAV imagery is a new approach. This research tackles these problems and strives to answer essential questions in coral reef classification:

- Can deep learning segmentation be used to classify UAV shallow shore coral reef imagery?
- Is it possible to classify imagery taken during non ideal conditions?
- Can deep learning segmentation be used to produce geophysical output only from drone imagery in the form of a bathymetric map?

This study will explore the different possibilities of coral reef classification under both ideal and non ideal conditions, using deep learning. This study will be, to my knowledge, one of the first to combine the deep learning approach to the classification of coral reefs from drone imagery. The aim of the study is to be a bridge between computer science and remote sensing, to explore and possibly fill the knowledge gap into classifying UAV imagery and to find a method that enables a geophysical output from imagery.

4 Methods

The flowchart shown in figure 12 provides a graphic overview of the followed method described in this section.

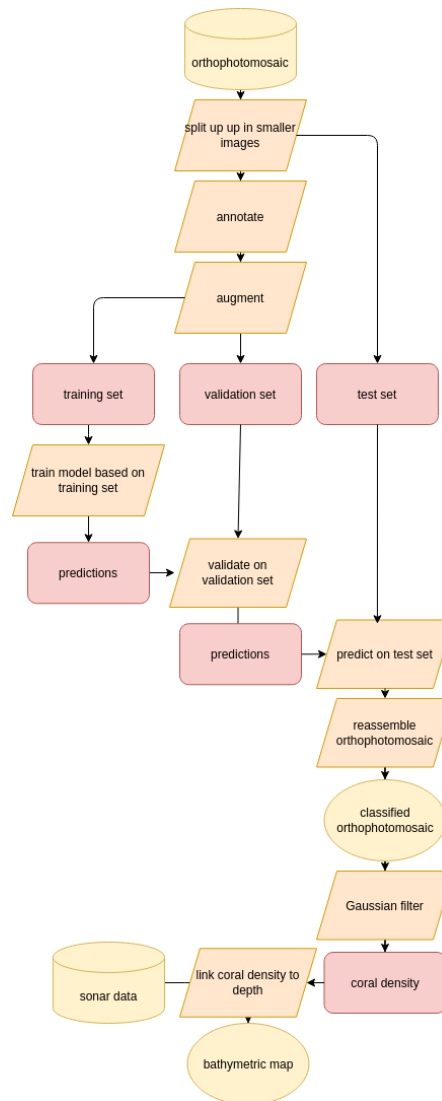


Figure 12: A flowchart describing the segmentation process. The red boxes show steps which caused for difficulties during the process.

4.1 Data

Both study sites were surveyed by drones. For both study sites there has been made an orthophotomosaic by the SfM approach. This means that data is available in two states, the separate photographs taken to make the orthophotomosaic, or the finished orthophotomosaic. The separate photographs contain a lot of overlap, as the SfM approach requires several pictures taken from the same spot, only at a different angle. Therefore, there was chosen to work with the finished mosaics. Those will be the easiest to re-assemble and will quickly provide an overview of the whole bays, which will be needed to produce geophysical output.

4.1.1 Moorea

Moorea data was captured during a drone-based spectral survey in 2015. The study site is located in the northern lagoon of Moorea Island ($17^{\circ}33S$, $149^{\circ}50W$) in the Society Archipelago (French Polynesia (Collin et al. 2018)). Moorea had been researched before by (Casella et al. 2017; Collin et al. 2018). For more information about the drone procedure and the specifics about the orthophotomosaic, see (Casella et al. 2017) for more details. Collin et al. (2018) generated a depth map with an ANN from LiDAR data. According to Collin et al. (2018) Moorea is bathed in oligotrophic seawater, thus clear including various taxa of reef-building corals, red calcareous algae, fleshy algae and a diversity of geomorphic features. The image which will be classified is shown in figure 13. This image was taken under ideal conditions, there are almost no distortions seen in the image. There is also little to no sun glint or wave influence. The Moorea orthophotomosaic was split up into tiles of 256 by 256 pixels which can be seen in figure 13.



Figure 13: A part of the Moorea orthophotomosaic, which will be used in this study. The red lines indicate the separate tiles. Assembled by (Casella et al. 2017).

4.1.2 Curaçao

Curaçao is located in the Dutch Caribbean region. The image was taken in Piscadera bay. As clearly can be seen in figure 14a, the image was taken during non ideal conditions. Especially in the top left of the image, but overall on the shoreline, there are some white water waves present. What also clearly can be seen, is that the right side of the image darker compared to the rest. In figure 14c can be seen that there are signs of sun glint combined with waves. The colour of the water is not



Figure 14: (a) A photomosaic from aerial drone images of the Curaçao shore environment. Close ups of (b) on land and transition area and (c) underwater environment.

as clear compared to Moorea, when moving into a deeper part of the bay, the water turns into a more distinct blue colour, while corals and other features become less detectable. As can be seen in figure 14b, clear signs of corals and algae can be distinguished. Some clear features in the image are branching corals, brain corals, algae, rubble and sand. The orthophotomosaic was split up into images of 512x512 pixels, resulting into 1103 images. The deeper part of the orthophotomosaic, after tile 700 (fig 15), was not taken into account, as this part does not consist of clear features but mostly of blue water.

4.2 Annotation

During the annotation process, a manual mask containing different classes is made. Image masks are separate images, which also can be seen as a separate layer, in which each pixel is labelled as a specific class (fig. 16). For the annotation process, some of the images were selected manually because of their clear features, while others were selected randomly, to keep the input as varied as possible.

There were no reference points provided, so the classes were classified based on human detection. Therefore, they were classified based on prior knowledge and examples. Corals were mainly chosen by their distinct features. In Moorea, corals can clearly be distinguished from sand, due to the shape, texture and colour. In Curaçao it is slightly more difficult, but they are circular and have a distinct orange-brown colour (fig. 16). (Castelluccio et al. 2015) states that labelling for semantic segmentation always provides difficulties, as to where to define the borders between separate classes. Especially in the marine realm, where organisms have ill-defined edges and patchiness to add to the complexity of classification (González-Rivero et al. 2020). Especially in the Curaçao data set, these points were causes for concern.

The data annotation of the masks was done by making use of the Pixel Annotator tool (Bréhéret 2017). There are several programs for annotating masks, this one is a community initiative, thus free and is best suited for making these type of masks. This is a supervised classification tool packed in a ready-to-use GUI which makes use of the watershed algorithm to generate a mask easily and

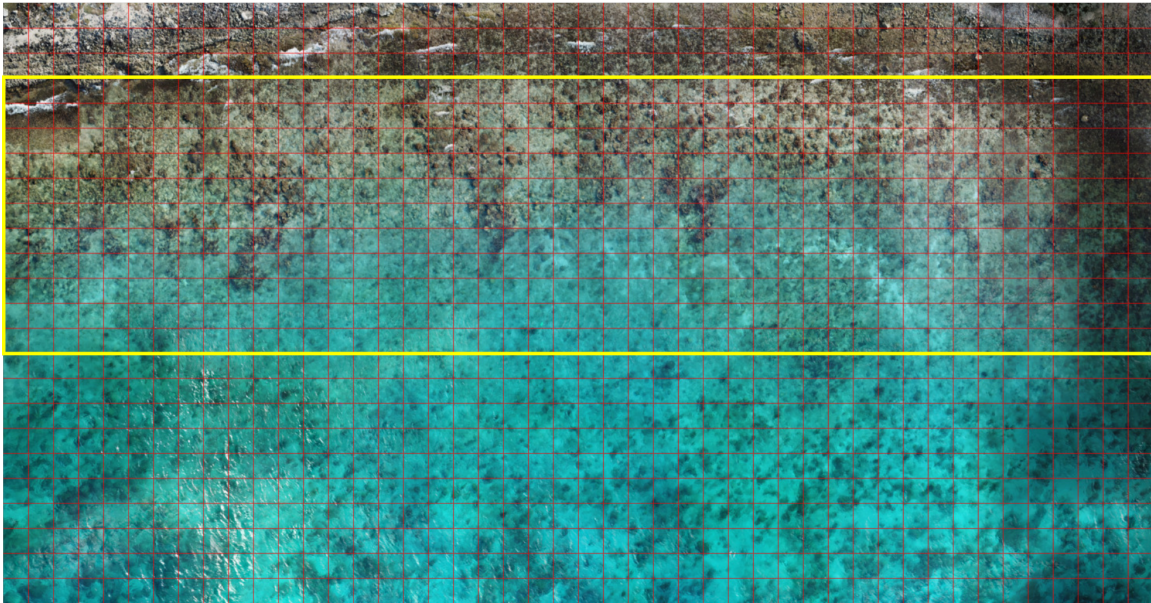


Figure 15: The red lines indicate how the orthophotomosaic was divided into separate tiles. The area in the yellow box indicates which tiles were used for the analysis. The area outside the yellow box was either on land or too submerged.

quickly. The user can select known areas to classify, then pixel annotator generates a mask to classify the whole image. The user can adapt the generated mask manually, to make it fit to their own preferences.

4.3 Data augmentation

The purpose of data augmentation is to generate new sample instances, and when the training samples are few, data augmentation is very useful for improving the robustness of the network (Wu et al. 2019). Normally, neural networks are trained on at least 1000 input images, however some problems classify over 20 thousands of images. In our case, our classified data set contains only around 400 tiles. However, to make the network train accurately, it needs enough images so it is not underfitting. The network needs images which contain different types of environments, otherwise it will overfit on a specific environment. Therefore, the mix of manually selected and randomly selected makes a good starting point. As only 20-40 tiles were manually classified, more tiles were needed to provide a stable framework for the network. This can be done by automatically altering the images, with just a little change.

To do this automatic alternation, I made use of the Augmentor Library (Bloice et al. 2019). Specific parameters can be set, such as small rotations and flips of 90 degrees. This process creates several images from the same image with the mask attached. This generates a diverse and higher amount of input files.

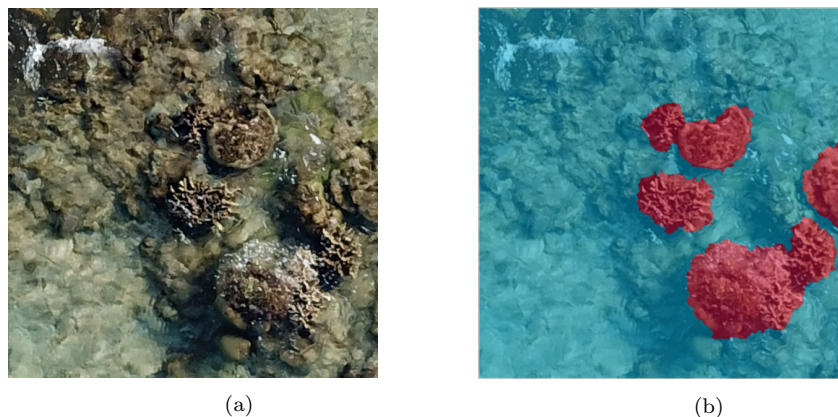


Figure 16: An example of a tile (a) and the tile with manual annotation (b), red represents coral and blue represents non coral

4.4 Binary classification

This research makes use of a convolutional neural network architecture pre-trained or initialised on a large dataset comprised by ten million images and one thousand classes, ImageNet (Deng et al. 2009). This means that our architecture has already been trained on images which are present in the ImageNet database. The database does not contain a lot of coral shore imagery, but it provides a good basis for the recognition of surfaces and shapes. This will mean that the algorithm needs less time to train and it will generalise more quickly.

4.4.1 Moorea

The Moorea data set was chosen as a starting point for binary classification. As Moorea has distinct features to binary classify - coral and non coral. In the Moorea case, the non-coral class is most likely to be sand. For the binary classification, 26 tiles of (256x256 pixels) were manually. These tiles are smaller compared to the Curaçao tiles, as the orthophotomosaic has a higher resolution. It was possible to label smaller tiles, as the features were more distinct. After labelling, the tiles are augmented into 135 tiles in total. 98 tiles are used as training set, while 32 are used as validation set. A script containing a CNN in fast.ai was written. The algorithm was trained with a weight decay of 0.1 and a learning rate of 5×10^{-4} for 16 epochs.

4.4.2 Curaçao

For the Curaçao case, it is harder to distinguish clear differences between coral and non coral. Brain corals and branching corals clearly stand out during the labelling process, whilst rocks, or maybe rocks/corals covered in algae, make the annotation process harder. Several model options were explored, in which some are explained in the appendix. Eventually, we decided was best to split the Curaçao data set into two data sets, a shallow set and a deeper one. The classification of the complete orthophotomosaic did not succeed, on which is elaborated more in the appendix. The two sets were trained separately. For the shallow bay, 30 images were manually annotated. They were augmented into 170 tiles in total, with 135 tiles as training data and 45 tiles as validation data. The learner was trained with a weight decay of 0.01 and a learning rate of 1×10^{-4} for 19 epochs. For the deeper part of the bay, 25 images were manually annotated. They were augmented into 120

tiles, with 90 tiles as training data and 30 tiles as validation data. The model trained with a weight decay of 0.01 and a learning rate of 1×10^{-4} for 23 epochs.

4.5 Bathymetry

The ultimate goal of this research is to create a bathymetric map from the segmented output. So far, the results of the segmentation approach will be a binary map, showing coral versus non coral. We are going to explore the possibilities to extract bathymetry from a coral density map. The coral density map is produced by sliding a Gaussian filter over the segmented output. The segmented output of Curaçao will be used. A Gaussian filter is a filter whose input is a Gaussian function. For our problem, this function represents a specific range (sigma) from a specific pixel in which the amount of coral pixels are calculated. The function slides over the image, from left to right and calculates the amount of coral pixels in a specific range for each individual pixel. In the binary map, coral pixels are given the value 255 and non coral pixels are given the value 0. If there are a lot of coral pixels in one specific range, the output is higher compared to when there are only non coral pixels in a range. This will create a 'blurred' image, ranging from values between 0 and 255. Two ranges of pixels or sigma values will be used, range of 100 and a range of 250 pixels. Later, pixels were grouped in 6 and 10 classes, in two different maps. Therefore, the result of this will be 4 different maps.

To create a final bathymetric map, the data needs to be validated to actual depth measurements. In our case, there is active sonar data available. Sonar stands for Sound Navigation and Ranging. Active sonar transducers emit an acoustic signal into the water. If an object is in the path of the sound pulse, the sound returns an "echo" to the sonar transducer. If the transducer is equipped with the ability to receive signals, it measures the strength of the signal. By determining the time between the emission of the sound pulse and its reception, the transducer can determine the depth of the object. The sonar data is taken by Paolo Stocchi by using a stand up paddle board. Unfortunately, while using a paddle board, it is not possible to move towards shallow areas. This can damage branching corals when the board is moving past them. There is no validation data available of the area that has been classified.

5 Evaluation

Even though evaluation is an important step in determining how well a model performs, evaluation criteria are often underrepresented in computer vision approaches. This means results of evaluation criteria are given, but a full explanation of the meaning of these criteria is often missing. However, in remote sensing, it is standard practise to clearly explain the evaluation metrics and to evaluate a classification using new data which was not used during training of the classifier (Maxwell et al. 2018). Evaluation is important to understand input and flaws of the data.

5.1 Metrics

5.1.1 Training and validation error

The training error and validation error are calculated over the performance of the algorithm. The training error is the error of our model as calculated on the training data set. Ideally, a model's performance is tested on a new, not yet classified, data set. However, it is not possible to generate a representative error over a new data set. However, it is possible to estimate a representative error using random selection of data examples that were withheld from our training set, which in this case is the validation set (Zhang et al. 2020). In an ideal setting, the data in the validation set is not an augmented set, as this augmented set can have corresponding features with the training set. This could introduce bias. Our research augmented the tiles first, which could have introduced bias.

5.1.2 Confusion Matrix

A confusion matrix, also known as an error matrix, is a table which is used to describe the performance of a classification model (table 1 and 2). It shows the correct and incorrect predictions made by a classifier. It shows four types of predictions: true positive, false positive, false negative and true negative. In the case of coral classification, true positive is when the model predicts coral and the ground truth (actual) is coral. True negative is when the model predicts non coral and the actual is non coral. False positive is when the model predicts coral, but actually it is non coral. False negative is when the model predicts non coral, but actually it is coral.

Table 1: Standard binary confusion matrix layout. The rows represent actual classification, the columns represent the predicted classification.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Table 2: The standard binary confusion matrix altered to the binary coral classification case.

		Predicted	
		Coral	Non Coral
Actual	Coral	TP	FN
	Non coral	FP	TN

5.1.3 Accuracy

Accuracy is a straight forward and intuitive way to evaluate a models performance. Accuracy is defined as stated in the equation below, using table 2.

$$Accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

For most binary classification problems, accuracy is a acceptable metric to evaluate models. This depends on the type of classification problem and the amount of class imbalance. This is because it takes into account true positive and true negative above the equation, which means it looks at how well it classifies coral and non coral together.

One example where accuracy can mislead is a health related classification problem such as predicting tumors (table 3). In this example, accuracy will be 90 percent, however, 9 people are classified as non tumor when they should be in the class tumor. A result like that can have large consequences therefore in this case, accuracy misleads. Fortunately, for coral classification this is less of an issue, as results do not have to be that precise. As our model classifies masks, it classifies a field of pixels as coral or non coral. A slight difference between manual annotation and classification results can already cause false positives and false negatives. Therefore, we can be satisfied when the model classifies a similar amount of true positives and true negatives compared to the actual data.

Table 3: An example of a confusion matrix where accuracy misleads the classification results.

		Predicted	
		Tumor	No Tumor
Actual	Tumor	1	9
	No Tumor	0	89

5.1.4 Intersection over Union

Intersection over Union (IoU) is commonly used in object detection problems. In object detection, a bounding box is drawn around an object. So if for example, an algorithm needs to recognize a flower in a field of grass. For the training set, the user makes a bounding box around the flower. In the test set, a bounding box around the flower will be created. In reality, it is extremely unlikely that the (x, y)-coordinates of our predicted bounding box are going to exactly match the (x, y)-coordinates of the ground-truth bounding box (Rosebrock 2016), which can be seen in figure 17. The position of the box is a bit arbitrary, it needs to be around the flower, but the exact position of what the model predicts compared to the human annotator will be different. The IoU metric only takes into account the true positive above the equation, which means it looks at how well it classifies only the coral class. Therefore, values are always lower compared to accuracy. An Intersection over Union score larger than 0.5 is normally considered a “good” prediction. The IoU metric is commonly applied to multi class segmentation problems, but it is also of use in binary classification problems. Therefore, whilst training and evaluating the model, the IoU is an important value to take into account.

In the case of a binary confusion matrix (table 1), the intersection over union can be calculated as followed:

$$IoU = \frac{TP}{(TP+TN+FP)}$$

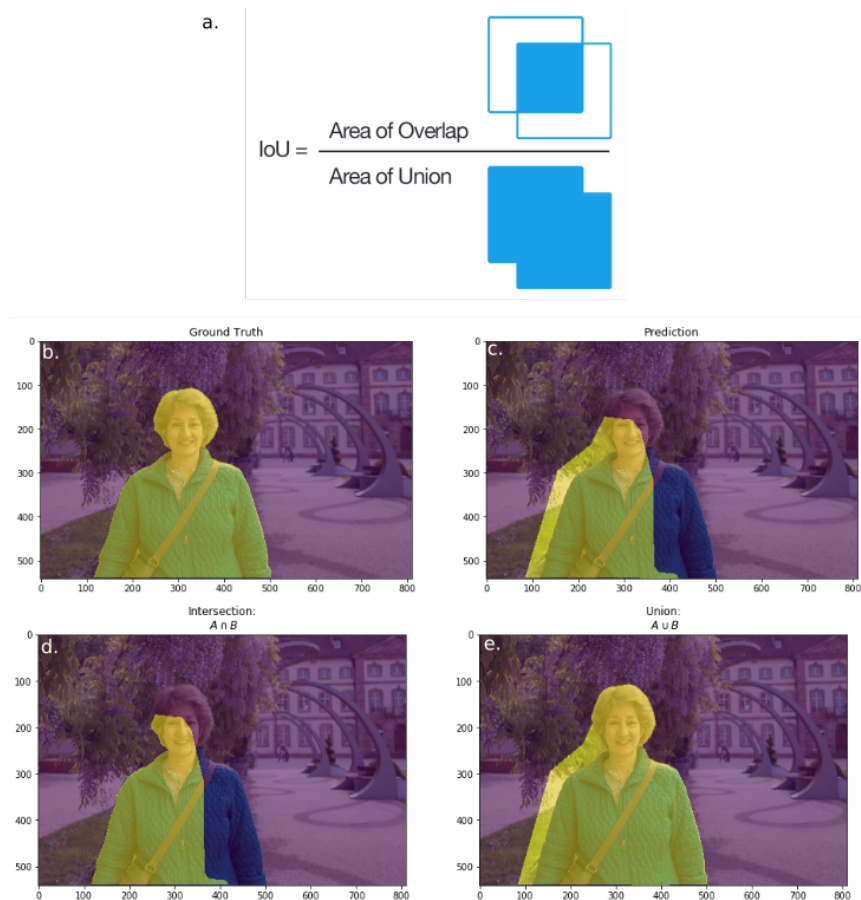


Figure 17: a) A graphical way to show the intersection over union, explained with b) ground truth, c) prediction, d) intersection of the ground truth and prediction and e) the union of ground truth and prediction, altered from Rosebrock (2016) and Jordan (2018)

5.1.5 Assembled metrics

With these evaluation techniques, it is only possible to evaluate the validation set. One way to evaluate the bays is by assembling all manually annotated tiles and comparing this manual annotation to classified orthophotomosaic. Then by using the confusion matrix as explained in table 1, it will provide accuracy and intersection over union measures. This will be different values compared to the validation set, as the validation set is chosen randomly from all augmented input data while all manually annotated tiles are fixed. In the rest of the thesis, these metrics explained will be described as assembled accuracy and assembled IoU. The calculation of the assembled metrics and the georeferencing process was done by making use of QGIS (QGIS Development Team 2009). The metrics will provide a more complete understanding of what the model is exactly doing.

6 Results

6.1 Binary

6.1.1 Moorea

Firstly, the performance of the model (learner) will be shown. It shows how well the model was trained, before validating or testing it to new data. As can be seen in table 4, the algorithm has trained successfully. Accuracy goes as high as 94 percent and intersection over union is 82 percent. The validation and training losses are low, and the difference between the two is little. This means that the model is not underfitting nor overfitting.

Table 4: The results of the binary Moorea algorithm after training for 16 epochs.

Training loss	Validation loss	Accuracy	IoU
0.14	0.15	94%	82%

Secondly, the validation metrics will be shown. As the validation set is always chosen randomly, validation metrics differ for each run. The average of five runs was taken as final validation metrics, which means the average of 5 averages of 32 tiles. The results of the confusion matrix can be seen in table 5. 91 percent of was classified as coral whilst 92 percent was classified as non coral. The average IoU over the 5 runs was 77 percent.

Table 5: Result of the various validation set runs for the Moorea classification.

		prediction	
		coral	non coral
actual	coral	91%	9%
	non coral	8%	92%

Besides validation metrics, we can also use the assembled metrics to check the results. Statistics of the assembled accuracy are given in a confusion matrix, in table 6. Overall assembled accuracy is 94 percent. The assembled IoU is 82 percent, which is similar compared to the learner and the validation metrics. It classified 89 percent of the corals correct and 97 percent of the non corals, which is a remarkable performance. Of the corals, 11 percent was predicted to be non coral. Only 3 percent of the non coral was predicted as coral. When comparing table 5 with table 6, the biggest difference is that the different between classes is higher in table 6 then compared to table 5. The percentages are in the same order of magnitude.

Table 6: The assembled metrics, a confusion matrix of the hand annotation compared to the test set.

		Predicted		
		corals	non corals	total
Actual	corals	89%	11%	100%
	non corals	3%	97%	100%
	total	92%	108%	

Figure 18 shows the manual annotation compared with the annotation classified by the algorithm. This is an example of what the values of the assembled metrics are based on. Visually, there is a lot

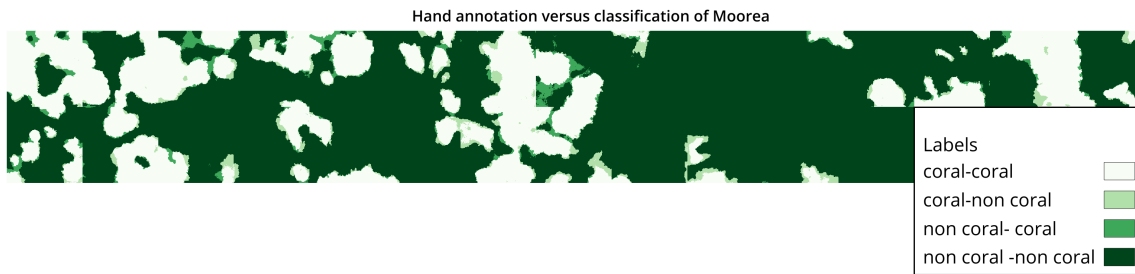


Figure 18: Comparing the manual annotation with the classification made by the CNN. White shows coral in both cases corals, dark green shows in both cases non coral. The two colours in between represent coral-non coral and non coral to coral respectively.

of overlap between the manual annotation and the classification. The algorithm even seems to pick up features which were not picked up by the human annotator. Figure 19b shows the final classified image, which is fully classified by the neural network. The algorithm is also able to detect more submerged objects, such as smaller corals and rocks. A clear cut between corals and non corals is seen, even though corals are patchy and can have fractal shapes.

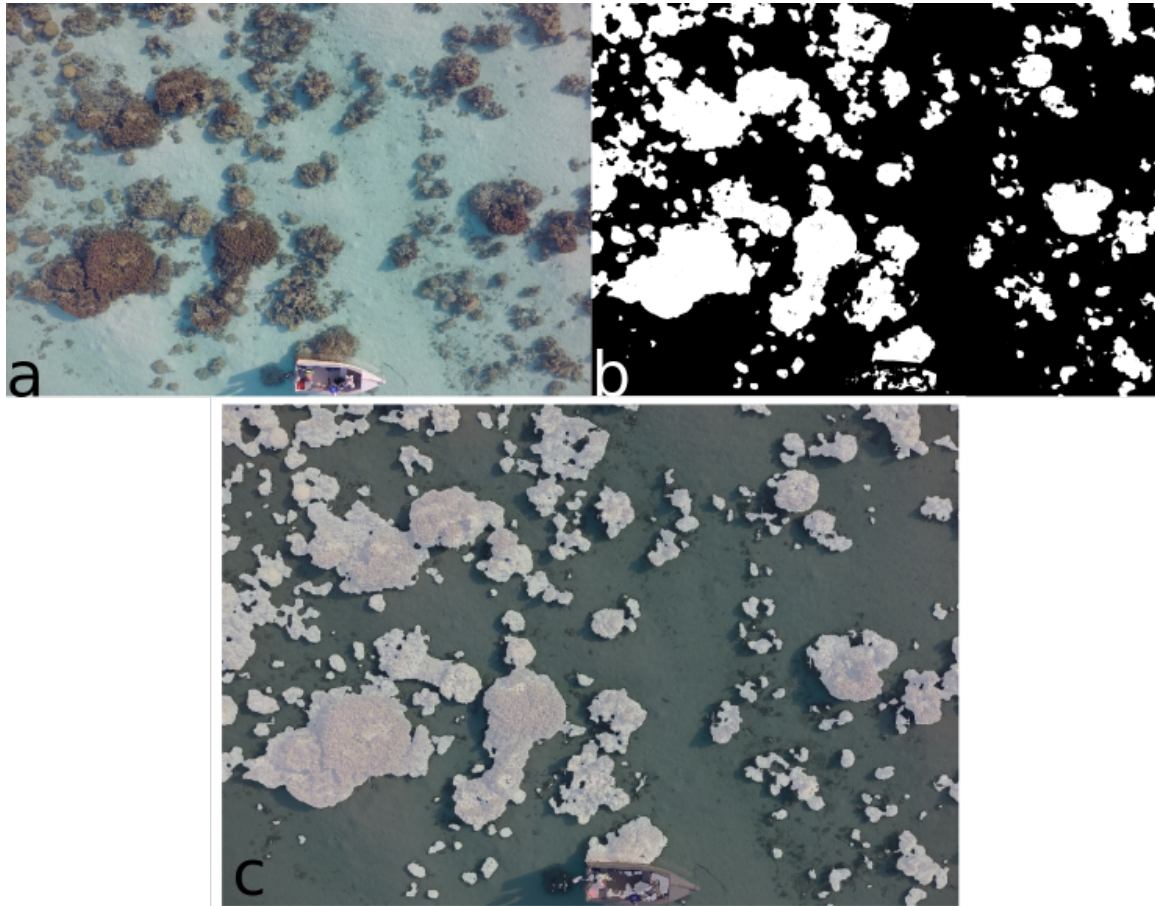


Figure 19: A figure showing (a) the initial Moorea photograph, (b) the classification made by the classifier and (c) the classification on top of the original image.

6.1.2 Curaçao

The results from the Curaçao classification are split up in the shallow part and the deep part. Classifying the dataset as a whole was also explored, results can be found in the appendix, in section C.1.

First the results of the shallow part, which will start with the results of the learner (table 7). Both accuracy and intersection over union are high, meaning that the learner trained well. The training and validation loss are low and the difference between them is not significant, it is probably not overfitting or underfitting.

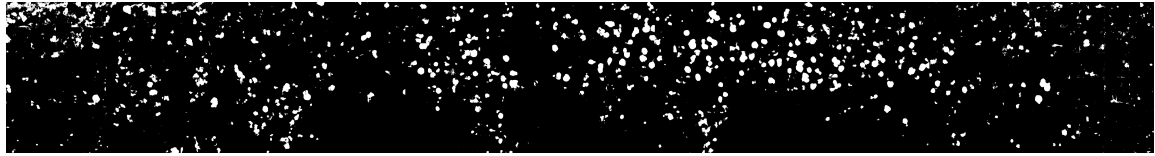
Table 7: Statistics of the Curaçao learner when trained on only augmented images from the shallow part.

Training loss	Validation loss	Accuracy	IoU
0.04	0.11	97%	83%

The same approach as for the Moorea dataset was taken to calculate the validation sets, meaning



(a) The cropped shallow reef Curaçao shoreline.



(b) The binary classification generated by the model, white is classified as coral and black is classified as non coral.



(c) The original image with the binary classification as overlay

Figure 20

the average of 5 random runs. From the validation metrics, it shows that both coral and non coral are recognized in equal amounts, both 86 percent. However, the average IoU calculated over the 5 random runs is much lower compared to the IoU of the learner, namely 42 percent, which would be below the threshold. This means the model is doing a good job in recognizing non coral, but is not doing a good job in recognizing coral. When looking at image 20, it visually shows that the model is recognizing corals, mostly brain and branching corals.

Table 8: Result of the various validation set runs for the shallow Curaçao classification

		prediction	
		coral	non coral
actual	coral	86%	14%
	non coral	13.4%	86.6%

For the assembled metrics, the same type of map as in figure 18 was made. However, since the manual annotated tiles were not chosen as a strip, but randomly, we did not to put that map in. The tiles do not fit together, and as one piece, it does not look representative. Instead, only the confusion matrix of the assembled tiles is shown, in table 9. The assembled IoU is 71 percent and the assembled accuracy is 96 percent. The difference between what is recognized is bigger compared to the validation metrics. 80 percent of corals is correctly classified, whilst 98 percent of non corals is correctly classified.

Table 9: The assembled metrics, a confusion matrix of the hand annotation compared to the test set of only the shallow part.

		predicted		
		corals	non corals	total
actuals	corals	80%	20%	100%
	non corals	2%	98%	100%
	total	82%	118%	

Secondly, the results of the deep part, which will start with the results of the learner in table 10. The difference between training and validation loss is relatively high. Accuracy is high with 91 percent, but IoU is just above the threshold of 50 percent.

Table 10: The results of the learner for the deep Curaçao classification

Training loss	Validation loss	Accuracy	IoU
0.19	0.27	91%	53%

The validation metrics however, look more promising as can be seen in table 11. As the both coral and non coral are classified with accuracies over 90 percent. The average IoU is 58 percent, which is higher compared to the learner.

Table 11: The results of the various validation runs for the deep Curaçao classification.

		prediction	
		coral	non coral
actual	coral	93.8%	6.2%
	non coral	5.2%	94.8%

As all metrics seem to point in the direction of a good classification, the result was expected to look accurate. However, this is not the case, as can be observed in figure 21. Besides, the image contains whole tiles which are classified as either non coral or coral, making the classification unusable. This 'checkerboard' output is the result of all runs and does not change when small tweaking options are changed.

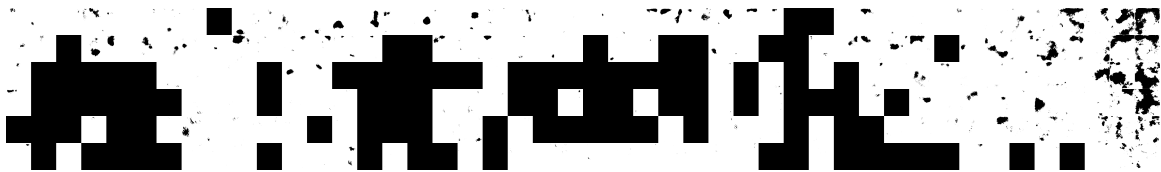
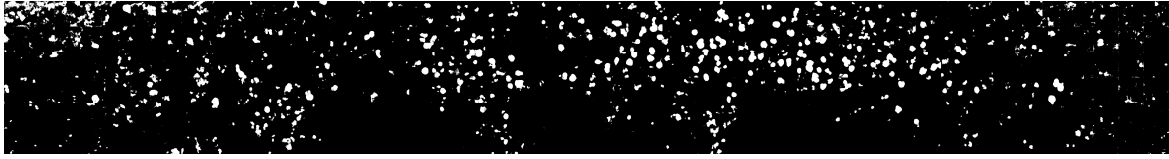


Figure 21: Final classification of the deep Curaçao area. The algorithm has difficulties recognizing corals and produces a 'checkerboard' output.

6.2 Coral Density

The coral density maps have been derived from the shallow shore classification. As the model was mostly picking up the brain corals, it represents brain coral density. The following corals density



(a) The binary classification generated by the model, white is classified as coral and black is classified as non coral.



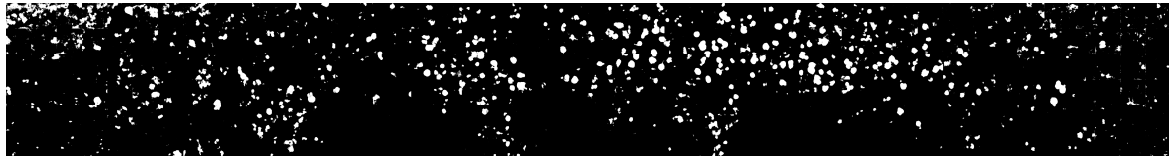
(b) Results after applying a Gaussian filter with the width of 250 pixels over image a, divided into 6 classes. The classes represent pixels values.



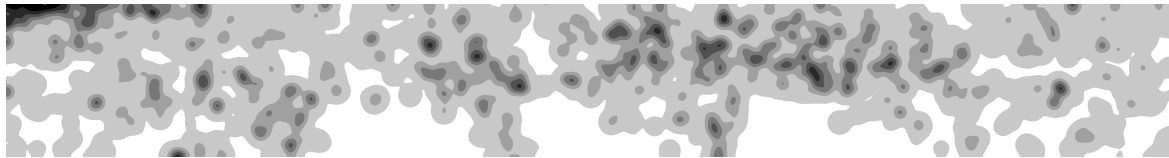
(c) Results after applying a Gaussian filter with the width of 250 pixels over image a, divided into 10 classes. The classes represent pixels values.

Figure 22

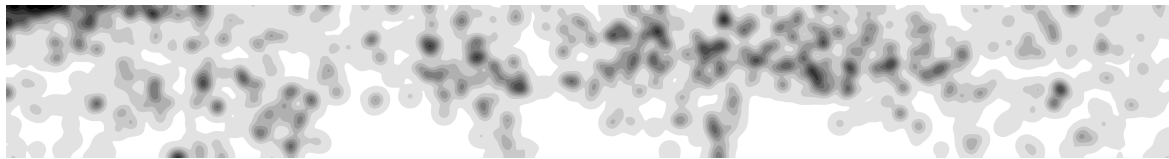
maps do not have units yet. The colour fields now represent specific pixel values, but what units they actually represent, needs to be derived first. For now, the maps represent where visually high abundances of (brain) corals are.



(a) The binary classification generated by the model, white is classified as coral and black is classified as non coral.



(b) Results after applying a Gaussian filter with the width of 100 pixels over image a, divided into 6 classes. The classes represent pixels values.



(c) Results after applying a Gaussian filter with the width of 100 pixels over image a, divided into 10 classes. The classes represent pixels values.

Figure 23

7 Discussion

7.1 Discussion on results

7.1.1 Moorea

The Moorea classification was successful and progressed relatively fast. The algorithm was quickly trained on minimal amount of hand annotated images. Therefore, for on shore pictures taken under ideal conditions, this approach is a success. The classified image, as seen in figure 19 looks visually correct. As the majority of tiles are classified with non coral, only stating accuracy would not be sufficient enough. The assembled IoU is 82 percent and validation IoU is 77 percent, which shows that the algorithm is detecting corals and not only detecting non coral. The Moorea algorithm has classified more non coral compared to coral (tab 6), but the difference is small, only 8 percent. The performance was still efficient enough to accurately classify 89 percent of coral pixels correctly. For figure 18, it seems like for some cases the model does a better job at recognizing the corals compared to the human annotator. Which could lead to the suggestion that in some ideal cases, a deep learning model could detect features better compared to the human annotator. This has not been investigated further and could be an interesting aspect to explore. In summary, with minimal effort and without the use of commercial software, this approach has brought an accurate binary classification in a more ideal case.

7.1.2 Curaçao

Classifying the Curaçao dataset proved to be more complex compared to the Moorea data set. There was chosen to split up the data set into a deep and a shallow part, as the deep classification proved to have some difficulties. After several attempts, the shallow classification provided visually accurate

results. With an overall assembled accuracy of 96 percent and an assembled IoU of 71 percent, the shallow classification can be called successful. However, the IoU of the validation set is much lower, only 42 percent. Which could lead to the suggestion that the model is actually better in classifying non coral compared to coral. That could mean that the data input of the validation set is imbalanced, meaning there were more tiles containing mostly non coral compared to tiles containing coral. It is unclear why the difference between the validation IoU and assembled IoU is high. It could be the random tiles which were in the validation runs were of less quality compared to the fixed manually annotated masks, which caused the assembled IoU to be higher.

Even though the metrics of the deep data set look promising, the final classification has a checkerboard layout. Some tiles are fully classified as coral while others are fully classified as non coral. The difference between training and validation loss shown in table 10 suggest that the machine is overfitting. This is could be why it is able to recognize corals in some tiles, but is not able to classify corals in others. This phenomenon can also explain the high validation numbers, as the validation set contains similar images as the test set. Another possibility for mismatch in the deep classification could be that the deep data set is highly imbalanced, as there are more non-coral pixels compared to coral pixels. As was mentioned in the section 5.1.3, an imbalanced data set could lead to biased results. The combination of blurry imagery, possibility of overfitting and a most likely imbalanced data set, was most likely the cause for mismatch in classifying the deep data set.

7.1.3 Comparison

When comparing the results to section 2.6, we find similar ranges of accuracies. Chabot et al. (2018), Collin et al. (2018), Ventura et al. (2018), and Wicaksono et al. (2019) find accuracies ranging from 70 to circa 90 percent. Our results are on the higher side of the spectrum, with accuracies going as high as 96 percent. None of the papers mentioned in section 2.6 makes use of the IoU metric. The introduction of the IoU in this research led to a higher understanding if the deep learning algorithm was actually classifying corals instead of non coral. It would therefore a great addition to other (remote sensing) classification literature, especially for future comparisons.

Chabot et al. (2018) find difficulties in distinction between over-segmentation of shallow, bright features and under-segmentation in deeper and darker features. We do not observe these patterns in this approach, mostly due to the quality of the deep water imagery. Over-segmentation in the shallow region is also not observed. All literature presented perform multi class segmentation approaches, whose performance is harder to compare to a binary approach. Multi class segmentation can be more challenging as more classes are harder for the machine to classify, which is what also can be found in this study. This research experimented with a multi class approach (appendix section C.2) however, there was not enough data available to accurately classify the imagery. The results of the metrics were not good enough to consider it a sufficient classification. Multi class classification for drone imagery should be researched more in order to be of use for the remote sensing community.

According to the IoU and the accuracy, the classification of the Moorea data set was better compared to the Curaçao set, as the value is higher. Even though the approach taken for Moorea and Curaçao is similar, it is difficult to make assumptions about a possible comparison between both data sets. The data is taken in different environments, the bays consist of different corals and the imagery is taken under completely different environmental settings. It should be possible to compare both classifications, when there is a non ideal and an ideal image available from both places. A metric or number could be made, in which we can assess the quality (f.e. the resolution and environmental conditions) of the imagery. This could provide more knowledge about the quality of the classification and could guide future research in estimating when data is usable for segmentation purposes.

We can be satisfied with the classified result as this research explored new research fields (ML and DL) and managed to deliver an accurate outcome compared to conventional methods such as OBIA. The benefit of OBIA is its possibility to export specific characteristics with the data. One of the limitations of OBIA is that it is highly dependent both on the kind of imagery and on the user experience (Ventura et al. 2018). Therefore, Ventura et al. (2018) recommends more consistent and less subjective procedures for feature selection and this is where CNNs can play an important role. However, as the CNN recognizes edges, shapes and features, the export of the specific characteristics is not necessary, which according to Chabot et al. (2018) might be a plus side in using deep learning compared to OBIA. This study proved that the automatic recognition of certain aspects was an addition to the workflow, as it in some cases even recognized features which the human annotator did not take into account.

7.2 Problems which arose during classification

7.2.1 Edge artifacts

This approach required the deconstruction of the generated orthophotomosaic and the reassembling of the orthophotomosaic. This has caused the final classification of the shallow bay to have some edge artifacts. This means in one tile a coral is recognized as a coral, whilst in another it is not (fig. 24). Whether the edge artifacts are a cause for concern depends on the input resolution of the geophysical models used. For some models, it can be good enough that the model has recognized a corals on the right position, for others it could need more detail. However, it is more accurate compared to complete tiles being misclassified, which happened when classifying the whole bay and the deep area. Besides, when classifying a large bay, the assembling process cannot be avoided, as enough input has to be generated.

7.2.2 Augmentation effects

It is often mentioned that a deep learning algorithm needs a lot of data to train on (González-Rivero et al. 2016), which was the bottleneck for this study. The goal was to manually annotate as little as possible, however, there is still a 'minimal' requirement for a annotation to work well. Even with data augmentation, there is a possibility that there was too little data for the model to accurately train on. Not all models suggests overfitting, but there is a slight chance they all suffer from overfitting because of high similarities in input data. Further research could explore the possibilities for minimal data input for deep learning using drone imagery in coral reef environments. Still, deep learning will play an important role in marine classification regarding big data problems. It would be the method of choice when the objective of the project is to solve big data problems in ocean ecosystems which are composed of thousands of images (Raphael et al. 2020).

7.2.3 Validation set

The usage of augmented images is normal in the deep learning community. However, there is a crucial difference in the learning process when data augmentation is applied. In this study, the images were augmented before they were put into both the training and validation set. This has caused for validation metric problems. The validation set is using already augmented images to base their metrics on, whilst the test set is the original. There is a chance the validation set contains several augmented images of the same tile, causing the total validation accuracy to drop, whilst this is not representative of the whole data set. As the learner automatically takes a random amount of tiles for training and validation set, the metrics differ every time the model runs. It would have

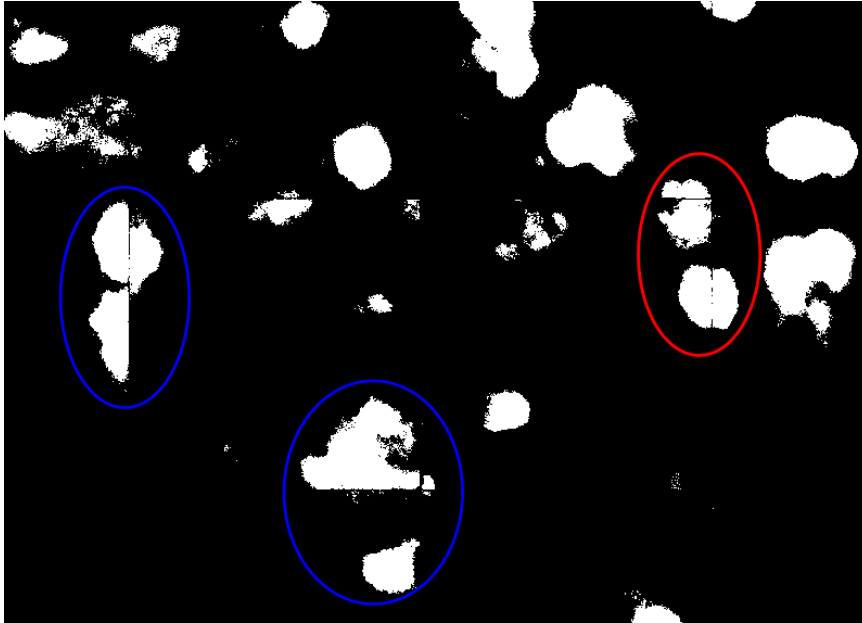


Figure 24: A zoomed in example of edge artifacts which can be found in the shallow Curaçao classification. The red circle represents a correctly classified coral. The blue circles represent incorrect classified corals, where on one tile they are classified as corals, whilst on another, they are not.

been better to augment only the images in the training set, so the validation set is not harmed. This would not lead to potential duplicates in the validation set. There was not chosen for this approach, since there was not enough data to only augment the training set, as only minimal amount of tiles would then be present in the validation set. Now, the amount of data in the validation set is around 30/40 tiles, which normally would be enough tiles to provide a good metric analysis. However, the addition of the assembled tile metric provides a good back up to evaluate the outcome of the model. Because of the augmented validation tiles, this has led to differences in validation accuracy per run.

This is why we chose to average this over 5 different runs, which would mean 5 different validation sets. The assembled metrics were added to complete the picture, however, in essence, this is the similar to a normal validation set. Still, there could be repetition, because tiles which are in the assembled orthophotomosaic could also be in the training set. For future research, it is recommended to keep the validation and training sets completely separate. This means that only the training set contains augmented tiles, whilst the validation set only contains tiles which were not used during training. This will provide the most representative result on how the model is performing.

A deficiency of this approach is; the trained model is not fed with a completely new data set, but with a test set which contains images which are also in the training and validation set, even in augmented version. Therefore, if the machine is fed a different bay as test set it will mostly likely not classify that bay correctly. The amount of data used as training set is also too little to generalise a learner. To remove of this problem a repository such as CoralNet could be made, which could be a researchers collaboration or a more community effort. Separate imagery of different orthophoto mosaics could be bundled. With the usage of Pixel Annotator, images can quickly be manually labelled. This could generalize future learners and it can potentially be used on different study sites. When the repository grows big enough this might also filter out manual annotation errors.

7.2.4 Image enhancing

A small shortcoming of this research is that it is not possible to clean data with CNNs yet. The learner will perform on the quality of data that is put into the model. Therefore, if the quality is poor, the CNN will not be able to clearly detect low laying semantics. High level semantics are even more difficult, which will lead to the learner not being able to detect features and therefore not being able to classify parts of tiles correctly. In the deeper part of the orthophotomosaic, the tiles have different colors due to the existence of corals below the water surface. If there are corals below images appear brown, whilst when there are no corals the images appear blue. This is where the learner has difficulties in classifying. As the model trains on shapes, edges and colors, when both edges and shapes are not clearly present, it is mostly likely it fully classifies on color. Therefore, whole tiles are classified as either coral or non coral. For future research on non ideal data, there could be looked at image enhancing techniques before starting the with a machine learning algorithm. For example, looking into recent developments in sharpening (using GANs) or applying certain filters to reduce noise. After this, there could be looked at which algorithm performs the best after image enhancing techniques.

7.2.5 Fast.ai

The fast.ai library proved to be an easy tool to apply deep learning to the data. For usage, the library can be used when a user has experience with programming. During the scope of this research, around September 2020, fast.ai released a new version of their library, with full-on new documentation and lessons. The algorithm I produced was just finished and working correctly, as it took some time to build the training algorithm. The new version might be more efficient, but the possibilities have not been explored, due to lack of time. Further research could explore the usage of fast.ai v2 in classifying coral reef imagery.

7.3 Bathymetric map

The initial goal of this study was to produce a depth map from the classified data. From the coral density maps, shown section 6.2 a bathymetric map could be derived when there is other data available. As explained in section 4.5 there is sonar data available. However, as explained the sonar data does not extend to the range of the classified image. Thus, we now have a classification of an area which normally would be difficult to acquire without damaging the branching corals. In this case this represents a step forward in acquiring data, as we have obtained a data from an area which is normally hard to survey. The results shown in figure 22b and figure 22c could represent a more gradual surface of the slope whilst figure 23b and 23c can represent islands of higher and lower depth, in the form of brain corals. All coral density figures show higher order features in terms of the amounts of corals and the 'valleys' between the corals.

The coral density maps could be linked to validation points in order to extrapolate a depth profile. Fortunately, instead of surveying the whole bay with expensive LiDAR, sonar points could provide a good and cheaper alternative for validation points. Even walking along the shore and/or walking in the shallow area using a sonar transmitter, should provide enough data points to correlate to the density maps. This could represent a step forward in making bathymetry maps and is something which is definitely worth exploring in future research.

If the multi class classification had succeeded, this would have provided an extra option produce bathymetric output. Different classes could represent a different height level, for example, corals are most likely to be close to sea level while sand is most likely to be lower than sea level. Ideally, the

multi class classification output is also combined with validation points or even also combined with the coral density maps.

7.4 Importance of this research

This project sparked the interest of The Caribbean Research and Management of Biodiversity (CARMABI) and especially for usage in the SEALINK project (<https://www.sealinkcaribbean.net/>). The SEALINK project will assess how land-derived and waterborne inputs affect the growth and survival of coral reefs in the Dutch Caribbean. It caused a higher interest to invest in drones and to explore the possibilities of drone imagery. They could, for example, build towards the first orthophotomosaic repository which could be used to classify several Caribbean coasts. This research classified an area which is normally hard to reach by conventional methods, as this area is very shallow and covered by branching corals. Further research could explore the possibilities of classifying of these super shallow areas, and maybe extrapolate results to deeper areas. Besides sparking the interest of CARMABI, this research touched different aspects for coral classification, combining knowledge from remote sensing and computer sciences. Both research fields could benefit in exploring each others methodology. The approach taken in this research could benefit from more expert knowledge from both fields.

8 Conclusion

This research explored exciting new possibilities for classifying drone imagery, combining approaches from computer science and remote sensing. This research proved that it is possible to use deep learning for the classification of UAV shallow shore imagery. We succeeded in classifying both ideal and non-ideal data with a binary classification. The approach was definitely successful for imagery taken under ideal conditions, while it was acceptable when applying on imagery taken during non ideal conditions. However, the approach would benefit from follow up research for the classification of non-ideal data. For ideal data, this method provided a quick classification, with minimal labelling effort and without the usage of commercial software.

Producing geophysical output in the form of bathymetry from the segmentation should be possible as well. We have succeeded to derive a coral density map from the binary segmentation, which can later be used to produce a bathymetric map when validation points are available. The remote sensing community would definitely benefit from exploring the possibilities of using CNNs besides only using OBIA. CNNs and OBIA work with similar characteristics, but CNNs provide the possibility of low to free costs and are able to run everywhere on every machine. Future research could focus on the exploration of deep learning to different types remote sensing options, as well as to combining data and finding more ways to accurately classify bays.

9 Acknowledgements

When I was starting this thesis, the pandemic just started and I was still hoping to end it on NIOZ. Sadly, the situation did not change and I had to finish my project at home. I would have loved to have met some more people and discuss this over coffee, but I think we have made the best of it. Still, I have enjoyed this project very much and most importantly, I have gained a lot of knowledge and gained new skills. I want to thank my supervisors, Paolo and Adam, for helping me and meeting up periodically. This has helped me keep my enthusiasm to dig deeper. I want to thank Francesca for being the UU supervisor, even though this is not her area of expertise. I want to thank my thesis working group, even though we stopped meeting in November. It was nice to share the beginning of thesis struggles with you. Finally, I want to thank Laurens, for teaching me stuff about neural nets and convolutions and always being there to listen to my stories.

References

- Airoidi, L (2003). “Effects of patch shape in intertidal algal mosaics: roles of area, perimeter and distance from edge”. In: *Marine Biology* 143.4, pp. 639–650.
- Alonso, Alice, Rafael Muñoz-Carpena, Robert E Kennedy, and Carolina Murcia (2016). “Wetland landscape spatio-temporal degradation dynamics using the new Google Earth Engine cloud-based platform: Opportunities for non-specialists in remote sensing”. In: *Transactions of the ASABE* 59.5, pp. 1331–1342.
- Alonso, Iñigo, Matan Yuval, Gal Eyal, Tali Treibitz, and Ana C. Murillo (2019). “CoralSeg: Learning coral segmentation from sparse annotations”. In: *Journal of Field Robotics* 36.8, pp. 1456–1477. ISSN: 15564967. DOI: 10.1002/rob.21915.
- Badea, Mihai-Sorin, Iulian-Ionuț Felea, Laura Maria Florea, and Constantin Vertan (2016). “The use of deep learning in image segmentation, classification and detection”. In: *arXiv preprint arXiv:1605.09612*.
- Bakker, Didier M. de, Fleur C. van Duyl, Rolf P.M. Bak, Maggy M. Nugues, Gerard Nieuwland, and Erik H. Meesters (2017). “40 Years of benthic community change on the Caribbean reefs of Curaçao and Bonaire: the rise of slimy cyanobacterial mats”. In: *Coral Reefs* 36.2, pp. 355–367. ISSN: 07224028. DOI: 10.1007/s00338-016-1534-9.
- Beijbom, Oscar, Peter J. Edmunds, David I. Kline, B. Greg Mitchell, and David Kriegman (2012). “Automated annotation of coral reef survey images”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1170–1177. ISSN: 10636919. DOI: 10.1109/CVPR.2012.6247798.
- Bennett, Mary K and Nicolas Younes (2020). “Automating Drone Image Processing to Map Coral Reef Substrates Using Google Earth Engine”. In: pp. 1–13.
- Blaschke, Thomas et al. (2014). “Geographic object-based image analysis—towards a new paradigm”. In: *ISPRS journal of photogrammetry and remote sensing* 87, pp. 180–191.
- Bloice, Marcus D, Peter M Roth, and Andreas Holzinger (Apr. 2019). “Biomedical image augmentation using Augmentor”. In: *Bioinformatics* 35.21, pp. 4522–4524. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz259. eprint: <https://academic.oup.com/bioinformatics/article-pdf/35/21/4522/30330763/btz259.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btz259>.
- Bréhéret, Amaury (2017). *Pixel Annotation Tool*. <https://github.com/abreheret/PixelAnnotationTool>.
- Casella, Elisa, Antoine Collin, Daniel Harris, Sebastian Ferse, Sonia Bejarano, Valeriano Parravicini, James L. Hench, and Alessio Rovere (2017). “Mapping coral reefs using consumer-grade drones and structure from motion photogrammetry techniques”. In: *Coral Reefs* 36.1, pp. 269–275. ISSN: 07224028. DOI: 10.1007/s00338-016-1522-0.
- Castelluccio, Marco, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva (2015). “Land Use Classification in Remote Sensing Images by Convolutional Neural Networks”. In: August. arXiv: 1508.00092. URL: <http://arxiv.org/abs/1508.00092>.
- Chabot, Dominique, Christopher Dillon, Adam Shemrock, Nicholas Weissflog, and Eric PS Sager (2018). “An object-based image analysis workflow for monitoring shallow-water aquatic vegetation in multispectral drone imagery”. In: *ISPRS International Journal of Geo-Information* 7.8, p. 294.
- Chen, Liang Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam (2018). “Encoder-decoder with atrous separable convolution for semantic image segmentation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11211 LNCS, pp. 833–851. ISSN: 16113349. DOI: 10.1007/978-3-030-01234-2_49. arXiv: 1802.02611.
- Collin, Antoine et al. (2018). “Very high resolution mapping of coral reef state using airborne bathymetric LiDAR surface- intensity and drone imagery”. In: *International Journal of Remote Sensing* 39.17, pp. 5676–5688. ISSN: 0143-1161. DOI: 10.1080/01431161.2018.1500072. URL: <https://doi.org/10.1080/01431161.2018.1500072>.
- Cracknell, Matthew J and Anya M Reading (2014). “Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information”. In: *Computers & Geosciences* 63, pp. 22–33.
- Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.
- Dertat, Arden (Nov. 2017). *Applied Deep Learning - Part 4: Convolutional Neural Networks*. URL: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>.
- Gandhi, Rohith (June 2018). *Support Vector Machine — Introduction to Machine Learning Algorithms*. Towards Data Science. URL: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.
- Gardner, Toby A., Isabelle M. Côté, Jennifer A. Gill, Alastair Grant, and Andrew R. Watkinson (2003). “Long-term region-wide declines in Caribbean corals”. In: *Science* 301.5635, pp. 958–960. ISSN: 00368075. DOI: 10.1126/science.1086050.
- Gogul, I. and V. Sathiesh Kumar (2017). “Flower species recognition system using convolution neural networks and transfer learning”. In: *2017 4th International Conference on Signal Processing, Communication and Networking, ICSCN 2017* March. doi: 10.1109/ICSCN.2017.8085675.

- Gomes, Inês, Laura Peteiro, Juan Bueno-Pardo, Rui Albuquerque, Sergi Pérez-Jorge, Eduardo R. Oliveira, Fátima L. Alves, and Henrique Queiroga (2018). “What’s a picture really worth? On the use of drone aerial imagery to estimate intertidal rocky shore mussel demographic parameters”. In: *Estuarine, Coastal and Shelf Science* 213. April, pp. 185–198. ISSN: 02727714. DOI: 10.1016/j.ecss.2018.08.020. URL: <https://doi.org/10.1016/j.ecss.2018.08.020>.
- Gómez-Ríos, Anabel, Siham Tabik, Julián Luengo, A. S.M. Shihavuddin, Bartosz Krawczyk, and Francisco Herrera (2019). “Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation”. In: *Expert Systems with Applications* 118, pp. 315–328. ISSN: 09574174. DOI: 10.1016/j.eswa.2018.10.010.
- González-Rivero, Manuel et al. (2020). “Monitoring of Coral Reefs Using Artificial Intelligence: A Feasible and Cost-Effective Approach”. In: *Remote Sensing* 12.3, p. 489.
- González-Rivero, Manuel, Oscar Beijbom, Alberto Rodríguez-Ramírez, Tazio Holtrop, Yeray González-Marrero, Anjani Ganase, Chris Roelfsema, Stuart Phinn, and Ove Hoegh-Guldberg (2016). “Scaling up ecological measurements of coral reefs using semi-automated field image collection and analysis”. In: *Remote Sensing* 8.1. ISSN: 20724292. DOI: 10.3390/rs8010030.
- Goodman, James A, Samuel J Purkis, and Stuart R Phinn (2013). “Coral reef remote sensing”. In: *A guide for mapping, monitoring and management*. 436p.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Hedley, John D et al. (2016). “Remote sensing of coral reefs for monitoring and management: a review”. In: *Remote Sensing* 8.2, p. 118.
- Hernández-Cruz, Luz Raquel, Samuel J Purkis, and Bernhard M Riegl (2006). “Documenting decadal spatial changes in seagrass and Acropora palmata cover by aerial photography analysis in Vieques, Puerto Rico: 1937–2000”. In: *Bulletin of Marine Science* 79.2, pp. 401–414.
- Heron, Scott F, Jeffrey A Maynard, Ruben Van Hooidek, and C Mark Eakin (2016). “Warming trends and bleaching stress of the world’s coral reefs 1985–2012”. In: *Scientific reports* 6.1, pp. 1–14.
- Hochberg, Eric J, Marlin J Atkinson, and Serge Andre (2003). “Spectral reflectance of coral reef bottom-types worldwide and implications for coral reef remote sensing”. In: 85, pp. 159–173. DOI: 10.1016/S0034-4257(02)00201-8.
- Howard, Jeremy and Sylvain Gugger (2020). “Fastai: A layered api for deep learning”. In: *Information (Switzerland)* 11.2, pp. 1–26. ISSN: 20782489. DOI: 10.3390/info11020108.
- Jawak, Shridhar D, Somashekhar S Vadlamani, Alvarinho J Luis, et al. (2015). “A synoptic review on deriving bathymetry information using remote sensing technologies: models, methods and comparisons”. In: *Advances in remote Sensing* 4.02, p. 147.
- Jordan, Jeremy (Dec. 2018). *Evaluating image segmentation models*. URL: <https://www.jeremyjordan.me/evaluating-image-segmentation-models/>.
- Kerr, Jeremy T and Marsha Ostrovsky (2003). “From space to species: ecological applications for remote sensing”. In: *Trends in ecology & evolution* 18.6, pp. 299–305.
- King, Andrew, Suchendra M. Bhandarkar, and Brian M. Hopkinson (2018). “A comparison of deep learning methods for semantic segmentation of coral reef survey images”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 2018-June, pp. 1475–1483. ISSN: 21607516. DOI: 10.1109/CVPRW.2018.00188.
- Klemas, Victor (2011). “Beach profiling and LIDAR bathymetry: An overview with case studies”. In: *Journal of Coastal Research* 27.6, pp. 1019–1028.
- Lary, David J., Amir H. Alavi, Amir H. Gandomi, and Annette L. Walker (2016). “Machine learning in geosciences and remote sensing”. In: *Geoscience Frontiers* 7.1, pp. 3–10. ISSN: 16749871. DOI: 10.1016/j.gsf.2015.07.003. URL: <http://dx.doi.org/10.1016/j.gsf.2015.07.003>.
- Leon, J X, Chris M Roelfsema, Megan I Saunders, and Stuart R Phinn (2015). “Geomorphology Measuring coral reef terrain roughness using ‘Structure-from-Motion’ close-range photogrammetry”. In: *Geomorphology* 242, pp. 21–28. ISSN: 0169-555X. DOI: 10.1016/j.geomorph.2015.01.030. URL: <http://dx.doi.org/10.1016/j.geomorph.2015.01.030>.
- Li, Fei-Fei, Justin Johnson, and Serena Yeung (May 2017). *Lecture 11: Detection and Segmentation*.
- Li, Wenwen and Chia-Yu Hsu (2020). “Automated terrain feature identification from remote sensing imagery: a deep learning approach”. In: *International Journal of Geographical Information Science* 34.4, pp. 637–660.
- Liu, Fangfang and Ming Fang (2020). “Semantic segmentation of underwater images based on improved deep lab”. In: *Journal of Marine Science and Engineering* 8.3. ISSN: 20771312. DOI: 10.3390/jmse8030188.
- Mas, Jean F and Juan J Flores (2008). “The application of artificial neural networks to the analysis of remotely sensed data”. In: *International Journal of Remote Sensing* 29.3, pp. 617–663.

- Maxwell, Aaron E, Timothy A Warner, and Fang Fang (2018). “Implementation of machine-learning classification in remote sensing: An applied review”. In: *International Journal of Remote Sensing* 39.9, pp. 2784–2817.
- Monismith, Stephen G. (2007). “Hydrodynamics of coral reefs”. In: *Annual Review of Fluid Mechanics* 39, pp. 37–55. ISSN: 00664189. DOI: 10.1146/annurev.fluid.38.050304.092125.
- Murfitt, Sarah L., Blake M. Allan, Alecia Bellgrove, Alex Rattray, Mary A. Young, and Daniel Ierodiaconou (2017). “Applications of unmanned aerial vehicles in intertidal reef monitoring”. In: *Scientific Reports* 7.1, pp. 1–11. ISSN: 20452322. DOI: 10.1038/s41598-017-10818-9. URL: <http://dx.doi.org/10.1038/s41598-017-10818-9>.
- Nagendra, Harini (2001). “Using remote sensing to assess biodiversity”. In: *International journal of remote sensing* 22.12, pp. 2377–2400.
- Parsons, Mark, Dmitry Bratanov, Kevin J. Gaston, and Felipe Gonzalez (2018). “UAVs, hyperspectral remote sensing, and machine learning revolutionizing reef monitoring”. In: *Sensors (Switzerland)* 18.7, pp. 1–20. ISSN: 14248220. DOI: 10.3390/s18072026.
- Pendleton, Linwood and Peter Edwards (2017). “Measuring the human ‘so what’ of large-scale coral reef loss?” In: *Biodiversity* 18.1, pp. 13–15. ISSN: 21600651. DOI: 10.1080/14888386.2017.1308271. URL: <http://doi.org/10.1080/14888386.2017.1308271>.
- Philipson, Petra and Tommy Lindell (2003). “Can coral reefs be monitored from space?” In: *AMBIO: A Journal of the Human Environment* 32.8, pp. 586–593.
- Phinn, Stuart R, Chris M Roelfsema, and Peter J Mumby (2012). “Multi-scale, object-based image analysis for mapping geomorphic and ecological zones on coral reefs”. In: *International Journal of Remote Sensing* 33.12, pp. 3768–3797.
- QGIS Development Team (2009). *QGIS Geographic Information System*. Open Source Geospatial Foundation. URL: <http://qgis.org>.
- Raphael, Alina, Zvy Dubinsky, David Iluz, and Nathan S. Netanyahu (2020). “Neural network recognition of marine benthos and corals”. In: *Diversity* 12.1. ISSN: 14242818. DOI: 10.3390/d12010029.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351, pp. 234–241. ISSN: 16113349. DOI: 10.1007/978-3-319-24574-4_28. arXiv: 1505.04597.
- Rosebrock, Adrian (Nov. 2016). *Intersection over Union (IoU) for object detection*. URL: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>.
- Saul, Steven and Sam Purkis (2015). “Semi-automated object-based classification of coral reef habitat using discrete choice models”. In: *Remote Sensing* 7.12, pp. 15894–15916.
- Shihavuddin, A. S.M., Nuno Gracias, Rafael Garcia, Arthur C.R. Gleason, and Brooke Gintert (2013). “Image-based coral reef classification and thematic mapping”. In: *Remote Sensing* 5.4, pp. 1809–1841. ISSN: 20724292. DOI: 10.3390/rs5041809.
- Smith, Leslie N (2017). “Cyclical learning rates for training neural networks”. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 464–472.
- Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.
- Ventura, Daniele, Andrea Bonifazi, Maria Flavia Gravina, Andrea Belluscio, and Giandomenico Ardizzone (2018). “Mapping and classification of ecologically sensitive marine habitats using unmanned aerial vehicle (UAV) imagery and Object-Based Image Analysis (OBIA)”. In: *Remote Sensing* 10.9, pp. 1–23. ISSN: 20724292. DOI: 10.3390/rs10091331.
- Verikas, Antanas, Evaldas Vaiciukynas, Adas Gelzinis, James Parker, and M. Charlotte Olsson (2016). “Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness”. In: *Sensors (Switzerland)*. ISSN: 14248220. DOI: 10.3390/s16040592.
- Wan, Jiaxin and Yi Ma (2020). “Multi-scale Spectral-Spatial Remote Sensing Classification of Coral Reef Habitats Using CNN-SVM”. In: *Journal of Coastal Research* 102.SI, pp. 11–20.
- Wang, Kai, Steven E Franklin, Xulin Guo, and Marc Cattet (2010). “Remote sensing of ecology, biodiversity and conservation: a review from the perspective of remote sensing specialists”. In: *Sensors* 10.11, pp. 9647–9667.
- Westoby, Matthew J, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds (2012). “Structure-from-Motion photogrammetry: A low-cost, effective tool for geoscience applications”. In: *Geomorphology* 179, pp. 300–314.
- Wicaksono, Pramaditya, Prama Ardha Aryaguna, and Wahyu Lazuardi (2019). “Benthic habitat mapping model and cross validation using machine-learning classification algorithms”. In: *Remote Sensing* 11.11, p. 1279.
- Williams, Ivor D., Courtney Couch, Oscar Beijbom, Thomas Oliver, Bernardo Vargas-Angel, Brett Schumacher, and Russell Brainard (2019). “Leveraging automated image analysis tools to transform our capacity to assess status and trends on coral reefs”. In: *Frontiers in Marine Science* 6.APR, pp. 1–14. ISSN: 22967745. DOI: 10.3389/fmars.2019.00222.

- Wu, Zhihuan, Yongming Gao, Lei Li, Junshi Xue, and Yuntao Li (2019). “Semantic segmentation of high-resolution remote sensing images using fully convolutional network with adaptive threshold”. In: 0091. DOI: 10.1080/09540091.2018.1510902. URL: <https://doi.org/10.1080/09540091.2018.1510902>.
- Yahel, Gitai, Anton F Post, Katharina Fabricius, Dominique Marie, Daniel Vulot, and Amatzias Genin (1998). “Phytoplankton distribution and grazing near coral reefs”. In: *Limnology and Oceanography* 43.4, pp. 551–563.
- Yu, Le et al. (2014). “Meta-discoveries from a synthesis of satellite-based land-cover mapping research”. In: *International Journal of Remote Sensing* 35.13, pp. 4573–4588.
- Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola (2020). *Dive into Deep Learning*. <https://d2l.ai>.
- Zharikov, Yuri, Greg A Skilleter, Neil R Loneragan, Thomas Taranto, and Bronwyn E Cameron (2005). “Mapping and characterising subtropical estuarine landscapes using aerial photography and GIS for potential application in wildlife conservation and management”. In: *Biological Conservation* 125.1, pp. 87–100.

Appendices

A Extra background

A.1 Deep learning

A.1.1 Neuron

A neuron is a computational element and the building block of an ANN. The most common model of a neuron is depicted in figure 25. A neuron has a set of inputs x_1, \dots, x_m . Each connection from the input to the processing unit is affected by different strengths called synaptic weights. A signal x_j at the input of synapse or connection j , connected to neuron k , is multiplied by synaptic weight w_{kj} . An adder sums all inputs forming a linear combination of them. An activation function is used for limiting the output of the neuron. Figure 25 also includes an external bias w_0 , having the effect of raising or lowering the net input to the activation function (Mas and Flores 2008).

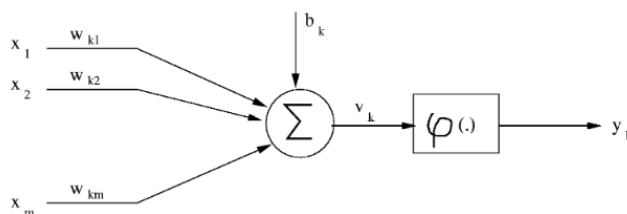


Figure 25: The anatomy of a basic neuron, used in ANNs. Figure is altered from (Mas and Flores 2008).

A.1.2 ResNet

CNNs started growing more complex, increasing the amount of hidden layers, whilst performances did not increase significantly. When CNNs grow too complex, the gradient updates become vanishingly small in the upper layers of the network, which causes significant difficulties during the training process. This phenomenon, termed the vanishing gradient problem, is addressed by (He et al. 2016). They present a new architecture which fixes this problem, the ResNet architecture. They introduce the so called ResNet block, which is also shown in figure 26. The most important part of the ResNet architecture is that it introduces a Skip Connection. The skip connection does not have any parameters and adds the output from the previous layer to the layer ahead. The skip connection passes information directly from the first layer of the block to the last. The intermediate layers then learn a residual from the input layer (King et al. 2018). This results in the ability to train much deeper networks than what was previously possible. This research caused for revolution in deep learning, it is later used as a base building block for many architectures. The ResNet building block is also used in the architecture used in this study.

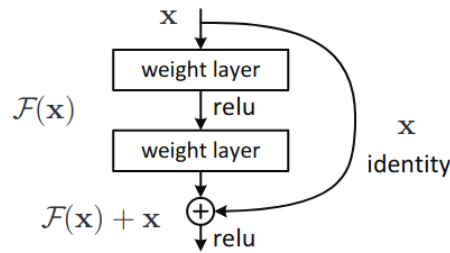


Figure 26: A residual learning building block (He et al. 2016)

A.2 Evaluation

Table 12: For clarity, again the standard binary confusion matrix layout. The rows represent actual classification, the columns represent the predicted classification.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

A.2.1 Precision

Precision shows which proportion of the predicted positives is truly positive. In an equation, precision can be described as

$$Precision = \frac{TP}{(TP+FP)}$$

True positives + False Positives is the sum of the first row.

A.2.2 Recall

Recall shows the proportion of the actual positives which are correctly classified. In an equation recall can be described as

$$Recall = \frac{TP}{(TP+FN)}$$

True Positives and False Negatives is the sum of the first column.

B Methods

B.1 Binary Curaçao

Since the binary approach worked well for Moorea, there was decided to do the same approach for Curaçao. The only differences would be the quality of data and the size of the data. Moorea tiles are 264x264, Curaçao tiles are 512x512, as resolution is not high enough to use smaller tiles. There are more Curaçao tiles compared to Moorea tiles, therefore, the amount of data should not be a problem. Unfortunately, therefore, the models could not be combined, as a combination of both models need the same input size. Around 40 images were manually annotated and were augmented

into 235 tiles, which was the highest amount used in this research. A new, slightly altered script was made. The learner was trained for 15 epochs, with a weight decay of 0.1 and a learning rate of 1×10^{-4} .

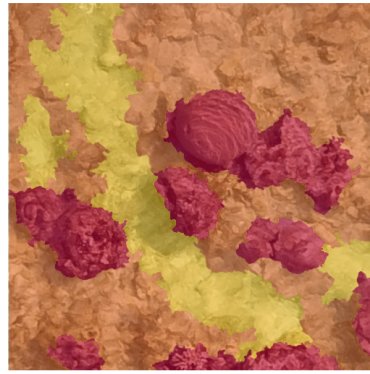
B.2 Multi class classification

B.2.1 Five classes

The images were labelled in 6 classes, which can be seen in table 10. There were no reference points provided, so the classes were classified based on human detection. Therefore, they were classified based on prior knowledge and examples. Corals were mainly chosen by their distinct features. They are circular and have a distinct orange-brown color. Rocks are rocks attached to the land. Rubble are smaller, fragmented rocks, which can also be seen in (shallow) water, they can have a dark brown to light brown colour. Sometimes, fragmented parts are seen as sphere-like features. Shallow water/sand is quite a large classes, which differs from clear, light brown, sand in very shallow areas to light blue water in deeper areas. A part has been classified as shallow water/sand when rubble parts were not distinctly seen. Waves is defined by the white water patches on some of the images. This classification has been chosen by clear features, but also by their relationship to depth. These classes can make it easier to link to a potential depth map.



(a) Example of a tile, in shallow water with clear features



(b) Tile with manually generated mask, labels are described in table 13

Figure 27

Table 13: First chosen classes in the segmentation.

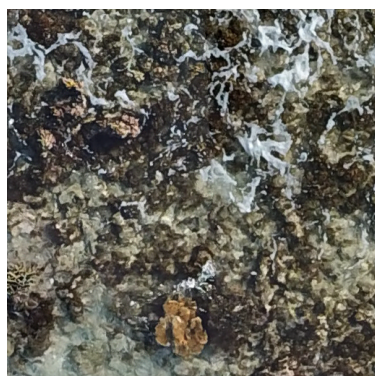
Label	Colour on mask	Class
0	Black	No data
1	Red	Corals
2	Yellow	Shallow water/sand
3	Orange	Rubble
4	Green	Rocks
5	Blue	Waves

B.2.2 Eight classes

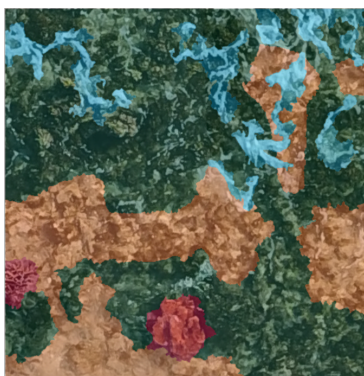
As there were difficulties for the model to make a distinction between specific classes, we decided to split up some of the classes. This meant for example splitting up shallow water/sand into shallow water. The addition of the classes dark rubble and deep water. In hindsight, dark rubble seems to be algae patches on top of rocks or corals. Deep water is where the color of the water seems blue and no distinct features can be seen. The 8 classes model was based on the 5 classes model

Table 14: The 8 classes chosen for the multi class approach

Label	Colour on mask	Class
0	Black	No data
1	Red	Corals
2	Yellow	Shallow water
3	Orange	Rubble
4	Green	Sand
5	Light blue	Waves
6	Dark blue	Deep water
7	Dark green	Dark rubble



(a) Tile 121, slightly distorted with waves



(b) Tile with manually generated mask, labels are described in table 14

Figure 28

C Results

C.1 Binary Curaçao

Firstly, there was trained of the whole Curaçao data set. Frankly, the results of the learner look remarkable. With an accuracy of 96 percent and an intersection over union of 79 percent. The difference between training loss and validation loss is also relatively little, meaning no over or underfitting.

However, when the learner is applied to the test set, the result does not look correct. The machine performs well in classifying the brain corals, those seem to be the most distinct features.

Table 15: Statistics of the Curaçao learner when trained on 235 augmented tiles

Training loss	Validation loss	Accuracy	IoU
0.07	0.1	0.96	0.79

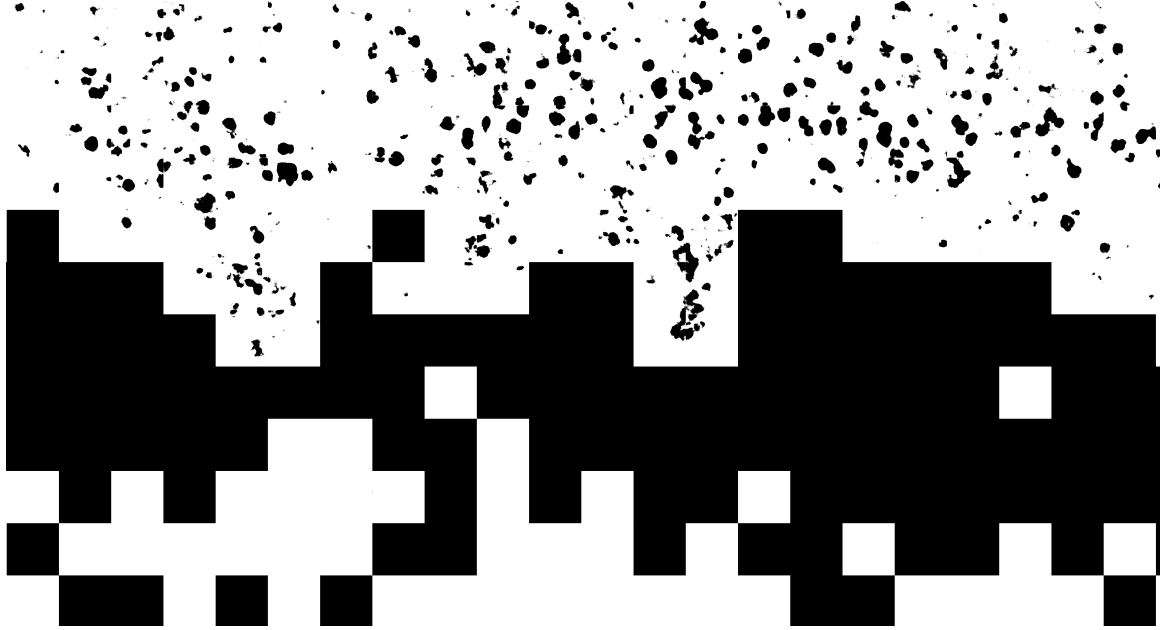


Figure 29: A cropped out part of the final classified Curaçao data set when the learner is applied to it. Black would indicate coral, whilst white would indicate non coral.

Besides, it does not perform well in the deeper parts. Tiles have been classified as either coral or non coral, when features are not clearly defined as clearly can be seen in figure 29. Because of this result, there was experimented with splitting up the deep and the shallow part of the bay. This is what is explain in the main section of the report.

C.2 Multi class classification

C.2.1 Five classes

For the 5 classes algorithm, weight decay was used as a regularization method, which was set on 0.01, the default setting. Learning rate was set on 1×10^{-4} . The model was trained for a total of 25 epochs. An epoch is one cycle through the full data set. So for this case, one cycle going through the 130 augmented images. Around 22 epochs, the model started to converge but it kept increasing. Eventually, it reached an IoU above 50% and an accuracy of 82 %. There is a difference seen in validation loss and training loss after 21 epochs, but this difference is not significant.

Table 16: Final statistics of the 5 classes algorithm when it has finished training

Training loss	Validation loss	Accuracy	IoU
0.21	0.48	0.84	0.54

Table 17: Statistics of the performance of the validation set when using the algorithm.

	Precision	Recall	IoU
0	1	1	0.99
1	0.15	0.16	0.09
2	0.25	0.20	0.12
3	0.46	0.51	0.32
5	0.12	0.11	0.06

C.2.2 Eight classes

The same amount of input images were labelled and they were also augmented into 130 input tiles. Weight decay was still used as the regularization method, with default settings of 0.01. Learning rate was also set on 1×10^{-4} . The algorithm again was trained for a total of 25 epochs. The model started to fluctuate after 21 epochs, increasing and decreasing in accuracy and IoU over the last 4 epochs. It stopped training at an accuracy of 70% and an IoU below 50%. After 20 epochs, a significant difference between training and validation loss is observed. Validation loss stays higher compared to the training loss.

Table 18: Final statistics of the 8 classes algorithm when it has finished training.

Training loss	Validation loss	Accuracy	IoU
0.54	0.98	0.7	0.39

Table 19: Statistics of the performance of the validation set when using the algorithm.

	Precision	Recall	IoU
0	1.00	0.98	0.98
1	0.16	0.23	0.11
2	0.30	0.12	0.09
3	0.26	0.16	0.11
4	0	0	0
5	0.05	0.34	0.05
6	0.05	0.02	0.01
7	0.25	0.26	0.14

For both the 5 classes as the 8 classes model, we could deduce that both models did not provide sufficient results. The IoU of the learners from both algorithms is quite low, the one of the 8 classes is even lower than the overall acceptable value. When looking at tables 17 and 19, the precision, recall and IoU values are too low to be characterized as a good classification.