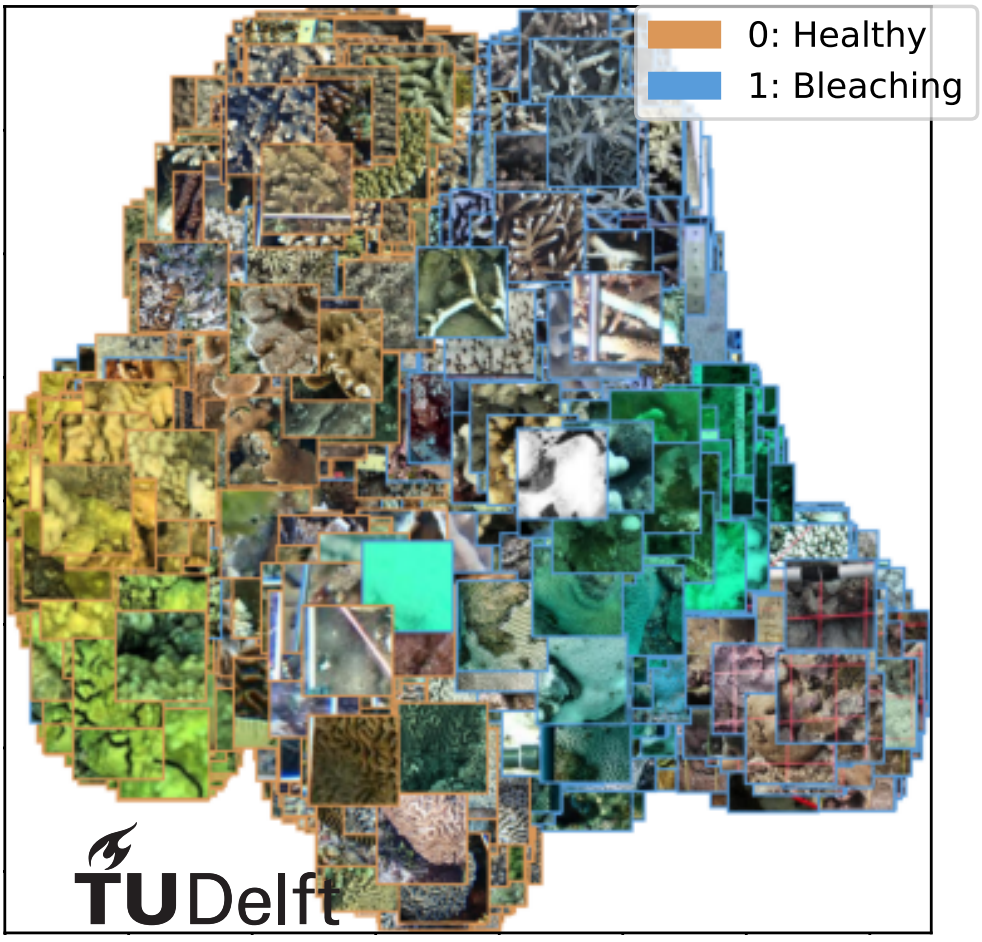
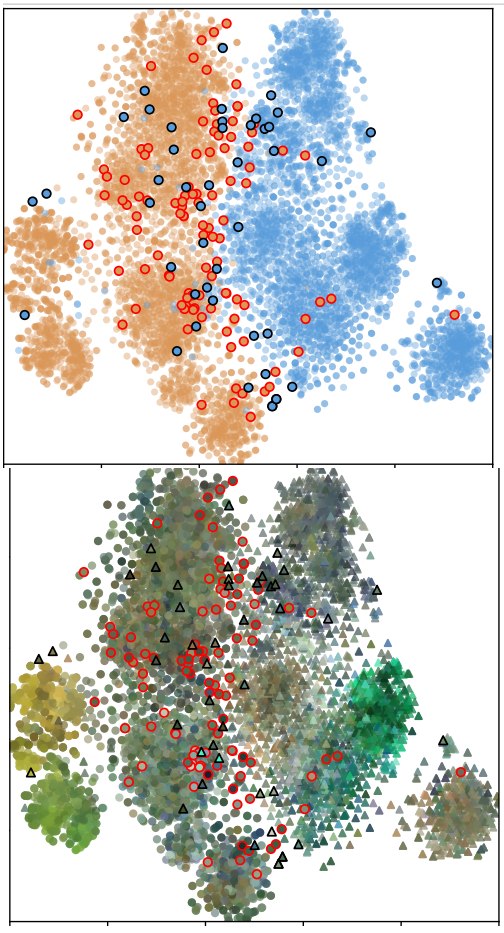
MSc Thesis

Analysing visual biases in coral imagery for bleaching detection

Jimmy J. C. Vlekke





Analysing visual biases in coral imagery for bleaching detection

by

Jimmy Johannes Cornelis Vlekke

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday December 15, 2022 at 01:45 PM.

Student number:	4563379	
Project duration:	March 24, 2022 – December 15, 2022	
Thesis committee:	Prof. dr. ir. M. J. T. Reinders,	TU Delft, Supervisor and Chair
	Dr. S. L. Pintea,	LUMC, Supervisor and committee member
	Dr. L. C. Siebert,	TU Delft, Committee member
	Ir. M. Bittner,	TU Delft, Committee member

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This report presents the work of my master's thesis project on the topic *Analysing visual biases in coral imagery for bleaching detection*. The text is structured in two parts, a scientific paper presenting our work in a compact manner using the format as in the Computer Vision and Pattern Recognition Conference (CVPR) and the appendix that introduces some of the topics of the thesis and elaborates upon the scientific paper.

This research was conducted at the Pattern Recognition and Bioinformatics Group at the Delft University of Technology under the supervision of Prof. dr. ir Marcel J.T. Reiders and Dr. Silvia Pintea.

I would first like to thank my daily supervisor Silvia. Her guidance and feedback during the weekly meetings was very valuable and she has helped shape the work presented in this thesis. I would also like to thank Marcel for his valuable feedback during our bi-weekly meetings as the head of the Pattern Recognition and Bioinformatics group at the Delft University of Technology. I would also like to express my appreciation to Dr. Luciano Cavalcante Siebert and PHD Candidate Marian Bittner for taking an interest in my work and agreeing to be a part of the evaluation committee. Additionaly, I would also like to thank the co-founders of Reef Support, Marcel Kempers and Yohan Runhaar, for their support, knowledge sharing and kindness. Finally, the care I got from the people close to me did not go unnoticed, I greatly appreciate their support.

Jimmy J. C. Vlekke
Delft, December 2022

Contents

1 Scientific Paper 1

2 Appendix 17
2.1 Introduction . . . 17
2.1.1 Corals . . . 17
2.1.2 Coral bleaching . . . 17
2.1.3 Coral reef monitoring . . . 18
2.1.4 Benthic survey imagery . . . 19
2.1.5 Automating coral reef monitoring . . . 19
2.1.6 Dataset biases . . . 20
2.2 Related Work . . . 20
2.2.1 Preprocessing of underwater imagery . . . 20
2.2.2 Classification problems on coral images using transfer learning . . . 20
2.3 Bias analysis for coral bleaching detection . . . 21
2.3.1 Bias detection methods . . . 21
2.4 Coral bleaching dataset analysis . . . 22
2.4.1 CoralNet Class Balanced Bleaching (CCBB) dataset . . . 22
2.4.2 Experimental setup . . . 23
2.4.3 Initial analysis for bleaching detection . . . 24
2.4.4 Manual bias analysis for bleaching detection . . . 25
2.4.5 Automatic bias analysis for bleaching detection . . . 26
2.5 Discussion . . . 28

1

Scientific Paper

Analysing visual biases in coral imagery for bleaching detection

Jimmy J.C. Vlekke
Delft University of Technology
Delft, Netherlands
jvlekke@tudelft.nl

Marcel Reinders
Delft University of Technology
Delft, Netherlands
marcel.reinders@tudelft.nl

Silvia L. Pintea
LKEB - LUMC
Leiden, Netherlands
s.l.pintea@lumc.nl

Abstract

Global warming causes coral bleaching which threatens the health and existence of coral reefs and therefore also the future of a lot of species, including human beings. Efforts to automate coral reef monitoring using annotated coral images to detect coral bleaching are hindered by the lack of a complete dataset that specifies the health and bleaching status of corals. We propose to combine publicly available data into a dataset and train a CNN for coral bleaching detection. This model performs surprisingly well. However, combining data from different sources gives rise to dataset biases which helps classifiers perform better and make them unreliable for unseen data. We try to detect such biases and document themsing several bias detection methods.

1. Introduction

Coral reefs are important ecosystems on which a lot of species rely, such as human beings for food, coastal protection, income and new medicine. Yet, coral reefs are increasingly threatened by external factors such as global warming which causes coral bleaching. Coral bleaching is the process caused by stress in which corals turn white when the aglae living inside the coral's tissue are expelled, making the coral more subject to mortality. Since massive coral bleaching events have occurred with increasingly frequency and intensity since the late 1970s [9, 19, 20], it has become an important ecological problem. This worries marine biologists alike. Coral reef managers work hard to observe, conserve, protect and manage coral reefs but they are facing challenges monitoring coral reefs as this is a time-consuming and expensive endeavour. Coral conservationists use benthic survey imagery to monitor coral health by, among other things, coral classification for benthic cover and biodiversity estimation, and coral bleaching detection to gain insights into the percentage of bleached corals throughout coral reefs. To assist coral reef managers by speeding up this process and making it more cost-effective, scientists are exploring methods to automate these tasks. For example, supervised learning methods in computer vision have been applied to labelled benthic survey images [5] for coral classification. However, this is not a trivial task as there is a scarcity of (consistently) labelled data and the data is difficult to work with [17], due to the complexity of underwater images and within class variations . Therefore there is a need for large datasets of labelled benthic survey images.

The annotation process is very time-consuming because taxonomists have to manually annotate between 50 to 200 random points in an image which is a tediously repetitive task and requires well-trained taxonomists, as coral reefs are home to thousands of species that can exhibit great intra-class variations [3]. For some groups of benthic organisms, identification from visual imagery is difficult even when done by experts [4, 35]. Additionally, creators of datasets can basically choose their own interpretation of a label, and so the use may not be consistent across sources. Furthermore, long-term datasets are often scattered or spatially constrained and the field data is far from standardised [17] which results in big inter-dataset variations (e.g. image quality, viewpoint and lighting). Figure 1 shows two example patches with different labels that come from different sources that resulted in apparent variations. Imperfections and artefacts in benthic survey images (such as blurring, colour change and nekton scattering effects) [10, 37] may not only be due to inter-dataset variations but also due to intra-dataset variations. These problems do not only result in a bottleneck in the flow of information from monitoring programs to managers, which delays conservation decisions [16, 17], but also makes the application of computer vision techniques for this data more challenging. Thus, there is a need for a globally defined standardised protocol for benthic surveys, a benchmark dataset that can be used by researchers to report the performance of their models and modern deep learning models to tackle the classification tasks.

There are several coral reef imagery datasets publicly available. However, only a small selection of these datasets include labels such as 'bleached' and 'un-

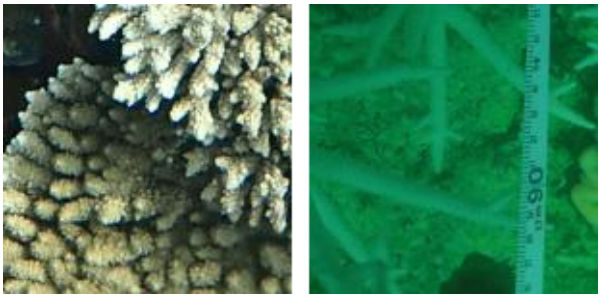


Figure 1. Example of non-bleached vs bleached coral image patches (Source: CoralNet [3]). The image on the left is a healthy Acropora coral and the image on the right is a bleached Montipora coral from a different source. These images illustrate the difficulties with the data as the image on the right has (i) a measuring tape covering part of the image, (ii) a dominant colour mask and (iii) more blur and less texture than the other image. These might form biases in the data.

bleached'/'healthy', which are necessary for to classify coral bleaching. Two of these are publicly available on Kaggle[1], a community Machine Learning and Data Science community where many datasets can be found. Both datasets are used in the same paper [22]. However, these datasets are not suitable for our research as they are quite small, containing only 720 images of bleached corals and 712 of healthy ones, and they hardly contain top view benthic survey images.

Only the CoralNet[2] project [2–4, 7, 53] contains enough desired labelled data. This project contains an abundant amount of underwater monitoring data from many different sources. Thus, to address the lack of datasets with coral bleaching being labelled, we put together a dataset. A selection of the CoralNet public data has been processed to get this dataset suitable for coral bleaching detection. To perform baseline performance, a modern Convolutional Neural Network (CNN) [25] is tested on this data. However, we observed biases in the data. This is not surprising, given the wide variation of sources from which this data comes.

We deploy several bias detection methods on this data to examine the viability of this dataset for coral bleaching detection using computer vision methods. Using these methods, we verify that biases are indeed present. Data bias detection and mitigation is an important prerequisite for robust and reliable automation of coral bleaching detection. Biases in data might translate to biased algorithmic outcomes when used to train machine learning algorithms (i.e. the bias becomes encoded in the model's weights) [39]. Such biases might affect the behaviour of users and reinforce biases [31]. The detection of biases in data is, therefore, paramount for the effective use of most machine learning models, and in this specific case, for further research in coral bleaching detection. Hereto we set out to answer the following research question: *How can we detect dataset biases and what biases exist in the coral bleaching dataset?*

The contributions of this paper are threefold: (i) We propose the composition of a benthic survey imagery dataset using publically available CoralNet data for coral bleaching detection; (ii) We evaluate a state of the art CNN model on this data; and (iii) We perform bias detection on this dataset and document the biases.

2. Related Work

2.1. Preprocessing of underwater imagery

Benthic survey datasets can suffer from issues such as blurring, noise, colour diminishing and light attenuation that are caused by varying conditions such as depth, water temperature, turbidity and current [11]. To minimise these artefacts and improve image quality, image enhancement and restoration techniques have been developed. These methods can be divided among three categories; model-free, model-based and data-driven [26, 52]. Where model-free methods perform simple pixel value adjustments, model-based or prior-based methods require prior assumptions to model the physical formation process of underwater images. Data-driven methods use deep learning models in combination with a lot of data to learn how to restore underwater images. Most of these methods improve the visual image quality by performing at least one of the following; edge enhancement, colour or illumination correction and image dehazing [3, 11, 12, 21, 36, 42]. Even though the methods of both the model-free and model-based category often improve the visual quality of most underwater images [26], experimental research has shown that using most model-free image enhancement algorithms to preprocess the images does not necessarily improve the accuracy of a CNN that tries to label these images [11]. To this day, there is no method that can effectively be used to enhance underwater images that come from a range of diverse sources [52].

We use a model-free method to preprocess underwater images because a model-based method would not be possible given the wide variety of sources of which we do not know the priors. Moreover, the method possibly reduces the possible effects of biases in the dataset since it normalises the data.

2.2. Classification problems on coral images

There is quite some research on classification of corals using coral images since the early 2000s [30, 32, 48]. These started out by using manually engineered image features in

[1] https://www.kaggle.com/datasets/sonainjamil/bleached-corals-detection and https://www.kaggle.com/datasets/sonainjamil/bhd-corals

[2] https://coralnet.ucsd.edu/

combination with simple classifiers to classifying a limited number of benthic substrates (e.g. sand, dead coral, living coral), the results were promising but the output is constrained to monitoring coral cover. Since then, advancements have been made in this research field, especially once more modern deep learning methods such as CNNs were trained to classify corals on various taxonomic levels (mostly on genus level) [15, 16, 28]. A CNN, for example, can learn feature maps that extract different features at different depths of the network. Low-level layers capture low-level features (i.e. corners and edges) and high-level layers capture high-level features (i.e. shape and texture) and are generally more class-specific [55]. To effectively train these complex deep learning models, a large amount of training data is necessary. Thus, transfer learning is commonly used in coral classification tasks [15, 16, 28]. This means that the picked CNN architecture is first pre-trained on a large dataset (e.g. ImageNet [8], containing millions of images and thousands of classes). Then, the CNN is fine-tuned by training the CNN on the context-specific dataset. Unlike these methods that use transfer learning, we propose to not use transfer learning, but to train the models from scratch, as biases can be transferred too [41].

Regarding the classification of a coral's health status (e.g. dead or healthy), several attempts have been made to classify corals including the 'alive' status of these corals [37, 45]. But the classification is limited to whether the coral is alive or dead and does not include any information on whether the corals are bleaching or not. Similar to these works, we also focus on the classification task of coral bleaching detection.

2.3. Bias detection in image datasets

In recent years, research efforts in deep learning have focused on explaining the decisions made by CNNs. Due to the complexity and flexibility of CNNs, CNNs are able to capture meaningful representations of images (complex high-dimensional data) for classification tasks which makes it challenging to deduce the reasoning behind the decisions of these models. To increase the explainability of CNNs, methods have been developed that try to explain the predictions by highlighting parts of the image deemed important by the model using saliency maps [55] or class activation maps [13, 43, 56]. These methods have been used to expose biases learned by a model from an image dataset. But using these methods to detect biases remains, however, a tedious endeavour because it requires a lot of manual work looking at the results to find prediction biases [50]. Moreover, the bias may not be explicitly present in the image but hidden in a latent representation of the input data [39], making it harder to use these methods for bias detection in image datasets. We do not use these methods as it is a very time consuming task to analyse the results and they are hard to interpret, especially given the context because we are no experts in the field of marine biology.

Influential prior work is limited for (semi-)automated bias detection as it is quite a new approach in the field. One such method proposed a step-by-step framework for bias detection based on heatmaps and clustering [33]. The original image and attribution map pairs are reduced in dimensionality and concatenated into one vector. Spectral clustering is then performed to find clusters that each should represent a cluster. This method successfully detected and eliminated biases that affected the model's classification performance. Similar to these works, we focus on automatic bias detection.

3. Bias analysis for coral bleaching detection

3.1. Data collection and preprocessing

Data collection. The CoralNet[3] project [3] contains an abundant amount of underwater monitoring data from many different locations, labelled by marine biology experts. However, there is no option to simply download all images from this project. Therefore, we build a web scraper to acquire a dataset. A selection of the public CoralNet data is processed to get a dataset suitable for coral bleaching detection. This entails that image patches of which the labels contain information on the health/bleaching status of the labelled coral are extracted from CoralNet. Semantic labels containing less than a thousand samples are ignored. The resulting data collection consists of 46,214 images of which 10,000 images are used to create the CoralNet Class Balanced Bleaching (CCBB) dataset that we use for our experiments in Section 4. We only use a selection of the data because we want to create a balanced dataset where every label is equally represented, where the label with the least amount of samples forms the limiting factor. These images are from 161 sources from all over the world. The oceans and seas from which these sources come are the Pacific Ocean, Indian Ocean, Red Sea, Caribbean Sea and the Persian Gulf. The semantic labels from CoralNet are merged together to obtain only two classes; 'bleaching' and 'healthy'. In the process, a preference is given to the semantic labels of a coral where the genus (or any other taxonomic level) exists in both the 'healthy' and the 'bleaching' counterparts. The CCBB dataset that we use for the experiments exists of 5,000 images of bleaching corals and 5,000 images of healthy corals, resulting in a total of 10,000 images. Each class is composed of 5 CoralNet labels of which 1,000 images are randomly picked to get a total of 5,000 images for each class. This is done to mitigate the possibility of a bias caused by a class imbalance. The exact composition of the dataset can be reviewed in Table 1.

[3] https://coralnet.ucsd.edu/

Taxonomy*	Class	Label	Label name	Sources (public)	Annotations (downloaded)**	Labels that shares sources
Acropora	Healthy	5877	ACROPORA BRANCHING HEALTHY	5 (2)	12382 (3124)	203, 5881, 5869, 5897
		5881	ACROPORA NONBRANCHING HEALTHY	5 (2)	9039 (5731)	203, 5877, 5869, 5897
	Bleaching	203	BLEACHED ACROPORA BRANCHING	41 (4)	6228 (6149)	5897, 5869, 5877, 1764, 2060, 2485, 5881
		2485	BLEACHED ACROPORA	72 (9)	14390 (14212)	1764, 203, 2060
Platygyra	Healthy	5897	PLATYGYRA HEALTHY	5 (2)	1086 (1076)	203, 5881, 5869, 5877
	Bleaching	1764	BLEACHED PLATYGYRA	46 (3)	2912 (2807)	2485, 2060, 203
Montipora	Healthy	5869	MONTIPORA ENCRUSTING HEALTHY	5 (2)	10774 (6344)	203, 5881, 5877, 5897
	Bleaching	2060	BLEACHING ENCRUSTING MONTIPORA	13 (0)	1340 (1336)	2485, 1764, 203
Porites Lobata***	Healthy	2245	PORITES LOBATA HEALTHY	2 (0)	1414 (1414)	-
	Bleaching	2217	PORITES LOBATA BLEACHED	101 (28)	4967 (4021)	-

Table 1. The CCBB dataset composition by using CoralNet labels.
*All of these corals are scleractinian (stony/hard) corals.
**Not all annotated image patches from each label were downloaded due to the nature of the CoralNet website. The CCBB dataset contains 1,000 random image patches of each label.
***Porites Lobata is a species whereas the other labels are coral genera.
The CCBB dataset is large, especially for the classification task of coral bleaching, uses commonly annotated coral genera, and is well-balanced and diverse in terms of coral genera. Despite Acropora being over-represented, it is divisible into branching and non-branching. But Porites is only represented by a specific species which makes it more specific than the others.

Image patches retrieved from a CoralNet label come from sources that use this label. Sources have images of which a certain amount of random points (pixels) are selected for annotation. To annotate these points, context information is necessary, being the surrounding pixels, which results in an image patch of 150x150 pixels. The image patches that are labelled are either annotated by humans or by a deep learning model where each source can specify their own model to be used. Image patches labelled by the CoralNet AI will only be considered annotated when the label is predicted with a confidence level higher than the source's alleviate confidence threshold. Each source's admin sets their own threshold (ranging from 0%, auto-confirming everything, to 100%, auto-confirming nothing) depending on whether they want more automation or more human confirmation. Each source is either public or private. In the case of it being private, no information is available to those that do not have access to the source other than the returned image patches leaving no metadata for such image patches.

Data preprocessing. Several pre-processing steps are applied to the data to prepare it for coral bleaching detection. The samples in the CCBB dataset come from image patches of 150x150 pixels that are extracted using random point annotations. This means that whenever a random point is selected that is closer than $150/2 = 75$ pixels to the border of the image, the image patch will be padded with black pixels. Roughly 10% of the images are cropped and resized as these images have black borders within the predefined threshold of 30% of the image being black borders. The class distribution for this 10% is very close to 50% which means that this pre-processing step should not be a possible source of bias (i.e. blur imposed by resizing the image).

Another preprocessing step that we perform on the samples in the CCBB dataset is Contrast-Limited Adaptive Histogram Equalization (CLAHE) [36]. This method reduces the problem of noise amplification that can be experienced when only performing adaptive histogram equalisation without the limitation on contrast amplification. The contrast in the images is improved by distributing the most frequent intensity values more evenly across the whole intensity value range as seen in Figure 2. To perform CLAHE, the RGB images are first converted to the HSV (Hue Saturation Value) space where only the Value (or brightness) dimension is used to perform CLAHE after which we convert it back to RGB. We use this preprocessing step to minimise the possible effects of a bias caused by the variability in the brightness of the images as these values are normalised across the whole range of values after applying CLAHE.

Bias hypothesis. When we visually explore and observe the data and run some initial bias detection experiments as described in Section 4.2, some seemingly (sub-) class-specific artefacts stand out to us. We hypothesise that these can be potential biases. The possible biases are (i) a colour bias, (ii) an intrusive objects bias and (iii) a frequency bias. The colour bias is defined by certain prominent colour masks among samples of certain classes. The intrusive objects bias is defined by samples that have objects in the forefront of the image (e.g. a grid of lines, measurement tape and tubes). The frequency bias is defined by samples that have mostly high frequencies (sharp image) or a lack thereof (blurred image).

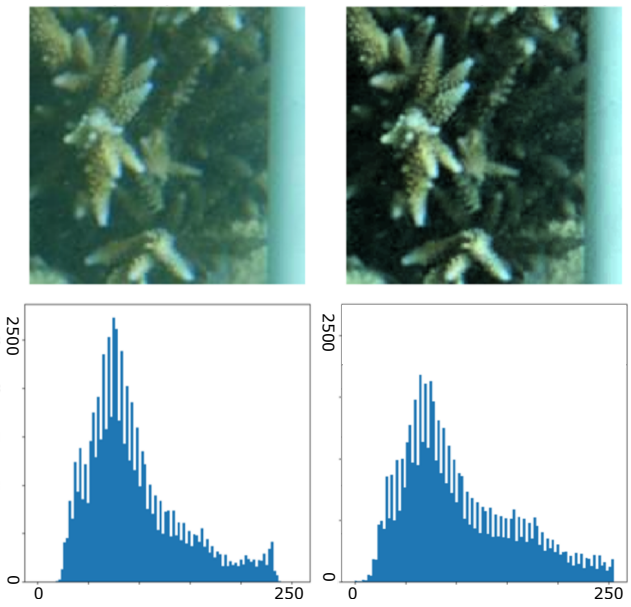


Figure 2. Contrast-Limited Adaptive Histogram Equalization (CLAHE) is applied to an image patch where the x-axis represents the intensity values of a HSV image and the y-axis represents the frequency count. The input image (first column) is preprocessed using CLAHE (second column). The resulting image shows an improved contrast that increases the amount of detail in the image. The histograms (second row) of the images show the values of the Value dimension in the HSV space. The Value distribution for the preprocessed image is bit more spread out and the maximum is smaller.

3.2. Bias detection methods

t-distributed Stochastic Neighbor Embedding (t-SNE). t-SNE [51] is a method to visualise high-dimensional data on a two or three-dimensional space using non-linear dimensionality reduction. This method is especially effective to visualise the structure of large datasets of images using feature embeddings from a CNN, making it an effective tool to analyse the relations between images fed to the network. The output, a two or three-dimensional mapping of the extracted features, displays how the network groups images based on similarities as interpreted by the network.

In our experiments in Section 4, we use this method to visualise the relation between samples using the extracted feature embeddings. We quantify the hypothesised biases with measures (see Section 3.3) to highlight samples in the t-SNE plots (e.g. average colour to paint data points). The results should help us interpret how the trained CNN uses these biases to group samples.

Learning to Split for Automatic Bias Detection (LS). LS [1] is a recently published method in the field of automatic bias detection where the source of bias is unknown during training and validation. This meta-learning method tries to maximise a generalisation gap between a training and testing split. It does so by training a Predictor on a classification task. The predictions from the converged model will be used to train a Splitter model that learns to place correctly predicted samples in the training set which cannot generalise on the test split while adhering to two constraints. These constraints are defined to avoid finding a split with (i) a shortage of training samples and (ii) a class imbalance among the splits. Thus, the method involves a bi-level optimisation problem in which the Splitter plays an adversarial game with the Predictor that is trained until convergence in every iteration of training the Splitter.

When using this method in our experiments in Section 4, the idea is to detect biases in the dataset by analysing the training and testing split found by this method. If this method is able to find splits for the CCBB dataset on which the Predictor cannot generalise well, underrepresented groups might be identified that are a cause of one or more of the hypothesised biases.

3.3. Bias quantification methods

Colour bias: quantifying colour To quantify the colour of an image, images are converted from RGB to the HSV [47] colour space to obtain the Hue value using Eq. (1 - 3). The Hue value is the colour portion of the HSV model that is able to encode information on the colour of images in just a single value. This is necessary to visualise the distributions of the colours of the images in the experiments in Section 4.3.1.

Given an RGB image $I^{\text{rgb}} = (I^{\text{r}}, I^{\text{g}}, I^{\text{b}})$, we convert it to HSV as in [47]:

$$V = \max(I^{\text{r}}, I^{\text{g}}, I^{\text{b}}) \tag{1}$$

$$S = \begin{cases} \frac{V - \min(I^{\text{r}}, I^{\text{g}}, I^{\text{b}})}{V}, & \text{if } V \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$H = \begin{cases} \frac{60(I^{\text{g}} - I^{\text{b}})}{V - \min(I^{\text{r}}, I^{\text{g}}, I^{\text{b}})}, & \text{if } V = I^{\text{r}} \\ 120 + \frac{60(I^{\text{b}} - I^{\text{r}})}{V - \min(I^{\text{r}}, I^{\text{g}}, I^{\text{b}})}, & \text{if } V = I^{\text{g}} \\ 240 + \frac{60(I^{\text{r}} - I^{\text{g}})}{V - \min(I^{\text{r}}, I^{\text{g}}, I^{\text{b}})}, & \text{if } V = I^{\text{b}} \\ 0, & \text{if } I^{\text{r}} = I^{\text{g}} = I^{\text{b}} \end{cases} \tag{3}$$

Intrusive objects bias: quantifying lines An automatic line detection method (LinE segment TRansformers (LETR)) [54] is used to identify image patches with line-shaped objects. LETR uses a backbone network that generates two feature maps that are fed to the coarse and the fine encoders. The line entities are refined by the coarse and the fine decoders respectively after which the final line segments are detected by feed-forward networks. We opt for this deep learning method because a more traditional computer vision algorithm for line detection using colour masks is unreliable due to the varying colours of objects in

the images. Straight lines in an image strongly suggest the presence of such an intrusive object, but since the method is not flawless and might find lines in reefs that are not actually from objects, a threshold of two lines is selected with a prediction confidence threshold of 0.7. The ResNet-101 architecture is used for LETR and the images are re-scaled to 256x256 pixels. These are the default settings used by the authors of LETR and work for our use case.

Frequency bias: quantifying frequencies Given an RGB image $I^{\text{rgb}} = (I^{\text{r}}, I^{\text{g}}, I^{\text{b}})$, we convert it to grayscale using the ITU-R 601-2 luma transform [44] with the weights $(w_{\text{r}}, w_{\text{g}}, w_{\text{b}}) = (0.299, 0.587, 0.114)$.

$$\text{gray}(I^{\text{rgb}}) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} w_{\text{r}} I^{\text{r}}(x,y) + w_{\text{g}} I^{\text{g}}(x,y) + w_{\text{b}} I^{\text{b}}(x,y) \quad (4)$$

To quantify the number of low and high frequencies, we use the focus measure which is obtained by taking the variance of the Laplacian of a grayscale image using Eq. (5-7). The Laplacian of the source image is calculated by adding up the second x and y derivatives which results in a convoluted image. By taking the variance of this response, you obtain a degree of focus. We compute its Laplacian and then extract the focus measure as in [38]:

$$I^{\text{gray}} = \text{gray}\left(I^{\text{rgb}}\right) \quad (5)$$

$$L = \mathcal{L}(I^{\text{gray}}) = \frac{\partial^2 I^{\text{gray}}}{\partial x^2} + \frac{\partial^2 I^{\text{gray}}}{\partial y^2} \quad (6)$$

$$\text{Focus}(L) = \frac{\sum_{x=0}^{N-1}\sum_{x=0}^{N-1}(L(x,y)-\mu)^2}{N^2} \quad (7)$$

where N is the size of the image, x and y go over the width and height of the image, and $\mathcal{L}(\cdot)$ is the image Laplacian, where μ is the mean of the L.

Another method to quantify the amount of low and high frequencies in an image is by using the discrete variant of the Fast Fourier Transform (FFT) [6] using Eq. (8-10).

We calculate the FFT of the mean grayscale image as in [6].

$$V_{\text{split, class}} = \begin{bmatrix} I_0^{\text{rgb}} \\ . \\ . \\ I_V^{\text{rgb}} \end{bmatrix} \quad (8)$$

$$\bar{I}^{\text{gray}} = \sum_{n=0}^{V}\left(\text{gray}\left(I_n^{\text{rgb}}\right)\right) \quad (9)$$

$$\text{FFT}(\bar{I}^{\text{gray}})(k,l) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \bar{I}^{\text{gray}}(x,y) e^{-i2\pi\left(\frac{kx}{N}+\frac{ly}{N}\right)} \quad (10)$$

where $V_{\text{split,class}}$ are all images of one of the two classes belonging to $\mathcal{D}_{\text{split}}$, N is the size of the image, x and y go over the width and height of the image and $\text{FFT}(\cdot)$ is the discrete Fourier transform of the average grayscale image $\bar{I}_{\text{gray}}$.

Note that only the magnitude (real numbers) of the FFT has been used to display the FFT and that the average FFT has been calculated using the average grayscale image of each split and class. Since the calculation of a grayscale image, the FFT and the average are all linear equations, it does not matter in which order these are applied.

4. Coral bleaching dataset analysis

Model architecture. The specific architecture that we use for the experiments is the Residual Network (ResNet) [18] architecture. This architecture shows a superior performance on coral classification tasks when training a model from scratch or fine-tuning a model [14, 15, 27, 29, 34, 37], is well suited for smaller datasets [49] and is deeper yet smaller than some other architectures like VVGNet [46]. The experiments that required the use of a CNN are adjusted to use the ResNet-18 model.

Model setup. For the ResNet-18 model, we use the Stochastic Gradient Descent (SGD) [40] optimiser with a learning rate of 0.001 and a momentum of 0.9. This optimiser shows superior performance compared to Adam [24] in our experiments that can be found in the Appendix. For the learning rate we use a scheduler that would decay the learning rate by 0.1 every 7 epochs with a gamma of 0.1. As criterion we used the cross-entropy loss. Each experiment runs for 20 epochs with batches of 128 samples. To cross validate the results we use a 10-fold cross validation which amounts to a 90/10 training/validation split for each fold.

Resources. The experiments are all executed on the High Performance Computing (HPC) cluster of the Delft University of Technology [4]. The jobs on this cluster use one of the following GPUs: NVIDIA Quadro K2200, NVIDIA Tesla P100, NVIDIA GeForce GTX 1080 Ti or NVIDIA GeForce RTX 2080 Ti with CUDA enabled. The average time to run 1 epoch, in a 10-fold cross validation setting as explained above, is 12.4 seconds of which training takes 10.7 seconds and validation takes 1.7 seconds.

4.1. Initial analysis for bleaching detection

To get an indication of how well the task of coral bleaching can be learned by a CNN, ResNet-18 is trained from scratch without any data augmentations. The results were surprising, as the model shows exceptional performance as seen in Table 2. The high out-of-the-box performance of coral bleaching detection of the model on the CCBB dataset

[4] https://wiki.tudelft.nl/bin/view/Research/InsyCluster/

suggests the presence of biases (as we hypothesise at the end of Section 3.1) in the dataset which makes it easier for the model to discriminate classes.

Data	Accuracy	Loss	ROC AUC score
Training	0.9380	0.1655	0.9142
Validation	0.9078	0.2274	0.8285

Table 2. Training ResNet-18 on the CCBB dataset for coral bleaching detection. The performance metrics (accuracy, loss and ROC AUC score) are obtained from averaging the values at the last epoch, of the 10 folds when the model converged, on the training and validation sets. The performance gap of the model on the training and validation data are quite small, indicating no under- or overfitting. The high out-of-the-box performance on coral bleaching detection of the model on the CCCBB dataset suggests the presence of biases in the dataset which makes it easier for the model to discriminate classes.

4.2. Bias analysis for bleaching detection

We use several manual and automatic bias detection methods to detect the possible biases in the CCBB dataset (see Section 3.2). For each experiment we use the Predictor, a trained ResNet-18 model, and the train and test split as returned by the Learning to Split (LS) method.

The result from the LS method that we use in these experiments is selected among several runs, of which the LS learning curves are depicted in Figure 3, by picking the result with the biggest generalisation gap found. The biggest generalisation gap found is 29.81%. This is supposedly the most interesting split to analyse as the biases should be separated the most between these splits among all the LS runs.

The random split, that will be referred to as $\mathcal{D}^{\text{random}}$, is a random 75/25 (train/test) dataset split with a minimal generalisation gap of 0.19%. The worst split, that will be referred to as $\mathcal{D}^{\text{worst}}$, is the split with the biggest generalisation gap (29.81%) found by running the LS method several times. The data from these splits will be referred to as $\mathcal{D}^{\text{split}}_{\text{type}}$ where *split* is either *random* or *worst* and *type* is either *train*, *test* or not specified in case it refers to all the data of the *split*. More detailed information on $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ and the corresponding Predictors can be found in Figure 3.

4.3. Bias hypothesis testing

We test the hypothesised biases (as stated in Section 3.1) for their validity. To do so, Figure 4 show a t-SNE plot that shows the data distributions of $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ when extracted by the corresponding Predictor.

4.3.1 The colour bias hypothesis

For the colour bias experiment, the images are converted from RGB to the HSV colour space to obtain the Hue value

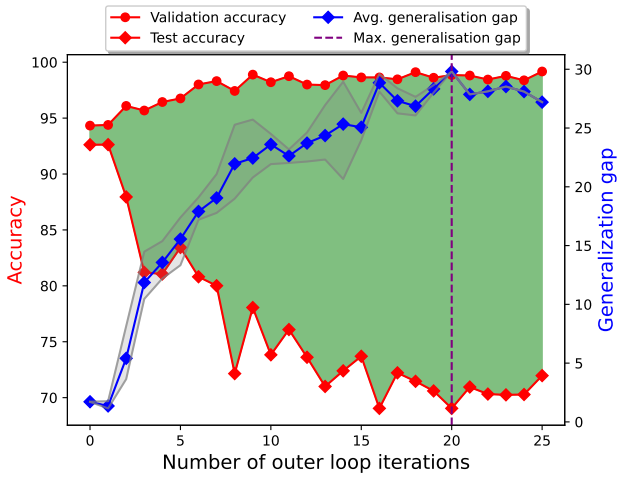


Figure 3. The average learning curve of three LS runs to obtain the maximum generalisation gap, of 29.81%, among these runs. The grey learning curves are the min and max for the runs. The generalisation gap is from $\mathcal{D}^{\text{worst}}_{\text{train}}$ (the validation accuracy) to $\mathcal{D}^{\text{worst}}_{\text{test}}$ (the test accuracy). The picked LS run, $\mathcal{D}^{\text{worst}}$, converged after 20 outer loop iterations when it found the maximum generalisation gap. The other runs converged earlier and therefore terminated earlier (at outer loop iterations 15 and 19). The LS bias detection finds a split on which the model highly overfits the training data, as indicated in the accuracy gap between training and test. We call this split the worst split.

Split	Gap	Train			Test		
		Acc*	Count	Ratio**	Acc*	Count	Ratio**
Random	0.19	93.10	74.5	$\frac{50.01}{49.99}$	92.91	25.5	$\frac{49.96}{50.04}$
Worst	**29.81**	98.85	76.4	$\frac{53.29}{46.71}$	69.04	23.6	$\frac{39.38}{60.62}$

Table 3. LS output for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ where each value is a percentage. $\mathcal{D}^{\text{worst}}$ has a significantly bigger generalisation gap than $\mathcal{D}^{\text{random}}$, thus the Predictor performs considerably less well on $\mathcal{D}^{\text{worst}}_{\text{test}}$. Even though the split sizes and ratios are relatively similar, there is a little bit of a class imbalance for $\mathcal{D}^{\text{worst}}_{\text{test}}$.
*Accuracy of the Predictor on the corresponding data.
**Class ratio of healthy vs bleaching.

using Eq. (1 - 3). This is necessary to visualise the distributions of the colours of the images in this experiment.

The t-SNE of $\mathcal{D}^{\text{random}}$ using the corresponding trained model can be observed in Figure 5 which shows no significant difference between $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$ for the distribution of the colours among samples. The model seems to cluster the samples based on colours and this happens for both the $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$. However, when comparing this to $\mathcal{D}^{\text{worst}}$ in Figure 5, it shows that the average HSV (colour) value only plays a role in the feature extraction of the model for $\mathcal{D}^{\text{worst}}_{\text{train}}$ as $\mathcal{D}^{\text{worst}}_{\text{test}}$ does not contain specific colour clusters. This strongly suggests that the colour of images does form a bias for the classes.

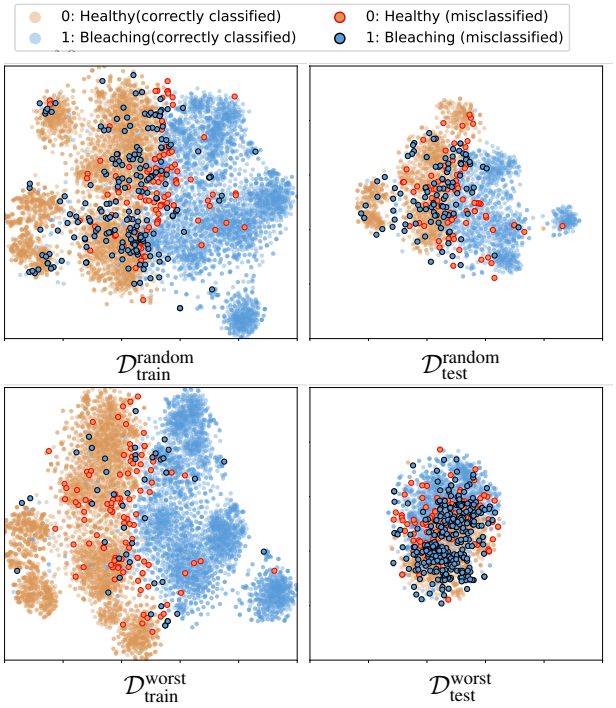


Figure 4. The t-SNE plots for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$, both using features extracted by the corresponding trained ResNet-18 models. The distribution of $\mathcal{D}^{\text{random}}_{\text{test}}$ is quite similar to $\mathcal{D}^{\text{random}}_{\text{train}}$ as the data is just randomly split. However, the distribution of $\mathcal{D}^{\text{worst}}_{\text{test}}$ is not similar to $\mathcal{D}^{\text{worst}}_{\text{train}}$. The Splitter obtained $\mathcal{D}^{\text{worst}}_{\text{train}}$ and $\mathcal{D}^{\text{worst}}_{\text{test}}$ as the Predictor correctly classified samples from $\mathcal{D}^{\text{worst}}_{\text{train}}$ and misclassified samples from $\mathcal{D}^{\text{worst}}_{\text{test}}$. This is supported by the distribution of the Predictor's extracted features as those are well divided for $\mathcal{D}^{\text{worst}}_{\text{train}}$ and more overlapping for $\mathcal{D}^{\text{worst}}_{\text{test}}$, thus it is more prone to misclassify $\mathcal{D}^{\text{worst}}_{\text{test}}$. The Predictor learned to extract features from the $\mathcal{D}^{\text{worst}}_{\text{train}}$ samples such that these discriminate well for only $\mathcal{D}^{\text{worst}}_{\text{train}}$ samples for the given classification task. Thus, this suggests that the $\mathcal{D}^{\text{worst}}_{\text{train}}$ contains biases that are less prevalent in $\mathcal{D}^{\text{worst}}_{\text{test}}$ such that the learned biases do not help with the classification of $\mathcal{D}^{\text{worst}}_{\text{test}}$ samples.

To quantify this, we use histograms to see the distributions of the hue values between and among splits, in Figure 6. The distributions of $\mathcal{D}^{\text{random}}$ are very similar, as expected for a random split as the samples are randomly distributed among the split. However, for $\mathcal{D}^{\text{worst}}$, there is a clear difference in the distribution of hue values among $\mathcal{D}^{\text{worst}}_{\text{train}}$ and $\mathcal{D}^{\text{worst}}_{\text{test}}$. This means that the hue plays an important role as a bias as the Splitter found that the Predictor would not be able to generalise well when the hue distributions would not be that similar between $\mathcal{D}^{\text{worst}}_{\text{train}}$ and $\mathcal{D}^{\text{worst}}_{\text{test}}$. These quantitative results verify the colour hypothesis.

Colour bias conclusion: *The colour forms a bias in the CCBB dataset. We show this using clusters formed by the average colour of samples in the t-SNE plots for $\mathcal{D}^{random}$ and $\mathcal{D}^{worst}$ which points to the lack of a colour bias in $\mathcal{D}^{worst}_{test}$. Moreover, a quantification of the average hue value of samples among and between splits verifies the existence of this bias in the CCBB dataset.*

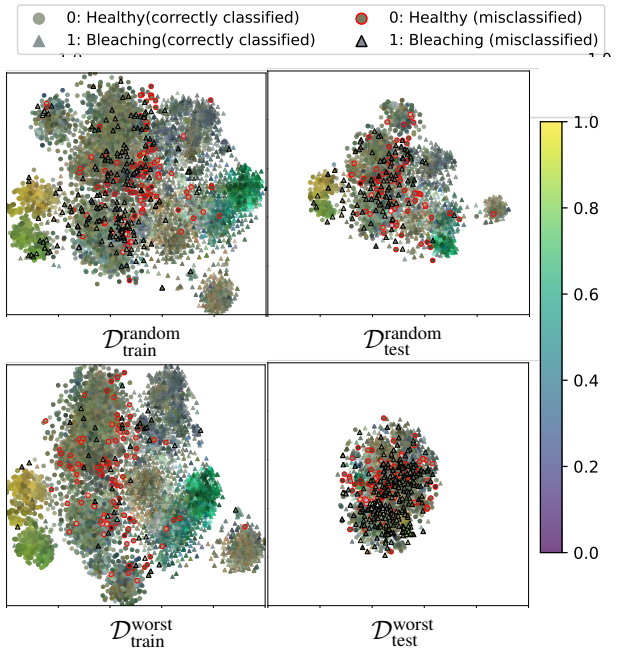


Figure 5. **Colour hypothesis.** The t-SNE plots for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ highlighting the average colour of samples, both using features extracted by the corresponding trained ResNet-18 models. The average colours of the samples seems to play an important role for models when extracting features. Samples with similar colours are clustered together, especially samples with more unusual colours (e.g. yellow, green or blue). The distributions of $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$ are very similar whereas the distribution of $\mathcal{D}^{\text{worst}}_{\text{test}}$ is very different from that of $\mathcal{D}^{\text{worst}}_{\text{train}}$, just like in Figure 4. $\mathcal{D}^{\text{worst}}_{\text{test}}$ is just one cluster of very similar colours and does not contain any cluster of a specific colour. From these t-SNE plots it appears that the biased healthy samples have greenish and yellowish colours (forming clusters in the lower left corner of the t-SNE plot) and a blueish colour for the biased bleaching samples (forming a cluster in the centre/lower right corner).

4.3.2 The intrusive objects bias hypothesis

For the intrusive objects bias there should be some kind of method to detect whether there are such objects in a sample. To achieve the detection of line-shaped objects, we use the line detection method LETR (see Section 3.3).

The t-SNE of $\mathcal{D}^{\text{random}}$ using the corresponding trained model can be observed in Figure 7 which shows no visible difference for the distribution of the amount of lines detected among samples in $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$. The model seems to form some clusters based on the amount of detected lines and this happens for both $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$.

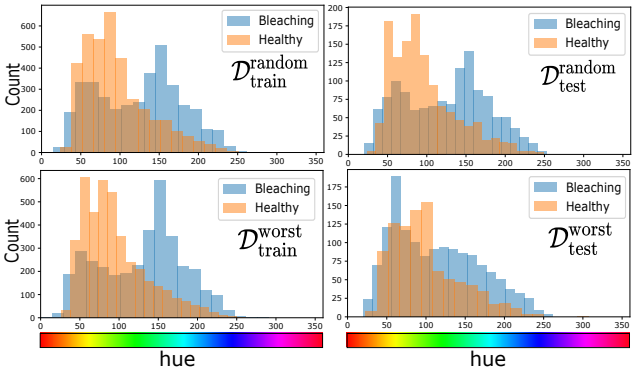


Figure 6. **Colour hypothesis.** Average hue histograms for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$. The hue distributions of $\mathcal{D}^{\text{random}}$ are very similar whereas the distributions of $\mathcal{D}^{\text{worst}}$ are very different. The hue distributions of $\mathcal{D}^{\text{worst}}_{\text{train}}$ shows two very different mode values for the two classes whereas the distributions of these two classes in $\mathcal{D}^{\text{worst}}_{\text{test}}$ are overlapping much more, especially the distribution of the hue of the bleaching class in $\mathcal{D}^{\text{worst}}_{\text{test}}$ is very different from the other splits but more similar to the distribution of the healthy class in $\mathcal{D}^{\text{worst}}_{\text{test}}$. These differences indicate that the Splitter from $\mathcal{D}^{\text{worst}}$ learned to find a split where the hue is important to fool the Predictor. As $\mathcal{D}^{\text{worst}}_{\text{train}}$ contains the hue biases and $\mathcal{D}^{\text{worst}}_{\text{test}}$ does not, the Predictor is more likely to misclassify the samples from $\mathcal{D}^{\text{worst}}_{\text{test}}$ as these do not contain the hue bias. Thus, the hue forms a bias in the dataset.

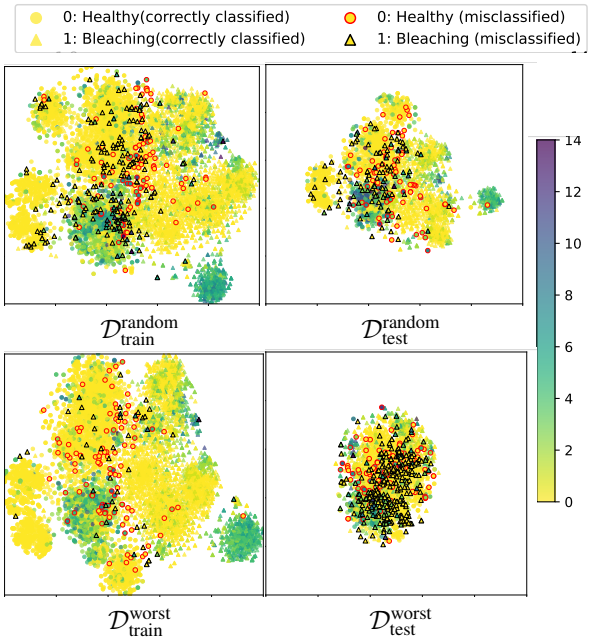


Figure 7. **Intrusive objects hypothesis.** The t-SNE plots for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ highlighting the number of detected lines for each sample, both using features extracted by the corresponding trained ResNet-18 models. The number of detected lines of the samples seems to play some role for models when extracting features. Samples with similar amount of detected lines are clustered together, especially samples with several detected lines are clustered together. The distributions of $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$ are very similar whereas the distribution of $\mathcal{D}^{\text{worst}}_{\text{test}}$ is very different from that of $\mathcal{D}^{\text{worst}}_{\text{train}}$, just like in Figure 4. The distribution of $\mathcal{D}^{\text{worst}}_{\text{test}}$ is just one cluster of very different amounts of detected lines that do not belong to a specific class based on the number of lines detected. Whereas the bleaching class in all but $\mathcal{D}^{\text{worst}}_{\text{test}}$ seems to form a cluster for the number of detected lines. Thus, from these t-SNE plots it appears that the invasive objects seems to form a bias, especially for some of the bleaching samples that have many lines detected (forming clusters in the centre/lower right corner).

However, when comparing this to distributions of $\mathcal{D}^{\text{worst}}$ in Figure 7, it shows that the number of lines detected only plays a role for formed clusters in the feature extraction of the model for $\mathcal{D}^{\text{worst}}_{\text{train}}$ as $\mathcal{D}^{\text{worst}}_{\text{test}}$ does not contain specific detected lines clusters based on classes. This strongly suggests that the amount of detected lines in images does form a bias when classifying coral bleaching images.

To quantify this effect, Table 4 shows the ratio of intrusive objects (samples where LETR detected two or more lines) and no intrusive objects for the classes in $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$. The ratio is very similar between $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$, however, this ratio is quite different for $\mathcal{D}^{\text{worst}}$. The samples of $\mathcal{D}^{\text{worst}}_{\text{train}}$ have a high ratio of non-intrusive object samples for the healthy class and a high ratio of intrusive object samples for the bleaching class. These proportions are somewhat flipped when looking at $\mathcal{D}^{\text{worst}}_{\text{test}}$. This indicates that the Splitter learned to split the data such that the model would learn the invasive objects bias in $\mathcal{D}^{\text{worst}}_{\text{train}}$ which does not hold in $\mathcal{D}^{\text{worst}}_{\text{test}}$. Thus, the Predictor is not able to classify the $\mathcal{D}^{\text{worst}}_{\text{test}}$ samples as accurately. Healthy samples have, in general, only a few samples with invasive objects (12.5%) whereas bleaching samples have more (26.5%). When looking at these ratios in $\mathcal{D}^{\text{worst}}_{\text{train}}$, they are larger, forming a stronger bias (11.1% for healthy and 29.5% for bleaching). However, these ratios are very similar for $\mathcal{D}^{\text{worst}}_{\text{test}}$ (17.3% for healthy and 16.9% for bleaching), which makes it harder for the Predictor to correctly classify these samples because of the learned bias.

Intrusive objects bias conclusion: *The invasive objects form a bias in the CCBB dataset. We show this using clusters formed by the number of detected lines for each sample in the t-SNE plots for $\mathcal{D}^{random}$ and $\mathcal{D}^{worst}$, in combination with a quantitative analysis of the amount of lines detected in splits, points to the presence of the intrusive objects bias in the dataset.*

4.3.3 The frequency bias hypothesis

Since we expect the type of camera setup used to acquire benthic survey images to bias the resulting images, we test whether there is a bias towards the amount of low and high

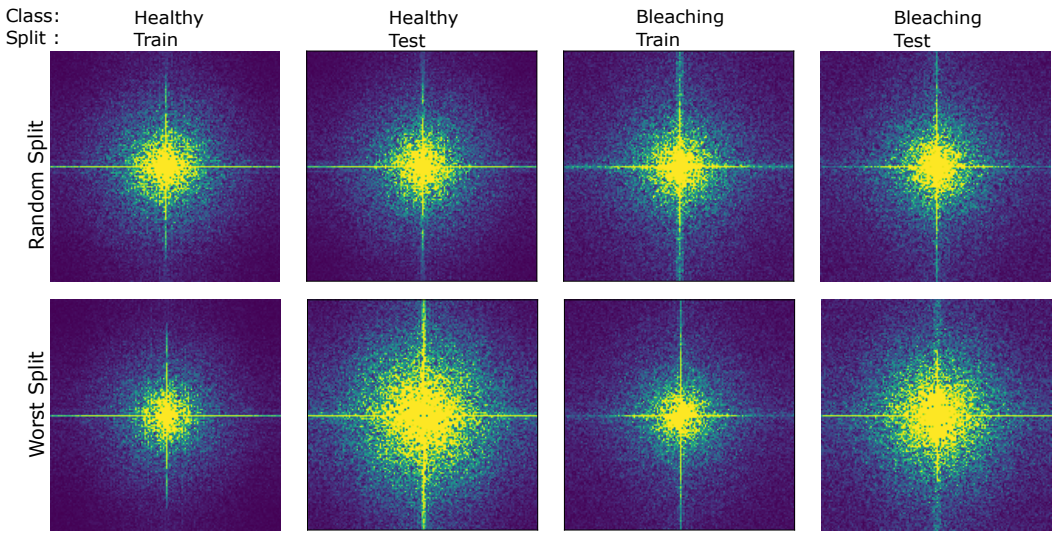


Figure 8. **Frequency hypothesis.** Average FFT for each class for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$. The difference between $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ in terms of high frequencies indicates that the Splitter learned to find a split where the amount of high frequencies is important to fool the Predictor. The Predictor performs well on $\mathcal{D}^{\text{worst}}_{\text{train}}$ which has samples that lack high-frequencies, while it performs poorly on $\mathcal{D}^{\text{worst}}_{\text{test}}$ which does have samples with more high-frequencies. Thus, the amount of high-frequencies forms a bias in the dataset.

Split	Train		Test	
	Healthy	Bleaching	Healthy	Bleaching
Random	12.2	25.0	12.6	29.4
Worst	11.1	29.4	17.3	16.9

Table 4. **Intrusive objects hypothesis.** Table comparing the ratio of detected lines (related to intrusive objects) for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$. The ratio for the classes in $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$ is similar whereas that of $\mathcal{D}^{\text{worst}}$ the ratio is inverted between $\mathcal{D}^{\text{worst}}_{\text{train}}$ and $\mathcal{D}^{\text{worst}}_{\text{test}}$. The differences between the distributions of $\mathcal{D}^{\text{random}}$ compared to that of $\mathcal{D}^{\text{worst}}$ indicates that the Splitter of $\mathcal{D}^{\text{worst}}$ learned to find a split where the number of detected lines (presence or absence of intrusive objects) is important to fool the Predictor, thus forming a bias in the dataset.

frequencies in an image as a result from different noise levels in cameras. Hereto we calculate the focus measure using Eq. (5-7). This measure is used to encode information on the frequencies in an image as a single value for the generation of the t-SNE plots that highlights the frequency bias. It can determine the amount of focus in an image, where a lower value means less focus and more blur, whereas a higher value means less blur and thus a sharper image. Blurred images generally lack high-frequencies whereas sharp images do not.

The t-SNE of $\mathcal{D}^{\text{random}}$ using the corresponding trained model can be observed in Figure 9 which shows no visible difference for the distribution of the focus measure among samples of $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$. The model seems to cluster the samples based on focus measure and this happens for both $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$. However, when comparing this to $\mathcal{D}^{\text{worst}}$ in Figure 5, it shows that the focus measure only plays a role in the feature extraction of the model for $\mathcal{D}^{\text{worst}}_{\text{train}}$ as $\mathcal{D}^{\text{worst}}_{\text{test}}$ does not contain specific focus measure clusters based on the classes. This strongly suggest that the amount of low and high frequencies in images does form a bias when classifying coral bleaching images.

To quantify the distribution of the focus measure among splits and classes, we create histograms, as seen in Figure 10. The distributions of the focus measure point to a bias in the dataset because only the focus measure distributions in $\mathcal{D}^{\text{worst}}_{\text{test}}$ of the classes are very similar. This indicates that the Splitter of $\mathcal{D}^{\text{worst}}$ learned that the focus measure forms a bias that helps the Predictor in the classification of the classes as healthy samples tend to have a lower focus measure but this does not hold for samples in $\mathcal{D}^{\text{worst}}_{\text{test}}$.

Besides using the focus measure in the t-SNE plots and the histograms, we also use the discrete variant of the Fast Fourier Transform (FFT), implemented using using Eq. (8-10), to display the average FT of each split and class for further analysis of the relation between classes, splits and frequencies. The result can be seen in Figure 8 which shows a clear difference between $\mathcal{D}^{\text{worst}}_{\text{train}}$ and $\mathcal{D}^{\text{worst}}_{\text{test}}$ compared to $\mathcal{D}^{\text{random}}$ which inhibits almost no visible difference between $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$.

Figure 11 shows the average amount of frequencies for the classes among and between splits. The differentiation between low, mid and high frequencies is defined by ranges (as circles) drawn from the centre of the FFT of each image to count the amount of frequencies within the given range. Low frequencies are within 20% of the centre of the image,

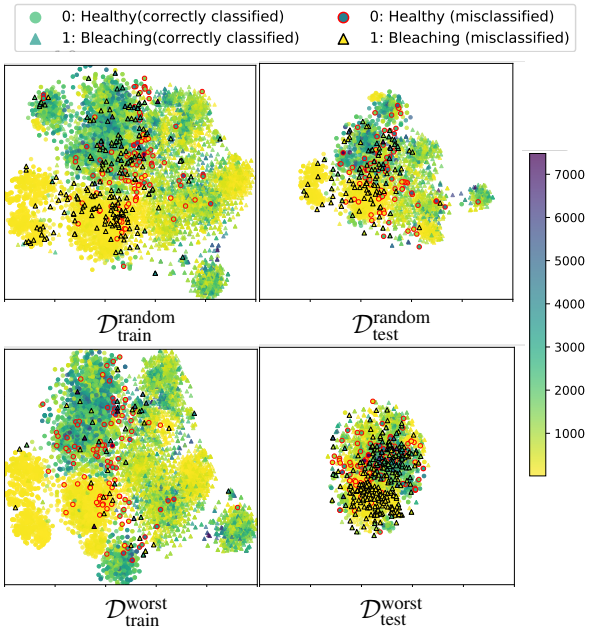


Figure 9. **Frequency hypothesis.** The t-SNE plots for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$ highlighting the focus measure for each sample, both using features extracted by the corresponding trained ResNet-18 models. The focus measure of the samples seems to play a role for models when extracting features. Samples with similar focus measures are clustered together, especially samples with very low or high focus measures are clustered together. The distributions of $\mathcal{D}^{\text{random}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}_{\text{test}}$ are very similar whereas the distribution of $\mathcal{D}^{\text{worst}}_{\text{test}}$ is very different from that of $\mathcal{D}^{\text{worst}}_{\text{train}}$, just like in Figure 4. The distribution of $\mathcal{D}^{\text{worst}}_{\text{test}}$ is just one cluster of very different focus measures that do not belong to a specific class based on the focus measure. From these t-SNE plots it appears that the biased healthy samples have either very low focus measures (forming clusters in the lower left corner of the t-SNE plots) or very high focus measures (forming clusters in the upper left corner) and an average focus measure for the biased bleaching samples (forming a cluster in the centre/lower right corner).

mid frequencies are within 30% of the centre of the image surrounding the low frequencies and high frequencies are the remainder 50% of the image. Also with this quantitative analysis, a bias is observed in the dataset as exposed by $\mathcal{D}^{\text{worst}}_{\text{test}}$ when comparing it with $\mathcal{D}^{\text{worst}}_{\text{train}}$ and $\mathcal{D}^{\text{random}}$.

Frequency bias conclusion: *The high-frequencies form a bias in the CCBB dataset. We show this using clusters formed by the focus measure of samples in the t-SNE plots for $\mathcal{D}^{random}$ and $\mathcal{D}^{worst}$, in combination with a quantitative analysis of the focus measure and the FFT of the splits, points to the presence of the low- and high-frequencies bias in the dataset.*

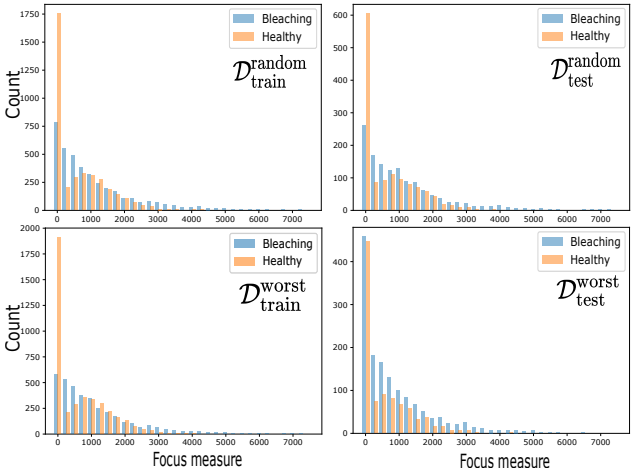


Figure 10. **Frequency hypothesis.** Focus measure histograms for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$. The focus measure distributions of $\mathcal{D}^{\text{random}}$ are very similar whereas the distributions of $\mathcal{D}^{\text{worst}}$ are quite different. The focus measure distributions of $\mathcal{D}^{\text{worst}}_{\text{train}}$ shows two different mode values for the two classes whereas the distributions of these two classes in $\mathcal{D}^{\text{worst}}_{\text{test}}$ are much more overlapping. But more obvious is the difference in the amount of low focus measures for the classes in every split but $\mathcal{D}^{\text{worst}}_{\text{test}}$. These differences indicate that the Splitter from $\mathcal{D}^{\text{worst}}$ learned to find a split where the focus measure is important to fool the Predictor. As $\mathcal{D}^{\text{worst}}_{\text{train}}$ contains the focus measure bias and $\mathcal{D}^{\text{worst}}_{\text{test}}$ does not, the Predictor is more likely to misclassify the samples from $\mathcal{D}^{\text{worst}}_{\text{test}}$ as these do not contain this bias. Thus, the focus measure forms a bias in the dataset.

5. Discussion

Bleaching and non-bleaching can accurately be classified when training a CNN on the CCBB dataset, but we suspect that data biases ease this task. We find that the hypothesised colour, intrusive objects and frequency biases are present in the CCBB dataset. However, this does not mean that the biases in the CCBB dataset are limited to these three biases.

The biases are a problem with the dataset but this is not exclusive to this dataset. Anyone that collects a coral dataset from different sources will encounter biases. The coral images are highly biased because of the differences between data acquisition methods used by the sources. They use different cameras, images are taken under inconstant and varying conditions, and some images will contain measuring tools (measuring tape, grid, colour card, etc). This is, however, also possible for other datasets that are acquired from combining data from different sources. Thus, when using any combined dataset that will be used to train a model, it is important to pay attention to possible dataset biases.

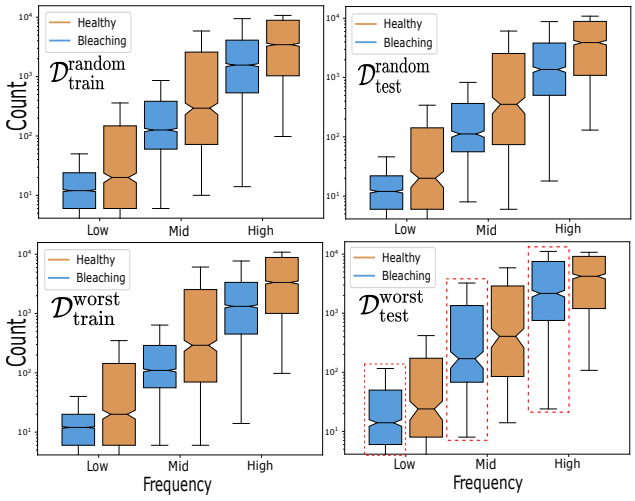


Figure 11. **Frequency hypothesis.** FFT frequencies boxplot for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$. The distributions of low, mid and high frequencies, defined by the average amount of frequencies found in the FFT of each image, are quite similar for $\mathcal{D}^{\text{random}}$. However, the distributions of $\mathcal{D}^{\text{worst}}$ are quite different. The amount of frequencies for the healthy class is very constant throughout each split but this does not hold for the bleaching class in $\mathcal{D}^{\text{worst}}_{\text{test}}$ (highlighted in red). These differences indicate that the Splitter from $\mathcal{D}^{\text{worst}}$ learned to find a split where the amount of frequencies is important to fool the Predictor. As $\mathcal{D}^{\text{worst}}_{\text{train}}$ contains the frequency bias and $\mathcal{D}^{\text{worst}}_{\text{test}}$ does not, or only to a lesser extend, the Predictor is more likely to misclassify the samples from $\mathcal{D}^{\text{worst}}_{\text{test}}$ as these do not contain this bias. Thus, the amount of frequency of images forms a bias in the dataset.

5.1. Limitations

A first limitation relates to the possible effect of the number of sources from which CoralNet labels come from. Take the label 'Porites Lobata healthy', for example, this label is only used in two sources which is a very low source count compared to most other labels (the labels used in this dataset on average come from 21.7 sources). This has an impact on the variability (which comes from the sources) of the image patches. Moreover, there exists a source imbalance for the classes 'healthy' and 'bleaching' of which the first has an average of 39.8 sources whereas the latter has an average of 3.6 sources. This could mean that a model trained on this dataset will generalise better on images of healthy corals as the model has seen more varied data of this class.

A second limitation is that the sources of image patches are mostly untraceable. As CoralNet is a hub for benthic survey projects (read sources) and each has a set of labels to which labelled image patches belong. Most sources are private, which means it is not possible to trace the source of image patches unless you have access to the private sources or the sources are public. Using a combination of sources, however, most likely introduces biases as each source uses different setups to acquire the data. This can, however, not be verified without the traceability of the sources of image patches.

A third limitation comes from the lack of information from image patches that come from private sources. All image patches of any CoralNet label come from sources that can define a confidence threshold to the annotation process of the prediction models linked to that source. For the private sources it is not possible to know this confidence threshold which could imply that the proposed dataset in this paper contains misclassified image patches. To counter the noise in the labelled data, the information from the public sources could be used as the confidence threshold for these labelled images are known. Whether the confidence threshold is often set lower than a 100% is questionable though, because only 1 out of the 40 public sources that we use in our dataset has a threshold lower than the default 100% (85% to be precise).

The limitations due to private sources can not be solved by only using public sources as this would result in a dataset with too few samples. Requesting access to all the relevant sources could solve these issues.

5.2. Future work

Future research could attempt to further research the robustness and reliability of coral bleaching detection using the proposed dataset provided in this work in combination with research to mitigate the found biases.

Continuing this line of work would be more encouraging if the CCBB dataset could actually be published, however, this is not possible due to copyright regulations.

Since this study is limited to only use image patches of CoralNet labels that explicitly mention the coral's health or bleaching status, the number of samples is drastically reduced. However, there are a lot more CoralNet labels that do not contain such information. Having this data available for coral bleaching detection would immensely increase the sample size.

A machine learning approach to be able to use these samples is Multiple Instance Learning (MIL) [23]. This is a method that assumes that there are bags of samples of which you do not know the individual label except for at least one sample in the bag or all labels are known in the bag. In the specific case of coral bleaching detection using the dataset from this paper, there are M healthy labels, N bleaching labels (with $N < M$) and K unknown samples (could be either class). Instead of discarding all K samples, bags can be formed to contain unknown samples with at least one bleaching sample and bags with only healthy labels. The power of MIL in this case is that it allows us to use samples of which we do not know the labels and therefore increase the amount of samples available. Something that should not be overlooked is that CoralNet labels without the additional

status information might include corals that are dead, which could be solved by adding another class.

6. Conclusion

We collect and assemble a dataset with healthy and bleached images and perform coral bleaching detection by training a CNN. The model accurately learns to classify coral bleaching. But we find, using bias detection methods, that this dataset contains biases. These biases have manifested themselves in the shape of undesired model predictions, the misclassification of bleached and healthy corals in this case. These biases have been introduced due to, among other things, the wide variety of sources where the images come from, the complexity of underwater imagery and the complicated nature of corals. It is important to identify dataset biases and to design methods to mitigate them because anyone that will collect a dataset is bound to encounter dataset bias problems. Particularly when collecting a dataset from a wide variety of sources that do not use the same protocols and measuring devices to acquire the data.

The code for the experiments described in this work and to reproduce the results and figures can be found at `https://github.com/theangryhobbit/coral-bleaching-detection`.

References

[1] Yujia Bao and Regina Barzilay. Learning to split for automatic bias detection. *NeurIPS*, 2022. 6

[2] Oscar Beijbom. *Automated annotation of coral reef survey images*. University of California, San Diego, 2015. 3

[3] Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1170–1177. IEEE, 2012. 2, 3, 4

[4] Oscar Beijbom, Peter J Edmunds, Chris Roelfsema, Jennifer Smith, David I Kline, Benjamin P Neal, Matthew J Dunlap, Vincent Moriarty, Tung-Yung Fan, Chih-Jui Tan, et al. Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation. *PloS one*, 10(7):e0130312, 2015. 2, 3

[5] Elena Bollati, Cecilia D'Angelo, David I Kline, B Greg Mitchell, and Jörg Wiedenmann. Development of a multi-excitation fluorescence (mef) imaging method to improve the information content of benthic coral reef surveys. *Coral Reefs*, 40(6):1831–1847, 2021. 2

[6] E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 4(12):63–70, 1967. 7

[7] Qimin Chen, Oscar Beijbom, Stephen Chan, Jessica Bouwmeester, and David Kriegman. A new deep learning engine for coralnet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3702, 2021. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4

[9] C Mark Eakin, Hugh PA Sweatman, and Russel E Brainard. The 2014–2017 global-scale coral bleaching event: insights and impacts. *Coral Reefs*, 38(4):539–545, 2019. 2

[10] Mohamed Elawady. Sparse coral classification using deep convolutional neural networks. *arXiv preprint arXiv:1511.09067*, 2015. 2

[11] Redouane Es-sadaoui, Imad El Bouazzaoui, Lahoucine Azergui, and Jamal Khallaayoune. Underwater image processing: Technical study and experiments. *vol*, 2:20–29, 2017. 3

[12] Kristofor B Gibson. Preliminary results in using a joint contrast enhancement and turbulence mitigation method for underwater optical imaging. In *OCEANS 2015-MTS/IEEE Washington*, pages 1–5. IEEE, 2015. 3

[13] Loris Giulivi, Mark James Carman, and Giacomo Boracchi. Perception visualization: Seeing through the eyes of a dnn. *arXiv preprint arXiv:2204.09920*, 2022. 4

[14] Anabel Gómez-Ríos, Siham Tabik, Julián Luengo, ASM Shihavuddin, and Francisco Herrera. Coral species identification with texture or structure images using a two-level classifier based on convolutional neural networks. *Knowledge-Based Systems*, 184:104891, 2019. 7

[15] Anabel Gómez-Ríos, Siham Tabik, Julián Luengo, ASM Shihavuddin, Bartosz Krawczyk, and Francisco Herrera. Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation. *Expert Systems with Applications*, 118:315–328, 2019. 4, 7

[16] Manuel Gonzalez-Rivero, Oscar Beijbom, Alberto Rodriguez-Ramirez, Dominic EP Bryant, Anjani Ganase, Yeray Gonzalez-Marrero, Ana Herrera-Reveles, Emma V Kennedy, Catherine JS Kim, Sebastian Lopez-Marcano, et al. Monitoring of coral reefs using artificial intelligence: a feasible and cost-effective approach. *Remote Sensing*, 12(3):489, 2020. 2, 4

[17] Manuel González-Rivero, Oscar Beijbom, Alberto Rodriguez-Ramirez, Tadzio Holtrop, Yeray González-Marrero, Anjani Ganase, Chris Roelfsema, Stuart Phinn, and Ove Hoegh-Guldberg. Scaling up ecological measurements of coral reefs using semi-automated field image collection and analysis. *Remote Sensing*, 8(1):30, 2016. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[19] OVE HOEGH-GULDBERG. Coral reefs in a century of rapid environmental change. *Symbiosis*, 2004. 2

[20] Terry P Hughes, Kristen D Anderson, Sean R Connolly, Scott F Heron, James T Kerry, Janice M Lough, Andrew H Baird, Julia K Baum, Michael L Berumen, Tom C Bridge, et al. Spatial and temporal patterns of mass bleaching of corals in the anthropocene. *Science*, 359(6371):80–83, 2018. 2

[21] Kashif Iqbal, Rosalina Abdul Salam, Azam Osman, and Abdullah Zawawi Talib. Underwater image enhancement using an integrated colour model. *IAENG International Journal of computer science*, 34(2), 2007. 3

[22] Sonain Jamil, MuhibUr Rahman, and Amir Haider. Bag of features (bof) based deep learning framework for bleached corals detection. *Big Data and Cognitive Computing*, 5(4):53, 2021. 3

[23] James Keeler, David Rumelhart, and Wee Leow. Integrated segmentation and recognition of hand-printed numerals. *Advances in neural information processing systems*, 3, 1990. 13

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[25] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 3

[26] Risheng Liu, Xin Fan, Ming Zhu, Minjun Hou, and Zhongxuan Luo. Real-world underwater enhancement: Challenges, benchmarks, and solutions under natural light. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4861–4875, 2020. 3

[27] Alessandra Lumini, Loris Nanni, and Gianluca Maguolo. Deep learning for plankton and coral classification. *Applied Computing and Informatics*, 2020. 7

[28] Ammar Mahmood, Mohammed Bennamoun, Senjian An, Ferdous Sohel, Farid Boussaid, Renae Hovey, Gary Kendrick, and Robert B Fisher. Coral classification with hybrid feature representations. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 519–523. IEEE, 2016. 4

[29] Ammar Mahmood, Mohammed Bennamoun, Senjian An, Ferdous A Sohel, Farid Boussaid, Renae Hovey, Gary A Kendrick, and Robert B Fisher. Deep image representations for coral image classification. *IEEE Journal of Oceanic Engineering*, 44(1):121–131, 2018. 7

[30] Ma Shiela Angeli C Marcos, Maricor N Soriano, and Caesar A Saloma. Classification of coral reef images from underwater video using neural networks. *Optics express*, 13(22):8766–8771, 2005. 3

[31] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021. 3

[32] Anand Mehta, Eraldo Ribeiro, Jessica Gilner, and Robert van Woesik. Coral reef texture classification using support vector machines. In *VISAPP (2)*, pages 302–310, 2007. 3

[33] Agnieszka Mikołajczyk, Michał Grochowski, and Arkadiusz Kwasigroch. Towards explainable classifiers using the counterfactual approach–global explanations for discovering bias in data. *arXiv preprint arXiv:2005.02269*, 2020. 4

[34] Md Modasshir, Alberto Quattrini Li, and Ioannis Rekleitis. Mdnet: Multi-patch dense network for coral classification. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–6. IEEE, 2018. 7

[35] R Ninio, Steven Delean, K Osborne, and H Sweatman. Estimating cover of benthic organisms from underwater video images: variability associated with multiple observers. *Marine Ecology Progress Series*, 265:107–116, 2003. 2

[36] Stephen M Pizer. Contrast-limited adaptive histogram equalization: Speed and effectiveness stephen m. pizer, r. eugene johnston, james p. ericksen, bonnie c. yankaskas, keith e. muller medical image display research group. In *Proceedings of the first conference on visualization in biomedical computing, Atlanta, Georgia*, volume 337, page 1, 1990. 3, 5

[37] Alina Raphael, Zvy Dubinsky, David Iluz, Jennifer IC Benichou, and Nathan S Netanyahu. Deep neural network recognition of shallow water corals in the gulf of eilat (aqaba). *Scientific reports*, 10(1):1–11, 2020. 2, 4, 7

[38] Muhammad Riaz, Seungjin Park, Muhammad Bilal Ahmad, Waqas Rasheed, and Jongan Park. Generalized laplacian as focus measure. In *International Conference on Computational Science*, pages 1013–1021. Springer, 2008. 7

[39] Laurent Risser, Agustin Picard, Lucas Hervier, and Jean-Michel Loubes. A survey of identification and mitigation of machine learning algorithmic biases in image analysis. *arXiv preprint arXiv:2210.04491*, 2022. 3, 4

[40] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 7

[41] Hadi Salman, Saachi Jain, Andrew Ilyas, Logan Engstrom, Eric Wong, and Aleksander Madry. When does bias transfer in transfer learning? *arXiv preprint arXiv:2207.02842*, 2022. 4

[42] Timm Schoening, Melanie Bergmann, Jörg Ontrup, James Taylor, Jennifer Dannheim, Julian Gutt, Autun Purser, and Tim W Nattkemper. Semi-automated image analysis for the assessment of megafaunal densities at the arctic deep-sea observatory hausgarten. *PloS one*, 7(6):e38179, 2012. 3

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4

[44] M Series. Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond. *Recommendation ITU*, 2083, 2015. 7

[45] ASM Shihavuddin, Nuno Gracias, Rafael Garcia, Arthur CR Gleason, and Brooke Gintert. Image-based coral reef classification and thematic mapping. *Remote Sensing*, 5(4):1809–1841, 2013. 4

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[47] Alvy Ray Smith. Color gamut transform pairs. *ACM Siggraph Computer Graphics*, 12(3):12–19, 1978. 6

[48] Maricor Soriano, Sheila Marcos, Caesar Saloma, Miledel Quibilan, and Porfirio Alino. Image classification of coral reef components from underwater color video. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, volume 2, pages 1008–1013. IEEE, 2001. 3

[49] Lei Sun. Resnet on tiny imagenet. *Submitted on*, 14, 2016. 7

[50] Schrasing Tong and Lalana Kagal. Investigating bias in image classification using model explanations. *arXiv preprint arXiv:2012.05463*, 2020. 4

[51] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6

[52] Yan Wang, Wei Song, Giancarlo Fortino, Li-Zhe Qi, Wenqiang Zhang, and Antonio Liotta. An experimental-based review of image enhancement and image restoration methods for underwater imaging. *IEEE access*, 7:140233–140251, 2019. 3

[53] Ivor D Williams, Courtney S Couch, Oscar Beijbom, Thomas A Oliver, Bernardo Vargas-Angel, Brett D Schumacher, and Russell E Brainard. Leveraging automated image analysis tools to transform our capacity to assess status and trends of coral reefs. *Frontiers in Marine Science*, page 222, 2019. 3

[54] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4257–4266, 2021. 6

[55] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 4

[56] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 4

2

Appendix

2.1. Introduction

The world ocean covers approximately 71% of Earth's surface and contains 97% of our planet's available water [44]. It is not just a huge mass of water, it is also known as the main 'lung' of our planet as it produces at least 50% of the planet's oxygen which is more than all forests in the world combined [2]. Moreover, it absorbs roughly 30% of the carbon dioxide produced by humans. Coral reefs, also known as the tropical forests of the world ocean, play an important role in the ecosystems of the oceans. Coral reefs are essential for the regulation of the carbon dioxide levels in the oceans, keeping them balanced, which is vital for marine ecosystems. They are also the most densely populated marine environment and support many marine species by supplying a nutrient-rich habitat and a safe shelter. But coral reefs also have an indirect and direct impact on us, human beings. Most of us indirectly rely on coral reefs for food, coastal protection and as a source for new medicine. And, it is estimated that over half a billion people worldwide directly depend on coral reefs for, among other things, food, income and protection [1].

Coral reefs are threatened by, inter alia, coral bleaching, diseases, storms and human activity such as habitat destruction, over-fishing, pollution and the introduction of invasive species. Marine biologists are very concerned about the destruction of coral reefs [18] and the diminishing of coral reefs as they are one of the most productive and species-rich ecosystems in the world and play an important role on this planet.

2.1.1. Corals

Corals are marine invertebrates that typically live in compact colonies of many genetically identical individual polyps. Each polyp in these corals secretes a calcium-carbonate endoskeleton. This endoskeleton is left behind long after the polyp dies. Every new generation of polyps builds on the secreted endoskeleton of its predecessors. Thus, the large and hard physical structure of a coral is formed through numerous generations of endoskeleton secreting polyps. Coral growth is, therefore, not very fast. In general, a coral does not grow more than several millimetres per year [15]. These corals are called hard corals, but, there are also soft corals, these corals lack a skeleton and look similar to underwater plants. Corals can be divided into two main categories: hard and soft corals. Hard corals are, in general, the best indicator for a healthy reef. Both hard and soft corals are, taxonomically speaking, part of the class Anathozoa which falls under the phylum Cnidaria. Corals share the class Anathozoa with sea anemones. In total there are around 119 coral genera and over 6,000 coral species [53].

Coral reefs, found in tropical oceans and seas, are underwater structures that are primarily composed of colonies of corals. Most of these corals are so-called reef builders, which means that they are involved in building reefs, making them essential to coral reefs. Coral reefs are an important member of the benthic community. The benthos is the assemblage of organisms inhabiting the seafloor.

2.1.2. Coral bleaching

The primary food supply of almost all corals depends on protists and microscopic algae that live in the coral's tissues [18]. The symbiotic relationship between these algae and corals is known as mutualism since both species benefit from their symbiotic relationship [25]. These algae, the symbionts, thrive in warm, shallow and clear waters with enough nutrients. Coral reefs with corals that depend on such algae are therefore only

found in oceans and seas where these conditions are met. However, whenever these conditions change due to external factors, such as global warming, ocean acidification and pollution, a process called coral bleaching might occur. Due to the changing conditions, the symbionts that live in the corals die or become stressed and leave the corals. As corals get their colour from the algae that live within them, once these algae have left their host, the corals lose their colour and turn white or very pale, hence the term coral bleaching. The effect of coral bleaching on a coral can be seen in figure 2.1. Bleached corals are vulnerable as they have lost their primary source of energy and nutrients. Only a few bleached corals will manage to survive and recover, but most will eventually die.

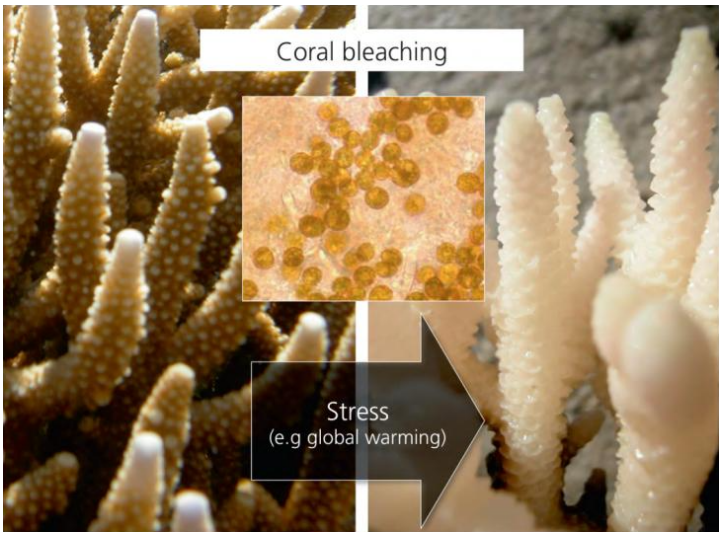


Figure 2.1: Coral bleaching. The usual brown colouration of this *Acropora* coral (left) is lost under environmental stress when the zooxanthellae (inset) are expelled from its tissue, leaving the white coral skeleton visible through the translucent tissue (right). Reprintend from [48].

Coral bleaching can occur on a global scale, in which massive amounts of corals bleach. Such mass global coral bleaching events greatly impact coral reefs, these events occur with increasing frequency and intensity since the late 1970s [26, 16, 27]. It is predicted that these events will occur annually from the year 2030 [13]. This would mean that the affected corals would only have a short recovery time. Such a high frequency of coral bleaching events would be destructive for many coral reefs. The diminishing of coral reefs would greatly impact a lot of species, as corals form the essential habitat of more than 25% of the world's marine species [4]. Though, also our species would be impacted by the diminishing of coral reefs. Some of the main benefits that we have from coral reefs include the following; they are a source of food and (new) medicine, provide jobs for local communities, protect coastlines from erosion and storms and provide opportunities for recreation [3].

2.1.3. Coral reef monitoring

Because of growing concerns, such as coral bleaching, institutions and organisations are conserving, protecting and managing coral reefs by monitoring them on a local and global scale. The main objective of monitoring coral reefs is to investigate how corals change over time, in which coral bleaching plays a huge role. Monitoring helps to understand coral reefs by gathering data. Typically, several types of data are collected, which includes; site surveying (information about the site), coral species surveying (information about coral species and possibly their status) and substrate surveying (information about non-coral species). Effective long-term monitoring of reefs establishes a baseline and is important for coral reef managers to shape, implement and reflect on successful policies that are required to protect and conserve the reefs [21, 23, 10]. To monitor coral reefs, divers would, in the early days, manually collect data from the reef during the dive. This quickly changed to image- and video-based surveying methods that require less in-water time (which is physically demanding [39]) and fewer operations. Moreover, the images and videos are re-analysable as they are permanent visual records of the surveyed site. Even though the time to conduct the survey decreases, the time needed to analyse the data generally increases for visual imagery-based surveying methods as the data still has to be annotated [29]. The data acquisition became more automated, Remotely Operated Vehicles (ROVs) or Autonomous Underwater Vehicles (AUVs) are used to quickly and accurately collect a lot of visual

data [34]. This also resulted in an exponential increase in the size of coral reef monitoring datasets, of which the data still needs to be annotated for it to be of significant use. Annotated datasets can be used by marine biologists to measure a range of interesting and important metrics such as the reef's biodiversity (benthic cover) and conservation status using indices such as the Shannon index [33].

Annotating the data is, however, time-consuming as taxonomists have to manually annotate the imagery which is a tediously repetitive task and demands a lot of time. Moreover, it requires well-trained taxonomists, as coral reefs are home to thousands of species. Yet, for some groups of benthic organisms, identification from visual imagery is difficult even when done by experts [8, 43]. Moreover, long-term datasets are often scattered or spatially constrained and the field data is far from standardised. This all results in a bottleneck in the flow of information from monitoring programs to managers, which delays conservation decisions [21, 22].

2.1.4. Benthic survey imagery

Coral reef visual imagery data is often gathered in the form of benthic survey images. These are images acquired by divers or AUVs. There exist several techniques to collect this data, the most commonly used technique for coral reef monitoring datasets is that of the classical photographic quadrat survey. This technique involves a transect line of a certain distance that is laid out across a coral reef, the diver follows this transect line and every time places a quadrat of a certain size (often 1 m^2) on the benthos and takes a picture of this quadrat from a fixed distance [29]. The result is a set of images from a coral reef that should more-or-less have the same dimensions.

To annotate these images, random point annotations are used. Software is used to randomly select a predefined number of points, each point representing a pixel in the image, that need to be labelled by an expert. Figure 2.2 shows how random point annotations are extracted from a benthic survey image.

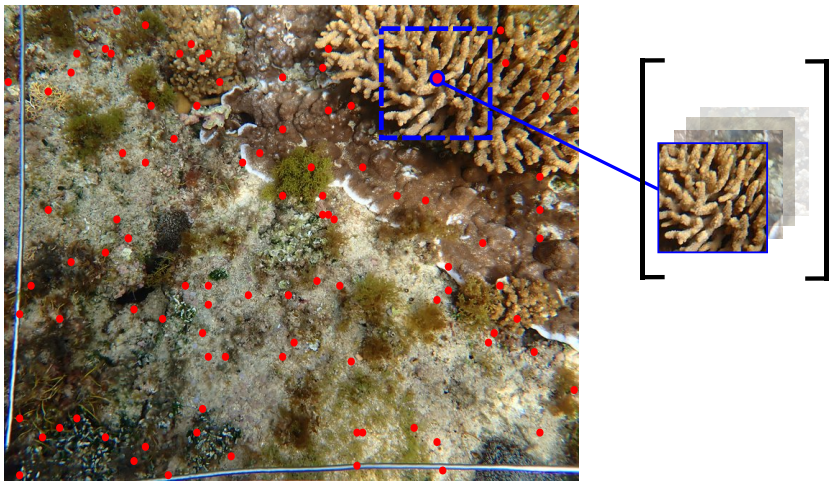


Figure 2.2: Random point annotations. A 100 random points (in red) are placed on the benthic survey image of which each point will be annotated. To train a classifier, the random points will be extracted from the image as image patches of 150 by 150 pixels. The surrounding pixels act as context for the classifier to extract information from. This results in a 100 annotated image patches. The images are from the publicly available 'Norfolk_isl_Apr_2022' source on CoralNet.

The camera setup used for these surveys is often specific to the survey. Most camera setups only use regular reflectance cameras that capture images from the visible light spectrum in the RGB colour space, however, some surveys use additional cameras that capture other wavelengths and different colour spaces [12, 9]. Capturing wide-band fluorescence photographs is a method that might add information to the content of benthic coral reef surveys [10].

2.1.5. Automating coral reef monitoring

The process of annotating benthic survey imagery was in dire need of modernisation as these datasets can be so large that the analysis would take years of manual annotating. So, to automate the annotation process, computer vision and machine learning methods have been applied to these datasets [10]. The first of these computer vision-based methods included the use of hand-crafted features in combination with a classifier [38, 41, 45, 37, 50, 7, 8].

More recently, the field of deep learning has gotten more promising results for the analysis of visual data.

Therefore, researchers in coral reef monitoring began to implement deep learning methods to improve automation of benthic survey data analysis. These methods include the use of Convolutions Neural Networks (CNNs) [30]. The use of CNNs for coral bleaching detection will be briefly expanded upon in this thesis.

2.1.6. Dataset biases

It is important to critically analyse datasets before using them to automate tasks, such as coral bleaching detection in the case of this thesis. Datasets may contain biases which might be learned by a model. These biased algorithmic outcomes might affect the behaviour of the users of these biased results. As an outcome, the users might introduce biases again into the acquisition of additional data [40]. This is a feedback loop that could become a vicious circle, which should be avoided. The detection of biases in data is, therefore, paramount for the effective use of most machine learning models, and in this specific case, for further research in coral bleaching detection. The detection of biases in a coral bleaching dataset and the detection of biases in the dataset is exactly what we will explore in this thesis.

2.2. Related Work

2.2.1. Preprocessing of underwater imagery

Benthic survey datasets are obtained through underwater photography. When waterproof cameras are used underwater, they can suffer from many issues that translate to artefacts in the obtained images. These issues include blurring, noise, colour diminishing and light attenuation that are caused by varying conditions such as depth, water temperature, turbidity and current [17]. To minimise these artefacts and improve image quality, image enhancement and restoration techniques have been developed. These methods can be divided among three categories; model-free, model-based and data-driven [54, 31]. The major drawback of model-free methods is that these only rely on the observed information which is hardly enough to enhance underwater images due to their complexity, especially with degraded underwater images. Model-based methods, on the other hand, may also not be enough to enhance underwater images because the priors are not always transferable from one scenario to the other and these priors may not even be available [31]. Even though the methods of both the model-free and model-based category often improve the visual quality of most underwater images [31], experimental research has shown that using most model-free image enhancement algorithms to preprocess the images does not necessarily improve the accuracy of a CNN that tries to label these images [17].

2.2.2. Classification problems on coral images using transfer learning

To effectively train these complex deep learning models, a large amount of training data is necessary. However, one of the main problems with coral reef monitoring data is that only a small portion of the data is actually labelled. Thus, most of the datasets used in literature do not have many samples. Transfer-learning is used to increase the model's performance when there is a shortage of data as this tends to be more effective than training a small network from scratch [56]. A CNN, for example, can learn feature maps that extract different features at different depths of the network. Low-level layers capture low-level features (i.e. corners and edges) and high-level layers capture high-level features (i.e. shape and texture) and are generally more class-specific [57]. That is partly why CNNs perform so well across different domains. Allowing features to be transferred from one domain to another whilst maintaining their discriminating power. This is especially useful in the case of coral reef classification where there is not a lot of labelled data available. Thus, a common solution in research is to use a pre-trained CNN as a generalised feature detector that is pre-trained on a dataset from a different domain containing a large variety of objects and backgrounds. The pre-trained CNN, therefore, learns to extract unique information such as colour, texture and shape. The model is then fine-tuned on context-specific data, coral reef images in the case of coral reef classification. However, pre-training is only a form of transfer learning if the data on which the model is pre-trained comes from a different domain than the data on which the model is fine-tuned and evaluated. Transfer learning is therefore also commonly used in coral classification tasks [21, 35, 20]. Thus, this means that the picked CNN architecture is first pre-trained on a large dataset (e.g. ImageNet [14], containing millions of images and thousands of classes). Then, the CNN is fine-tuned by training the CNN on the context-specific dataset.

2.3. Bias analysis for coral bleaching detection

2.3.1. Bias detection methods

t-distributed Stochastic Neighbor Embedding (t-SNE). t-SNE [52] is a method to visualise high-dimensional data on a (low) two or three-dimensional space using non-linear dimensionality reduction. This means that the iterative algorithm is able to separate data that is not linearly separable, thus a straight line. It models the probability distribution of neighbours around each point based on the Cauchy kernel (Student's t-distribution with 1 degree of freedom) to most accurately reconstruct the high-dimensional feature space. The student's t-distribution, as opposed to the Gaussian distribution, has a fatter tail that does not heavily penalise less similar data pairs in the low-dimensional projection. This helps the algorithm avoid the crowding problem, thus it spreads the data points more evenly in the low-dimensional feature space. The perplexity, a parameter that has to be set by the user to control the fitting, defines whether to mainly focus on the nearest neighbours (low perplexity) or to also take the further away neighbours into account (high perplexity). We use a perplexity value of $N^{\frac{1}{2}}$ where N is the number of samples and the perplexity value is capped between 5 and 50 (as suggested by the authors of t-SNE).

This method is especially effective to visualise the structure of large imagery datasets using feature embeddings from a CNN, making it an effective tool to analyse the relations between images fed to the network. The output, a two or three-dimensional mapping of the extracted features, displays how the network clusters extracted features. Where neighbouring images share similarities as interpreted by the network.

Gradient-weighted Class Activation Mapping (Grad-CAM). Figure 2.3 shows how Grad-CAM [47] uses a class activation map to generate a heatmap on top of the input image that highlights regions in the image that are important for the classification of the image. The class activation map is obtained by using the gradients of the classification score (before softmax) with respect to the feature map activations of the selected convolutional layer (after which ReLu has been applied) to localise the regions of the image that influence the classification score the most. This boils down to a linear combination of matrix products of the weight matrices and the gradient with respect to activation functions till the target convolutional layer, the layer to which the gradients are being propagated to. ReLu is applied on the resulting linear combination because the heatmap should only highlight the positive influence of a class, thus the pixels that positively influence the prediction probability of the target class.

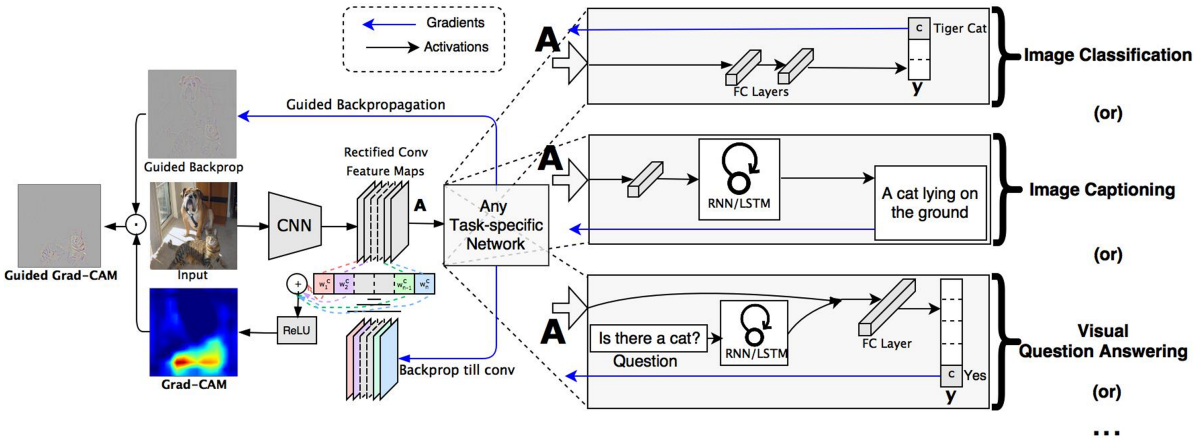


Figure 2.3: Grad-CAM overview. Given an image and a class of interest as input, the image is forward propagated through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class, which is set to 1. The signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localisation (blue heatmap) which represents where the model has to look to make the particular decision. Finally, the heatmap is pointwise multiplied with guided backpropagation to get the Guided Grad-CAM visualisations which are both high-resolution and concept-specific. Reprintend from [47].

Learning to Split for Automatic Bias Detection (LS). LS [5] is a recently published method in the field of automatic bias detection where the source of bias is unknown during training and validation. This meta learning method involves a bi-level optimisation problem where the inner- and outer-loop have to cooperate with each other to maximise a generalisation gap between a training and testing split. In the inner-loop it trains a Predictor on a classification task. Once converged, the predictions from the converged model will be used in the outer-loop by a Splitter model that learns to place correctly predicted samples in the training set,

which cannot generalise on the test split, while adhering to two constraints. These constraints are defined to avoid finding a split with (i) a shortage of training samples and (ii) a class imbalance among the splits. This amounts to a Splitter playing an adversarial game with the Predictor that is trained until convergence in every iteration of training the Splitter.

The LS algorithm (see Algorithm 1) uses Eq. 2.1 to calculate the regularity constraints and Eq. (2.2 - 2.3) to calculate the losses. The algorithm is able to find challenging splits by refining the Predictor based on the iteratively updated Splitter.

Algorithm 1 Learning to Split (LS). Reprinted from [5].

Input: dataset $\mathscr{D}^{\text{total}}$.
Output: data splits $\mathscr{D}^{\text{train}}$, $\mathscr{D}^{\text{test}}$.
  Initialize $Splitter$ as random splitting.
  repeat ▷ Outer-loop
    Apply $Splitter$ to split $\mathscr{D}^{\text{total}}$ into $\mathscr{D}^{\text{train}}$ and $\mathscr{D}^{\text{test}}$.
    Initialise $Predictor$ and train $Predictor$ on $\mathscr{D}^{\text{train}}$ using emperical risk minimisation.
    Eveluate $Predictor$ on $\mathscr{D}^{\text{test}}$ and compute its generalisation gap.
    repeat ▷ Inner-loop
      Sample a mini-batch from $\mathscr{D}^{\text{total}}$ to compute the regularity constraints Ω_1, Ω_2 (Eq. 2.1).
      Sample another mini-batch from $\mathscr{D}^{\text{test}}$ to compute $\mathscr{L}^{\text{gap}}$ (Eq. 2.2).
      Update $Splitter$ to minimise the overall objective $\mathscr{L}^{\text{total}}$ (Eq. 2.3).
    until $\mathscr{L}^{\text{total}}$ stops decreasing.
  until gap stops increasing.

The regularity constraints from LS are defined as:

$$\Omega_1 = \mathrm{D}_{KL}(\mathbb{P}(z)||\text{Bernoulli}(\delta)), \tag{2.1}$$
$$\Omega_2 = \mathrm{D}_{KL}(\mathbb{P}(y|z=1)||\mathbb{P}(y)) + \mathrm{D}_{KL}(\mathbb{P}(y|z=0)||\mathbb{P}(y)).$$

where Ω_1 avoids finding a split with a shortage of training samples where the marginal distribution $\mathbb{P}(z)$ represents the ratio of the train and test split. The Splitter is penalised when this distribution moves too far away from the prior distribution Bernoulli(δ)) with δ which is the user defined split ratio, this amounts to train / (train + test). Ω_2 avoids finding an imbalanced split. The Splitter is penalised when the label marginals in the training split $\mathbb{P}(y|z=1)$ and the test split $\mathbb{P}(y|z=0)$ move too far away from the original label marginal $\mathbb{P}(y)$.

The gap loss from LS is defined by:

$$\mathscr{L}^{\text{gap}} = \frac{1}{|\mathscr{D}^{\text{test}}|} \sum_{(x_i,y_i)\in\mathscr{D}^{\text{test}}} \mathscr{L}^{\text{CE}}(\mathbb{P}_{Splitter}(z_i|x_i,y_i), \mathbb{1}_{y_i}(\widehat{y}_i)). \tag{2.2}$$

where x_i is the input, y_i is the label, $\widehat{y}_i$ is the Predictor's prediction for which the cross entropy loss between $\widehat{y}_i$ and the Splitter's prediction correctness over the testing split is minimised.

The total loss from LS is defined by the sum of the regularity constraints and the gap loss:

$$\mathscr{L}^{\text{total}} = \mathscr{L}^{\text{gap}} + \Omega_1 + \Omega_2. \tag{2.3}$$

2.4. Coral bleaching dataset analysis

2.4.1. CoralNet Class Balanced Bleaching (CCBB) dataset

We introduce the CCBB dataset using images scraped from the CoralNet[1] project [7, 6, 8, 11, 55]. Figure 2.4 shows the pipeline for the data acquisition and preprocessing of the data. Reference the scientific paper for a table with an extensive overview of the dataset composition.

Data augmentations. To artificially increase the variability of the distribution of the training samples, we augment training data during training time. The aim of these augmentations is to avoid overfitting the model on the training data. The augmentations that we use do not change the spatial pattern of the target class and are inspired by augmentations proposed in previous research on coral classification using benthic survey

[1] `https://coralnet.ucsd.edu/`

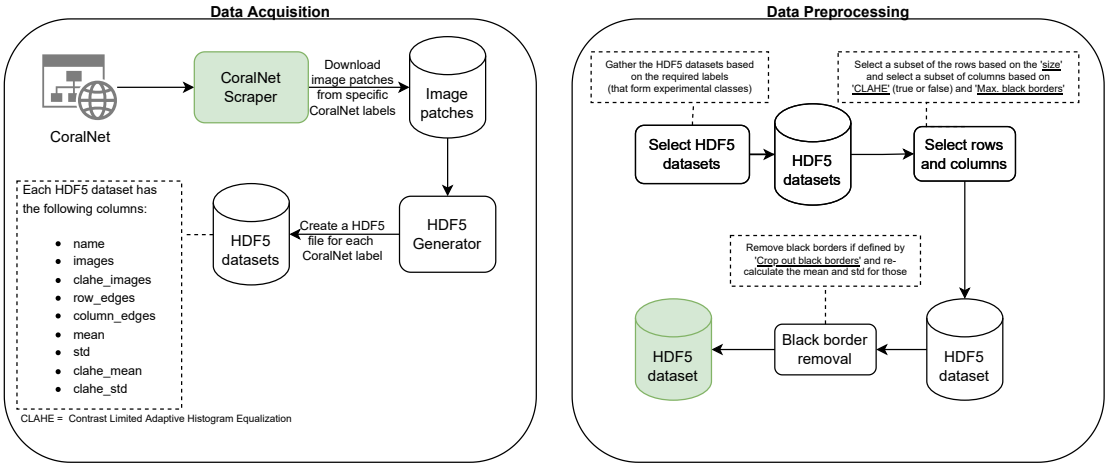


Figure 2.4: Pipeline for the data acquisition from CoralNet and for the for creation of the experimental datasets including the preprocessing steps. We scrape publicly available data from CoralNet to obtain a large dataset of healty and bleaching coral images. The assembly of the final dataset depends on the parameters defined in the preprocessing pipeline.

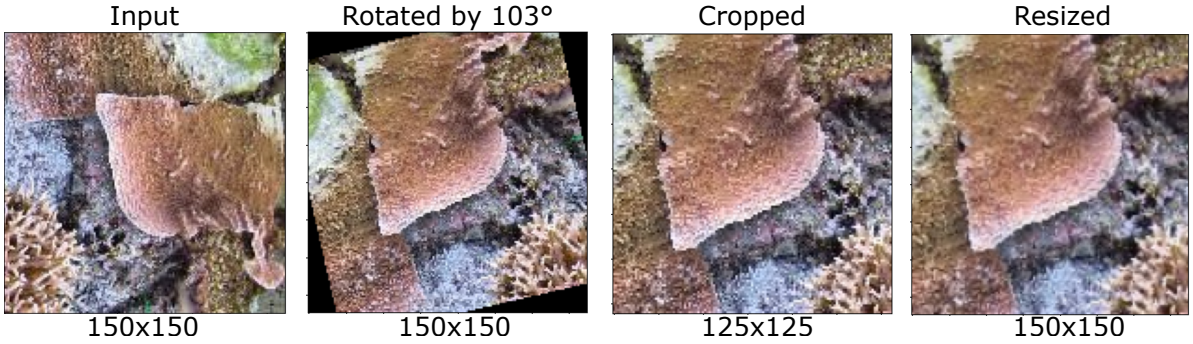


Figure 2.5: The custom random rotate data augmentation. The input image of 150x150 pixels is first rotated by a random angle, then cropped (at the centre) to remove the black corners after which the image is resized to the original size of 150x150 pixels.

imagery [20, 19]. The augmentations, if used in any experiment, are (i) a random rotation followed by a centre crop to remove the possibly added black corners due to rotation, as seen in Figure 2.5, and (ii) a random centre crop. Both augmentations do not change the centre of the image patch, as this represents that which is actually annotated. And both augmentations finish with a re-scale to resize the augmented images to the original dimensions. Another augmentation used on the training data but also on the validation data is a normalisation using the mean and standard deviation of the training samples.

2.4.2. Experimental setup

Model architecture. The architecture that we use for the experiments is the Residual Network (ResNet) [24] architecture. The specific version that we use is the ResNet-18 architecture, depicted in Figure 2.6, which shows superior performance on coral classification tasks when training a model from scratch or fine-tuning a model [42, 36, 20, 19, 46, 32], it is well suited for smaller datasets [51], it is deeper yet smaller than some other architectures like VVGNet [49], outperforms SqueezeNet [28] (see Table 2.1), and the on ImageNet [14] pre-trained model is available for PyTorch[2]. The experiments that required the use of a CNN have been adjusted to use the ResNet-18 model.

The authors of ResNet introduced the concept of an 'identity shortcut connection' which allows the model to skip layers. Stacking these residual blocks makes it possible to train much deeper networks by solving the degradation problem. Figure 2.7 shows such a residual skip connection.

Model setup. For the ResNet-18 model, we used the Stochastic Gradient Descent (SGD) optimiser with a

[2] `https://pytorch.org/hub/pytorch_vision_resnet/`

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2.6: ResNet architecture. We use the 18-layer ResNet architecture in our experiments. Reprinted from [24].

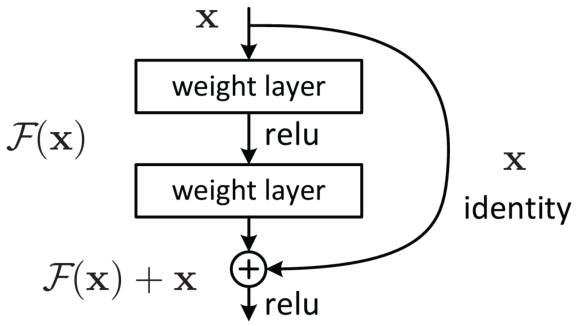


Figure 2.7: Residual learning: a building block. The 'identity shortcut connection' allows for information to flow from one layer to a deeper layer by skipping the layers in between. Reprinted from [24].

learning rate of 0.001 and a momentum of 0.9. This optimiser has shown superior performance compared to Adam in our experiments (see Table 2.1). For the learning rate, we used a scheduler that would decay the learning rate by 0.1 every 7 epochs with a gamma of 0.1. As a criterion, we used the cross-entropy loss. Each experiment ran for 20 epochs with batches of 128 samples. To cross-validate the results we used a 10-fold cross-validation which amounts to a 90/10 training/validation split for each fold.

Hyperparameter optimisation		Accuracy		Loss	
		Importance	Correlation	Importance	Correlation
Model	ResNet-18	0.129	**0.248**	**0.084**	-0.029
	SqueezeNet	0.056	-0.248	0.052	0.029
Optimiser	SGD	**0.243**	0.204	0.072	**-0.236**
	Adam	0.151	-0.204	0.09	0.236

Table 2.1: Hyperparameter tuning for the model and the optimiser when training and validating a CNN on the CCBB dataset. Correlation is the linear correlation between the hyperparameter and the chosen metric (accuracy or loss on the validation set in this case). So a high correlation means that when the hyperparameter has a higher value, the metric also has higher values and vice versa. Importance[†] is calculated using a random forest trained on hyperparameters as input and the metric as the target output. To obtain the accuracy and loss of the validation split, the model has been trained using a 5-fold cross-validation scheme, thus the results are averaged across these 5 runs. The accuracy should be maximised (more positive correlation is better) and the loss should be minimised (more negative correlation is better). The results indicate that the combination of the ResNet-18 model with the SGD optimiser performs the best for coral bleaching detection on the CCBB dataset. [†]`https://docs.wandb.ai/ref/app/features/panels/parameter-importance/`

2.4.3. Initial analysis for bleaching detection

To get an indication of how well the task of coral bleaching can be learned by a CNN, we train ResNet-18, both from scratch and fine-tuned, without and with data augmentations (as described above). When fine-

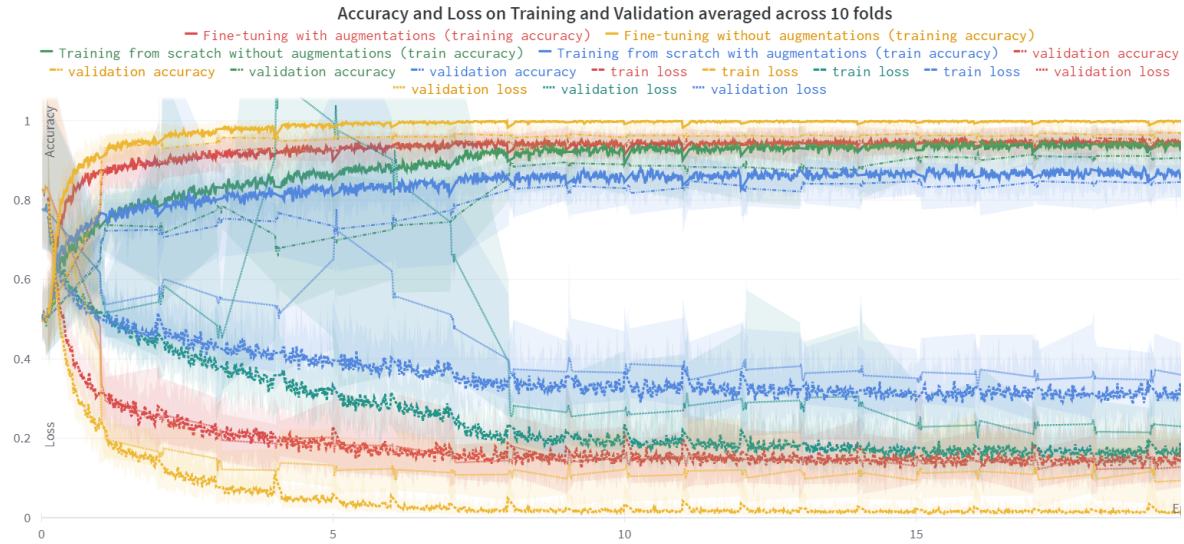


Figure 2.8: Learning curves of training ResNet-18 models on the CCBB dataset for coral bleaching detection. The x-axis represents the amount of epochs and the y-axis is shared, representing the accuracy (where 1 = 100%) and the loss. None of these models seem to overfit on the data since there are no big gaps between the training and validation curves, they are quite similar. The models seem to converge quite fast, some models need a bit more time for the performance on the validation set to improve and converge. But generally after 10 epochs the model is converged. The validation accuracy of the models is never lower than 80% for a converged model. From these learning curves it is clear that the fine-tuned models outperform the models trained from scratch. And that the data augmentations decrease the performance gap between the training and validation sets.

tuning the model, we refer to fine-tuning the CNN by initialising the model using the pre-trained ResNet-18 model before training it. Thus we do not freeze the weights of ResNet-18 as typically done when performing transfer learning with the pre-trained network as a feature extractor where only the last fully-connected layer is replaced and trained. The results are surprising, as the model showed exceptional performance, especially when using transfer learning by fine-tuning the model, as seen in Figure 2.8 and Table 2.2. Besides the surprisingly accurate models, the difference between the performances given the different setups are predictable. It is expected from a fine-tuned model to perform better than training a model from scratch. Just as it is explainable that data augmentations close the performance gap between the training and validation set because the augmentations (performed during training) help the model learn from a bigger data distribution, helping it generalise better for unseen data, the validation samples.

Training method	Augmentations	ROC AUC score
Training from scratch	Yes	0.8285
	No	0.8947
Fine-tuning	Yes	0.9448
	No	**0.9579**

Table 2.2: Training ResNet-18 on the CCBB dataset for coral bleaching detection. The ROC AUC score is obtained from the last epoch, when the model converged, on the validation set and averaged across the 10 runs using the 10-fold cross-validation scheme. The fine-tuned models outperforms the models trained from scratch, despite using augmentations or not, because it has already been trained on a lot of images and therefore knows how to extract useful image features. The augmentations make it harder for the model to learn the exact data distribution as the samples are augmented during training time, increasing the variety of the samples.

2.4.4. Manual bias analysis for bleaching detection

Here we use Grad-CAM to explore possible biases in the CCBB dataset. The idea using this method is to perform an exploratory analysis on the biases in the predictions of a CNN trained on the CCBB dataset. The expectation is that the output of Grad-CAM for biased images is a heatmap that indicates that regions, such as intrusive objects, are important for the classification of coral bleaching such that these regions are highlighted in the heatmaps. This would then support the intrusive object hypothesis (as hypothesised in the scientific paper) and might even help us formulate new bias hypothesis. We use the fourth convolutional layer from a

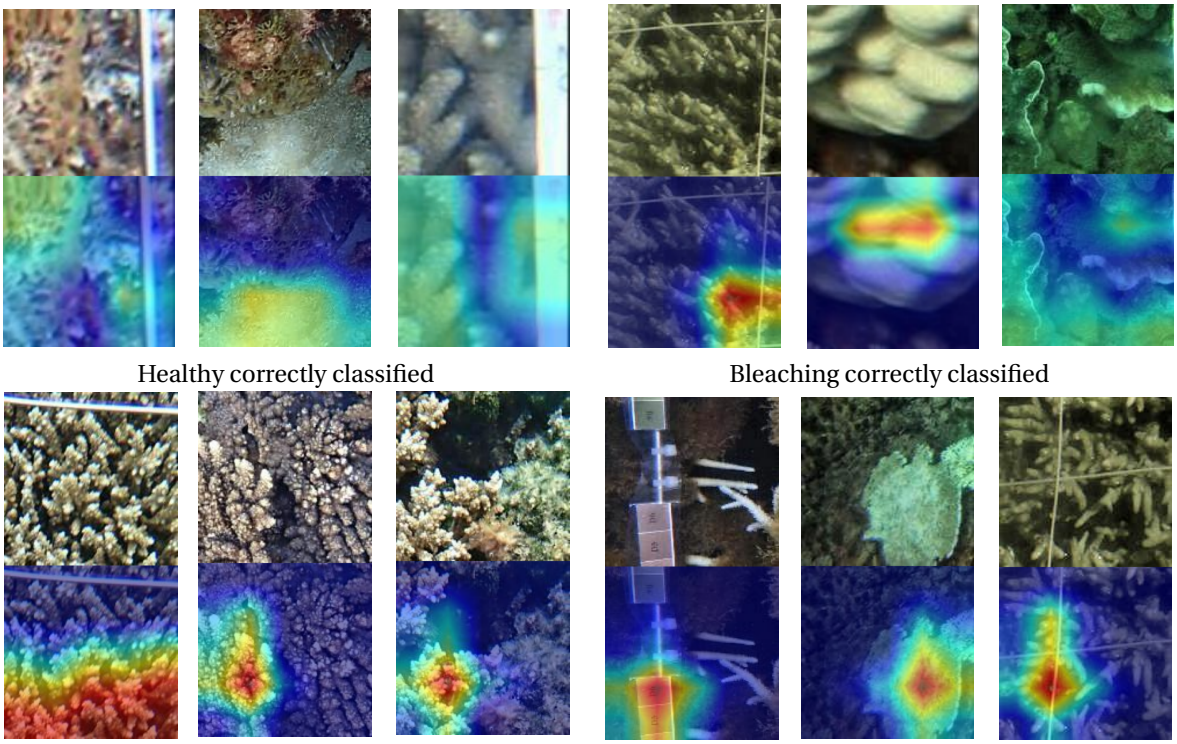


Figure 2.9: Image patches with their Grad-CAM activation maps stacked on top of each other. The class activations are highlighted using a heatmap with a gradient from blue to red where blue means little to no positive class activations and red means strong positive class activations. No clear conclusion can be drawn from these heatmaps, other than that the model is often able to focus on corals in the images to classify whether the image contains a bleached or healthy coral.

ResNet-18 model, trained (without transfer learning) on the CCBB dataset without any data augmentations, to get the class activations from. Some of these heatmaps are depicted in Figure 2.9. The results show that the model is able to focus on corals from time to time. However, it also occurs that the model does not really activate on any particular part of the image. Some of these images have invasive objects (such as a measuring tape or some lines) but the model often does not seem to activate based on these objects. The model only seems to activate based on an object for one such images in Figure 2.9 (first bleaching misclassified image). There also seems to be no visual relationship between the heatmaps, classes and whether these are correctly classified or not. We conclude that the Grad-CAM method to manually find biases in the CCBB dataset is a difficult task and the results show no clear biases.

2.4.5. Automatic bias analysis for bleaching detection

The result from the LS method that is used in these experiments is selected among several runs. By picking the result with the biggest generalisation gap, being 29.81%, we select the most interesting split to analyse as the biases should be separated the most between these splits among all the LS runs.

The random split, that will be referred to as $\mathscr{D}^{\text{random}}$, is a random 75/25 (train/test) dataset split with a minimal generalisation gap of 0.19%. The worst split, that will be referred to as $\mathscr{D}^{\text{worst}}$, is the split with the biggest generalisation gap (29.81%) found by running the LS method several times. The data from these splits will be referred to as $\mathscr{D}^{\text{split}}_{\text{type}}$ where *split* is either *random* or *worst* and *type* is either *train*, *test* or not specified in case it refers to all the data of the *split*.

Figure 2.10 shows a t-SNE plot that shows the data distributions of $\mathscr{D}^{\text{random}}$ and $\mathscr{D}^{\text{worst}}$ when extracted by the corresponding Predictor. The data points are represented by the actual image patches which helps us interpret the data distribution and the possible formed clusters. There are several clusters formed that shows the importance of features extracted from the images. These include the colour mask of an image forming clusters of green, yellow, grey or blue images. Also the sharpness of an image plays a role in the grouping of images, as well as the presence of objects (e.g. a grid of red lines or white tubes).

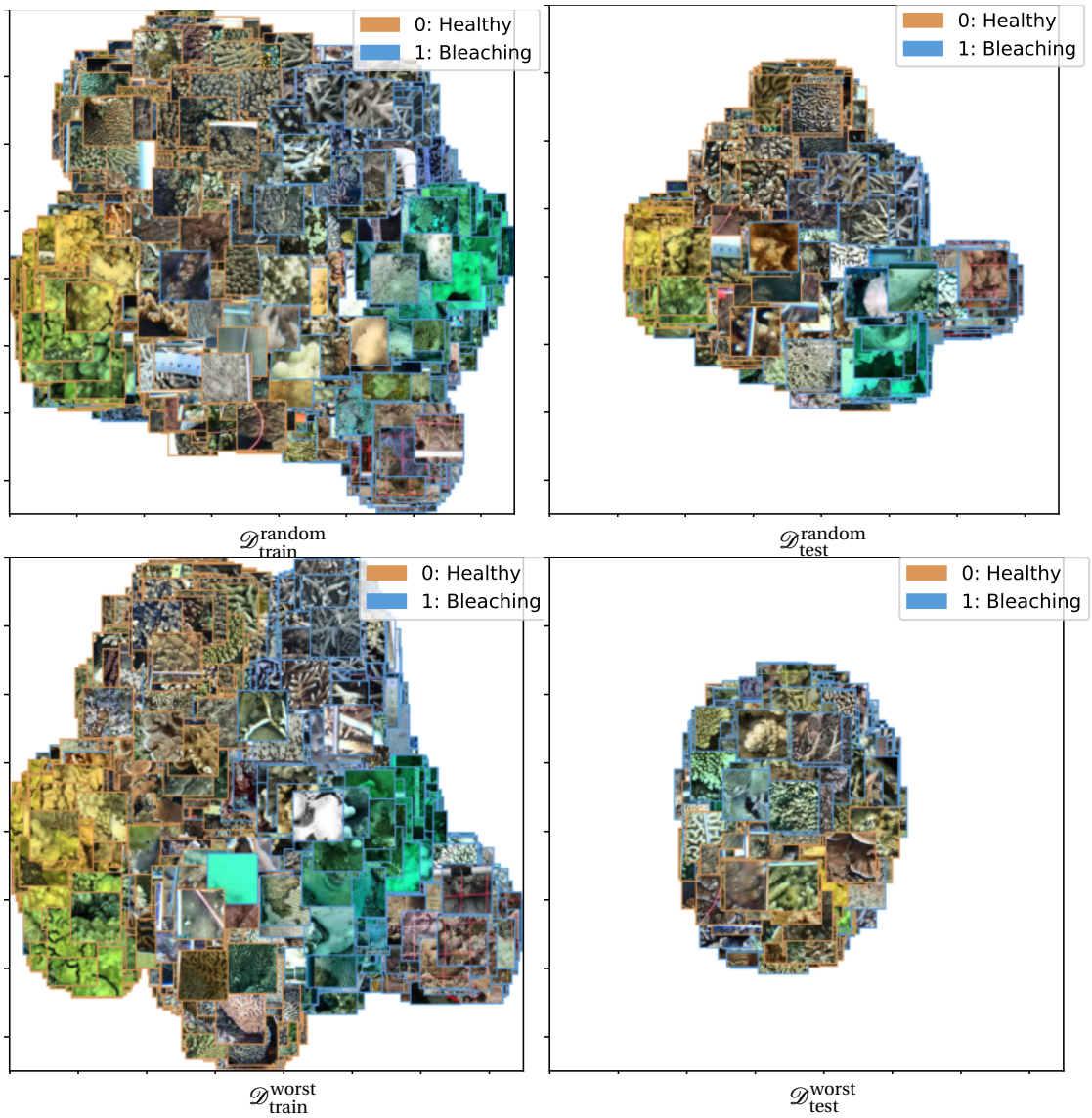


Figure 2.10: The t-SNE plots for $\mathcal{D}^{\text{random}}$ and $\mathcal{D}^{\text{worst}}$, both using features extracted by the corresponding trained ResNet-18 models. The images are clustered together based on visual features such as colour, sharpness and whether there are some objects in the image (e.g. a grid of red lines). The distribution of $\mathcal{D}_{\text{test}}^{\text{random}}$ is quite similar to $\mathcal{D}_{\text{train}}^{\text{random}}$ as the data is just randomly split. However, the distribution of $\mathcal{D}_{\text{test}}^{\text{worst}}$ is not similar to $\mathcal{D}_{\text{train}}^{\text{worst}}$. The Splitter obtained $\mathcal{D}_{\text{train}}^{\text{worst}}$ and $\mathcal{D}_{\text{test}}^{\text{worst}}$ as the Predictor correctly classified samples from $\mathcal{D}_{\text{train}}^{\text{worst}}$ and misclassified samples from $\mathcal{D}_{\text{test}}^{\text{worst}}$. This is supported by the distribution of the Predictor's extracted features as those are well divided for $\mathcal{D}_{\text{train}}^{\text{worst}}$ and more overlapping for $\mathcal{D}_{\text{test}}^{\text{worst}}$, thus it is more prone to misclassify $\mathcal{D}_{\text{test}}^{\text{worst}}$. The Predictor learned to extract features from the $\mathcal{D}_{\text{train}}^{\text{worst}}$ samples such that these discriminate well for only $\mathcal{D}_{\text{train}}^{\text{worst}}$ samples for the given classification task. Thus, this suggests that the $\mathcal{D}_{\text{train}}^{\text{worst}}$ contains biases that are less prevalent in $\mathcal{D}_{\text{test}}^{\text{worst}}$ such that the learned biases do not help with the classification of $\mathcal{D}_{\text{test}}^{\text{worst}}$ samples.

The CoralNet label bias hypothesis. The t-SNE of $\mathscr{D}^{\text{random}}$ using the corresponding trained model can be observed in Figure 2.11 which shows no significant difference between $\mathscr{D}^{\text{random}}_{\text{train}}$ and $\mathscr{D}^{\text{random}}_{\text{test}}$ for the distribution of the samples based on the CoralNet label they come from. The model seems to cluster the samples based on the CoralNet label they come from and this happens for both the $\mathscr{D}^{\text{random}}_{\text{train}}$ and $\mathscr{D}^{\text{random}}_{\text{test}}$. However, when comparing this to $\mathscr{D}^{\text{worst}}$ in Figure 2.11, it shows that this only plays a role in the feature extraction of the model for $\mathscr{D}^{\text{worst}}_{\text{train}}$ as $\mathscr{D}^{\text{worst}}_{\text{test}}$ does not contain such specific clusters. This strongly suggest that the CoralNet label from which images come from does form a bias for classes when classifying coral bleaching images.

Figure shows pie charts in which we quantify the ratio of samples from CoralNet labels between and among splits. From this quantification we see that the CoralNet label forms a bias in the CCBB dataset. Most noticeable is the lack of the samples from CoralNet labels 1764 and 2060 in the $\mathscr{D}^{\text{worst}}_{\text{test}}$ split, these are samples from the bleaching class. This suggests that these samples might be very biased.

CoralNet label bias conclusion: *The CoralNet labels of which samples come from forms a bias in the CCBB dataset. We show this using t-SNE plots of samples by colouring the samples according to the CoralNet label. The difference in the distributions of $\mathscr{D}^{random}$ and $\mathscr{D}^{worst}$ in the t-SNE plots and in the qualitative analysis verify the CoralNet label bias.*

The saturation and value bias hypothesis. We use histograms to see and quantify the distributions of the saturation and value from the HSV image domain between and among splits (see Figure 2.13). The distributions of $\mathscr{D}^{\text{random}}$ are very similar, as expected for a random split as the samples are randomly distributed among the split. For $\mathscr{D}^{\text{worst}}$, there is also no difference in the distribution of saturation or value values among $\mathscr{D}^{\text{worst}}_{\text{train}}$ and $\mathscr{D}^{\text{worst}}_{\text{test}}$. Since the images are all preprocessed using CLAHE, the saturation distribution of the images is more normalised which explains that this potential bias has been mitigated. And for the value distribution of the images, there also seems to be no bias.

Saturation and value bias conclusion: *The saturation and value of the images do not form biases in the CCBB dataset as shown using the histograms to quantify the distributions of the saturation and value among and between LS splits.*

2.5. Discussion

The manual bias detection Grad-CAM is hard to interpret for this type of data, especially when the class activations are not focused on any objecst at all. Despite generating and analysing a huge amount of Grad-CAM results, we can not find any relation between heatmaps, specific regions in images, classes and predictions. Therefore, we conclude that the manual detection method is insufficient to detect biases in the CCBB dataset. However, when combining this method with other methods proposed in literature, such as to cluster the heatmaps, it might unlock the potential to use the Grad-CAM results more effectively.

The automatic bias detection method, in combination with some visualisation techniques, helps find biases. The CoralNet label seems to play an important role when LS tries to find a maximum generalisation gap. Despite verifying this bias, it most likely is not a bias on its own but rather a source of multiple biases. CoralNet labels come from specific sources and these differ for most labels. Given that the sources use different camera setups and acquire their data from different places using different protocols, source specific artefacts will appear. Thus, actually the sources of images cause biases and since the data is distributed among labels with from different sources, labels will have different biases. Based on which class these labels belong to, healthy or bleaching, the biases might become class specific causing dataset biases. Discovering which labels contain most biases does help identify sources that cause biases by limiting the search space. Yet, it would still require the coupling of image patches to specific sources which is only possible for a small part of the data that we use from CoralNet to obtain the CCBB dataset.

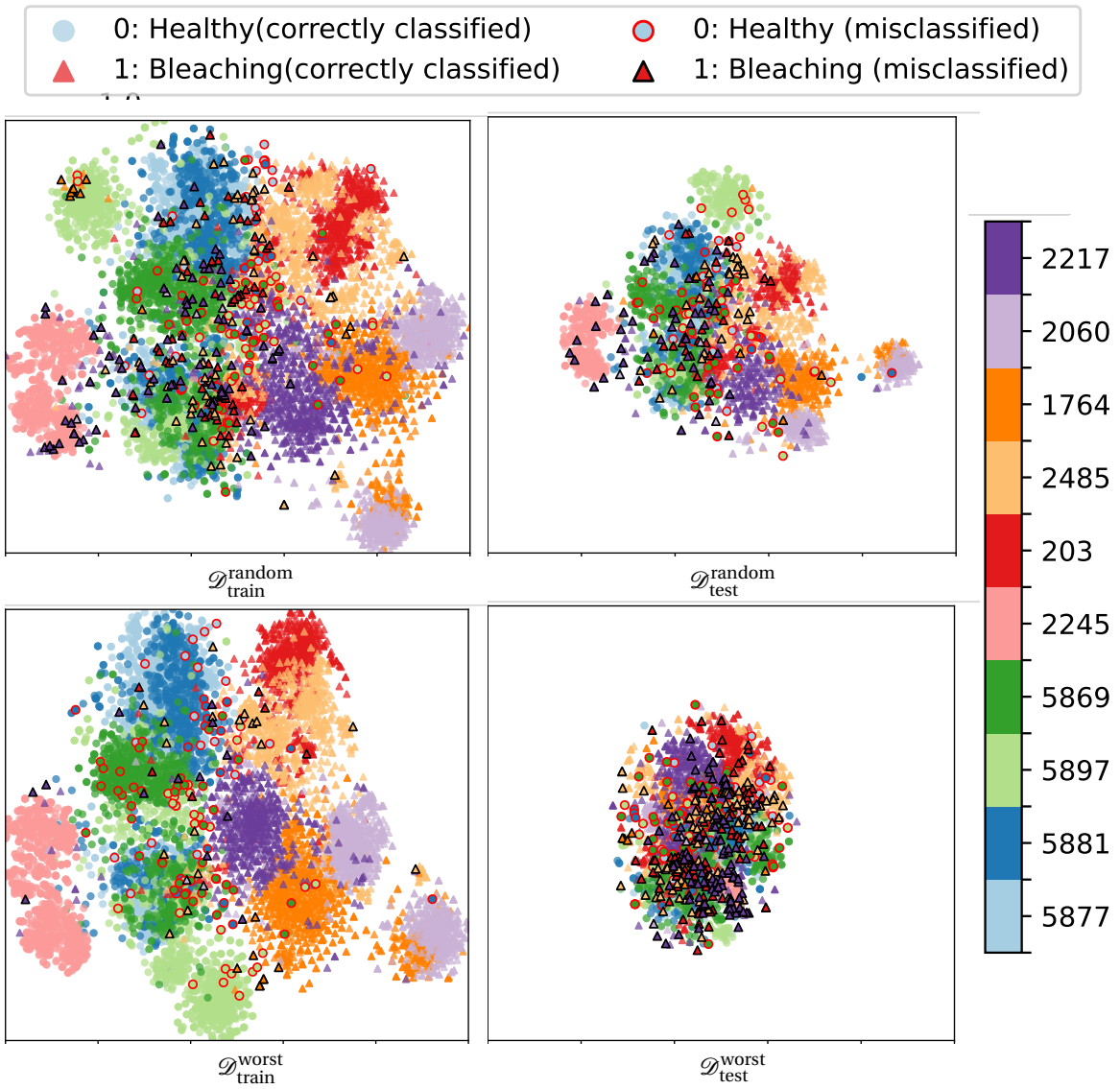
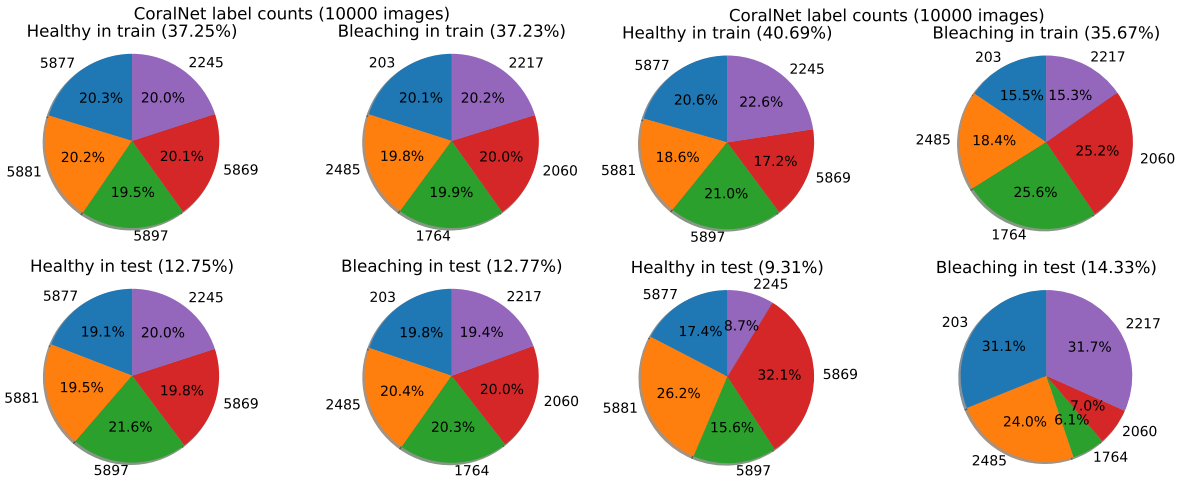


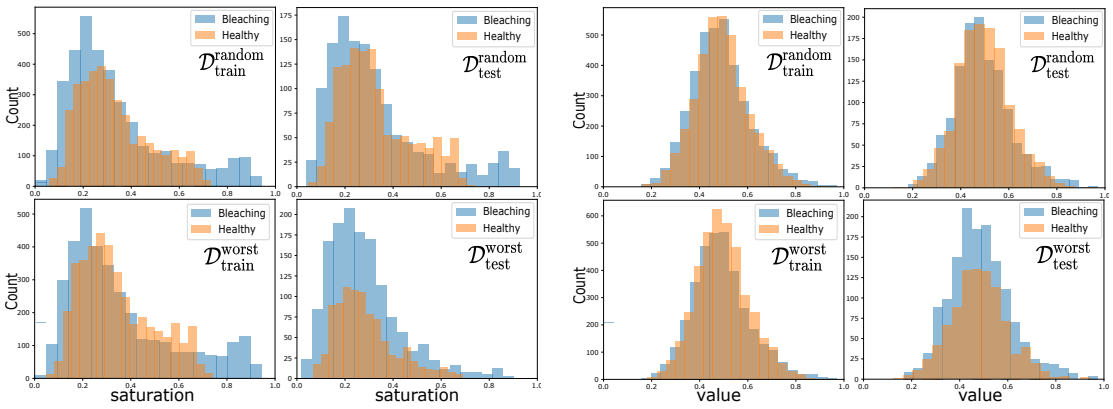
Figure 2.11: **CoralNet label hypothesis.** The t-SNE plots for $\mathscr{D}^{\text{random}}$ and $\mathscr{D}^{\text{worst}}$ highlighting the CoralNet label by colouring the samples, both using features extracted by the corresponding trained ResNet-18 models. The CoralNet label to play an important role for models when extracting features. Samples of the same CoralNet label are clustered together, this holds for almost all CoralNet labels, except for samples from the CoralNet labels 5877, 5869, 5881 and 5897. The distributions of $\mathscr{D}^{\text{random}}_{\text{train}}$ and $\mathscr{D}^{\text{random}}_{\text{test}}$ are very similar whereas the distribution of $\mathscr{D}^{\text{worst}}_{\text{test}}$ is very different from that of $\mathscr{D}^{\text{worst}}_{\text{train}}$, just like in Figure 2.10. $\mathscr{D}^{\text{worst}}_{\text{test}}$ is just one cluster of samples from all different CoralNet labels and it does not contain any cluster of specific CoralNet labels.



(a) Random split

(b) Worst split

Figure 2.12: **CoralNet label hypothesis.** Pie charts for $\mathscr{D}^{\text{random}}$ (a) and $\mathscr{D}^{\text{worst}}$ (b) comparing the ratio of samples from CoralNet labels represented in the splits. The ratio for the classes in $\mathscr{D}^{\text{random}}_{\text{train}}$ and $\mathscr{D}^{\text{random}}_{\text{test}}$ is similar whereas that of $\mathscr{D}^{\text{worst}}$ the ratio of $\mathscr{D}^{\text{worst}}_{\text{train}}$ is also quite balanced yet that of $\mathscr{D}^{\text{worst}}_{\text{test}}$ is very unbalanced. The difference between the distributions of $\mathscr{D}^{\text{random}}$ compared to that of $\mathscr{D}^{\text{worst}}$ indicates that the Splitter of $\mathscr{D}^{\text{worst}}$ learned to find a split where the CoralNet label of image patches helps to fool the Predictor, thus forming a bias in the dataset.



Average saturation histograms

Average value histograms

Figure 2.13: **Saturation and Value hypothesis.** Histograms for saturation (a) and value (b) for $\mathscr{D}^{\text{random}}$ and $\mathscr{D}^{\text{worst}}$. The saturation and value distributions of $\mathscr{D}^{\text{random}}$ and $\mathscr{D}^{\text{worst}}$ are very similar. This indicates that the saturation nor the value of the images form a bias in the CCBB dataset.

Bibliography

[1] Coral reefs facts.

[2] The ocean: Life and livelihoods.

[3] Coral reef ecosystems, Feb 2019.

[4] Laura Arenschield, Nov 2020.

[5] Yujia Bao and Regina Barzilay. Learning to split for automatic bias detection. *NeurIPS*, 2022.

[6] Oscar Beijbom. *Automated annotation of coral reef survey images*. University of California, San Diego, 2015.

[7] Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1170–1177. IEEE, 2012.

[8] Oscar Beijbom, Peter J Edmunds, Chris Roelfsema, Jennifer Smith, David I Kline, Benjamin P Neal, Matthew J Dunlap, Vincent Moriarty, Tung-Yung Fan, Chih-Jui Tan, et al. Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation. *PloS one*, 10(7):e0130312, 2015.

[9] Oscar Beijbom, Tali Treibitz, David I Kline, Gal Eyal, Adi Khen, Benjamin Neal, Yossi Loya, B Greg Mitchell, and David Kriegman. Improving automated annotation of benthic survey images using wide-band fluorescence. *Scientific reports*, 6(1):1–11, 2016.

[10] Elena Bollati, Cecilia D'Angelo, David I Kline, B Greg Mitchell, and Jörg Wiedenmann. Development of a multi-excitation fluorescence (mef) imaging method to improve the information content of benthic coral reef surveys. *Coral Reefs*, 40(6):1831–1847, 2021.

[11] Qimin Chen, Oscar Beijbom, Stephen Chan, Jessica Bouwmeester, and David Kriegman. A new deep learning engine for coralnet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3702, 2021.

[12] Arjun Chennu, Paul Färber, Glenn De'ath, Dirk de Beer, and Katharina E Fabricius. A diver-operated hyperspectral imaging and topographic surveying system for automated mapping of benthic habitats. *Scientific reports*, 7(1):1–12, 2017.

[13] Steven L Coles and Barbara E Brown. Coral bleaching—capacity for acclimatization and adaptation. 2003.

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

[15] Wolf-Christian Dullo. Coral growth and reef growth: a brief review. *Facies*, 51(1):33–48, 2005.

[16] C Mark Eakin, Hugh PA Sweatman, and Russel E Brainard. The 2014–2017 global-scale coral bleaching event: insights and impacts. *Coral Reefs*, 38(4):539–545, 2019.

[17] Redouane Es-sadaoui, Imad El Bouazzaoui, Lahoucine Azergui, and Jamal Khallaayoune. Underwater image processing: Technical study and experiments. *vol*, 2:20–29, 2017.

[18] Scott Freeman. *Biological science*. Pearson, Boston, 2017.

[19] Anabel Gómez-Ríos, Siham Tabik, Julián Luengo, ASM Shihavuddin, and Francisco Herrera. Coral species identification with texture or structure images using a two-level classifier based on convolutional neural networks. *Knowledge-Based Systems*, 184:104891, 2019.

[20] Anabel Gómez-Ríos, Siham Tabik, Julián Luengo, ASM Shihavuddin, Bartosz Krawczyk, and Francisco Herrera. Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation. *Expert Systems with Applications*, 118:315–328, 2019.

[21] Manuel Gonzalez-Rivero, Oscar Beijbom, Alberto Rodriguez-Ramirez, Dominic EP Bryant, Anjani Ganase, Yeray Gonzalez-Marrero, Ana Herrera-Reveles, Emma V Kennedy, Catherine JS Kim, Sebastian Lopez-Marcano, et al. Monitoring of coral reefs using artificial intelligence: a feasible and cost-effective approach. *Remote Sensing*, 12(3):489, 2020.

[22] Manuel González-Rivero, Oscar Beijbom, Alberto Rodriguez-Ramirez, Tadzio Holtrop, Yeray González-Marrero, Anjani Ganase, Chris Roelfsema, Stuart Phinn, and Ove Hoegh-Guldberg. Scaling up ecological measurements of coral reefs using semi-automated field image collection and analysis. *Remote Sensing*, 8(1):30, 2016.

[23] Sarah M Hamylton. Mapping coral reef environments: A review of historical methods, recent advances and future opportunities. *Progress in Physical Geography*, 41(6):803–833, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Ove Hoegh-Guldberg. Climate change, coral bleaching and the future of the world's coral reefs. *Marine and freshwater research*, 50(8):839–866, 1999.

[26] OVE HOEGH-GULDBERG. Coral reefs in a century of rapid environmental change. *Symbiosis*, 2004.

[27] Terry P Hughes, Kristen D Anderson, Sean R Connolly, Scott F Heron, James T Kerry, Janice M Lough, Andrew H Baird, Julia K Baum, Michael L Berumen, Tom C Bridge, et al. Spatial and temporal patterns of mass bleaching of corals in the anthropocene. *Science*, 359(6371):80–83, 2018.

[28] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[29] Paul L Jokiel, Ku′ulei S Rodgers, Eric K Brown, Jean C Kenyon, Greta Aeby, William R Smith, and Fred Farrell. Comparison of methods used to estimate coral cover in the hawaiian islands. *PeerJ*, 3:e954, 2015.

[30] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[31] Risheng Liu, Xin Fan, Ming Zhu, Minjun Hou, and Zhongxuan Luo. Real-world underwater enhancement: Challenges, benchmarks, and solutions under natural light. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4861–4875, 2020.

[32] Alessandra Lumini, Loris Nanni, and Gianluca Maguolo. Deep learning for plankton and coral classification. *Applied Computing and Informatics*, 2020.

[33] Anne E Magurran. *Measuring biological diversity*. John Wiley & Sons, 2003.

[34] A Mahmood, M Bennamoun, Senjian An, F Sohel, F Boussaid, R Hovey, G Kendrick, and RB Fisher. Automatic annotation of coral reefs using deep learning. In *Oceans 2016 mts/IEEE monterey*, pages 1–5. IEEE, 2016.

[35] Ammar Mahmood, Mohammed Bennamoun, Senjian An, Ferdous Sohel, Farid Boussaid, Renae Hovey, Gary Kendrick, and Robert B Fisher. Coral classification with hybrid feature representations. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 519–523. IEEE, 2016.

[36] Ammar Mahmood, Mohammed Bennamoun, Senjian An, Ferdous A Sohel, Farid Boussaid, Renae Hovey, Gary A Kendrick, and Robert B Fisher. Deep image representations for coral image classification. *IEEE Journal of Oceanic Engineering*, 44(1):121–131, 2018.

[37] Ma Shiela Angeli Marcos, Laura David, Eileen Peñaflor, Victor Ticzon, and Maricor Soriano. Automated benthic counting of living and non-living components in ngedarrak reef, palau via subsurface underwater video. *Environmental monitoring and assessment*, 145(1):177–184, 2008.

[38] Ma Shiela Angeli C Marcos, Maricor N Soriano, and Caesar A Saloma. Classification of coral reef images from underwater video using neural networks. *Optics express*, 13(22):8766–8771, 2005.

[39] Guilhem Marre, Cedric De Almeida Braga, Dino Ienco, Sandra Luque, Florian Holon, and Julie Deter. Deep convolutional neural networks to monitor coralligenous reefs: Operationalizing biodiversity and ecological assessment. *Ecological Informatics*, 59:101110, 2020.

[40] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[41] Anand Mehta, Eraldo Ribeiro, Jessica Gilner, and Robert van Woesik. Coral reef texture classification using support vector machines. In *VISAPP (2)*, pages 302–310, 2007.

[42] Md Modasshir, Alberto Quattrini Li, and Ioannis Rekleitis. Mdnet: Multi-patch dense network for coral classification. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–6. IEEE, 2018.

[43] R Ninio, Steven Delean, K Osborne, and H Sweatman. Estimating cover of benthic organisms from underwater video images: variability associated with multiple observers. *Marine Ecology Progress Series*, 265:107–116, 2003.

[44] M Pidwirny, 2006.

[45] Oscar Pizarro, Paul Rigby, Matthew Johnson-Roberson, Stefan B Williams, and Jamie Colquhoun. Towards image-based marine habitat classification. In *OCEANS 2008*, pages 1–7. IEEE, 2008.

[46] Alina Raphael, Zvy Dubinsky, David Iluz, Jennifer IC Benichou, and Nathan S Netanyahu. Deep neural network recognition of shallow water corals in the gulf of eilat (aqaba). *Scientific reports*, 10(1):1–11, 2020.

[47] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[48] Charles Sheppard, Simon Davy, Graham Pilling, and Nicholas Graham. *The biology of coral reefs*. Oxford University Press, 2017.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[50] M Dale Stokes and Grant B Deane. Automated processing of coral reef benthic images. *Limnology and Oceanography: Methods*, 7(2):157–168, 2009.

[51] Lei Sun. Resnet on tiny imagenet. *Submitted on*, 14, 2016.

[52] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[53] JEN Veron, MG Stafford-Smith, E Turak, and LM DeVantier. Corals of the world, version 0.01 beta. *Corals of the World*, 2020.

[54] Yan Wang, Wei Song, Giancarlo Fortino, Li-Zhe Qi, Wenqiang Zhang, and Antonio Liotta. An experimental-based review of image enhancement and image restoration methods for underwater imaging. *IEEE access*, 7:140233–140251, 2019.

[55] Ivor D Williams, Courtney S Couch, Oscar Beijbom, Thomas A Oliver, Bernardo Vargas-Angel, Brett D Schumacher, and Russell E Brainard. Leveraging automated image analysis tools to transform our capacity to assess status and trends of coral reefs. *Frontiers in Marine Science*, page 222, 2019.

[56] Lian Xu, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. Classification of corals in reflectance and fluorescence images using convolutional neural network representations. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1493–1497. IEEE, 2018.

[57] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.