ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
MASTER IN AUTOMATION ENGINEERING
M.Sc.A.

BY
Jean-Nicola BLANCHET

AUTOMATED TEXTURE-BASED RECOGNITION OF CORALS IN NATURAL SCENE IMAGES

MONTREAL, MARCH 7, 2016

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS:

M. Jaques-André Landry, memorandum director
Department of Automation Engineering, École de Technologie Supérieure

M. Éric Granger, committee president
Department of Automation Engineering, École de Technologie Supérieure

M. Matthew Toews, invited examiner
Department of Automation Engineering, École de Technologie Supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON FEBRUARY 17, 2016

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

This research was made possible through the financial support of École de Technologie Supérieure and the Natural Sciences and Engineering Research Council of Canada. Thanks to the Australian Institute of Marine Science for providing exceptionally high quality data on which this project relies. Thanks to my supervisor, Jacques-André Landry for all his collaborative work which made this project possible, for allowing me to work with great freedom and for financing the project.

Special thanks to my insightful friends, colleagues and professors: Alexandre Millette, Sébastien Déry, Marc-André Carbonneau, Isabelle Prévost, Christophe Pagano, Ouled Sghaier Moslem , Prof. Éric Granger, Prof. Luc Duong; as well as my supportive family.

AUTOMATED TEXTURE-BASED RECOGNITION OF CORALS IN NATURAL SCENE IMAGES

Jean-Nicola BLANCHET

ABSTRACT

Current coral reef health monitoring efforts rely on biodiversity data. Although cutting-edge imaging technology enables reliable and automatic collection of such data, simple RGB digital photography in combination with manual image annotation remains a popular solution. Unlike other acquisition methods, close range visible light imaging yields detailed species and surface coverage data, and requires cheaper equipment. Moreover, images acquired in the last few decades are limited to mere RGB photographs or analog VHS video, and contain important data for long term analysis. Unfortunately, manual expert labeling has become problematic due to the high volume of images and the lack of human resources available. Consequently, coral reef biodiversity data currently available is based mostly on small sample analysis. Previous automatic benthic image annotation systems have yielded unsatisfactory results compared to human performance for the same task. This is partly due to the high diversity of complex textures found in these images. We hypothesize that these complex textures require different features to be properly characterize. Motivated by the need for an improved automated benthic image annotation system, this work proposes a new approach based on a combination of multiple state-of-the art texture recognition methods. Firstly, methods to correct and enhance images will be investigated. Secondly, various state-of-the-art texture features will be used to overcome the texture diversity challenge: many statistical features, local binary patterns, textons, vector-quantized Scale-Invariant Feature Transform (SIFT) using the Improved Fisher Vector (IFV) method, Deep Convolutional Activation Feature (DeCAF), amongst others. Thirdly, a multi-classifier fusion method is proposed to efficiently aggregate the information from these multiple texture representations using a score-level fusion. Fourthly, rejection will be applied to further enhance accuracy. The results on the AIMS dataset (Australian Institute of Marine Science) and MLC2008 (Moorea Labeled Corals 2008) containing respectively 75 825 and 131 260 coral texture patches show that the proposed multi-classifier fusion method outperforms any other single method for the task of benthic image labeling.

Keywords: Coral reef, Natural scene, Image annotation, Labeling, Multi-classifier, Pooling, Features, Rejection

RECONNAISSANCE DE TEXTURE APPLIQUÉE À L'ANNOTATION AUTOMATISÉE DE CORAUX DANS DES IMAGES DE SCÈNES NATURELLES

Jean-Nicola BLANCHET

RÉSUMÉ

Les méthodes actuelles de suivi de la santé des récifs coralliens dépendent de données sur la biodiversité. Bien que les méthodes de pointes permettent une collecte automatisée et fiable de telles données, la simple photographie RVB en combinaison avec l'annotation manuelle des images reste une solution populaire. Contrairement aux autres méthodes d'acquisition, l'imagerie de près dans le spectre visible contient de l'information détaillée par rapport aux espèces et à leur occupation du substrat marin, en plus de rester la solution la moins couteuse. De plus, les images acquises dans les quelques dernières décennies sont limitées à de simples photographies RVB ou vidéos VHS analogues, et contiennent des données importantes pour les analyses historiques. Malheureusement, l'annotation manuelle par l'expert n'est plus viable due au volume élevé d'images acquises et à la faible disponibilité des ressources humaines. En conséquence, les données disponibles sur la biodiversité des récifs coralliens pour les dernières décennies sont généralement basées sur l'analyse de petits échantillons. Les systèmes d'annotation automatisés d'images benthiques proposés ont menés à des résultats insatisfaisants en comparaison à la performance de l'humain pour la même tâche. Ceci est dû en parti à la haute diversité de textures complexes trouvées dans ces images. On pose l'hypothèse que ces textures complexes nécessitent différentes caractéristiques afin d'être bien représentés. Motivé par le besoin d'un système amélioré d'annotation automatisé d'images benthiques, ce travail propose une nouvelle approche basée sur une combinaison de plusieurs méthodes de pointes utilisées dans le domaine de la reconnaissance de textures. Premièrement, la correction et l'amélioration de l'image seront explorées. Deuxièmement, diverses caractéristiques issues de la littérature de pointe seront mises à l'essai : des caractéristiques statistiques, les motifs binaires locaux, les textons, le descripteur SIFT quantifié via IFV (vecteur de Fisher amélioré) et le descripteur DeCAF (caractéristiques d'activation convolutionnelle profonde), parmis tout d'autres. Troisièmement, une méthode de fusion via de multiple séparateurs à vaste marge (SVM) faisant l'agrégation de l'information provenant de ces multiples représentations de textures sera proposée via une fusion par score. Quatrièmement, un seuil de rejet sera appliqué pour améliorer davantage les performances. Les résultats sur la base de données AIMS (Australian Institute of Marine Science) et MLC2008 (Moorea Labeled Corals 2008) contenant respectivement 75 825 et 131 260 points annotés démontre que la méthode de fusion multi-classifieur proposée performe mieux que tout autre approche basée sur un seul ensemble de caractéristiques pour la tâche d'annotation automatisé d'images benthiques.

Mots clés: Récifs coralliens, Scène naturelle, Annotation d'images, Étiquettes, Approche multi-classifieur, Caractéristiques, Rejet

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

AIMS	Australian Institute of Marine Science
CLAHS	Contrast Limited Adaptive Histogram Specification
CLBP	Completed Local Binary Pattern
CNN	Convolutional Neural Network
DTD	Describable Texture Dataset
DTD_{RBF}^{IFV}	Describable Texture Dataset Descriptor
FA	False Acceptance
FAR	False Acceptance Rate
FR	False Rejection
FRR	False Rejection Rate
GLCM	Grey Level Cooccurrence Matrix
GMM	Gaussian Mixture Model
IFV	Improved Fisher Vector
LBP	Local Binary Pattern
MCF	Multi-classifier Fusion
MLC	Moorea Labeled Corals
PCA	Principal Component Analysis
RBF	Radial Basis Function
RGB	Red Green Blue

ROI	Region of Interest
SVM	Support Vector Machine
TA	True Acceptance
TAR	True Acceptance Rate
TR	True Rejection
TRR	True Rejection Rate
UV	Ultraviolet
VHS	Video Home System

LISTE OF SYMBOLS AND UNITS OF MEASUREMENTS

cm	centimeters
mm	millimeters
px	pixel
$\mathbb{R}^N$	n-dimensional Euclidean space
s	second

INTRODUCTION

Coral reef across the world are currently facing great challenges that threaten their survival. Global warming, ocean acidification, and human coastal activities are all factors contributing to the decay of coral reef biodiversity over time. As reported by Hoegh-Guldberg *et al.* (2007), this is not without negative impact on human quality of life. Coral reefs offer services that are fundamental to many societies across the world, such as providing support to the fishing and tourism industries. This explains the emerging necessity for the scientific community to study the rapidly evolving health state of coral reefs and to investigate potential solutions to their long term preservation.

To study the evolving health state of coral reef, biodiversity data on the prevalence of algae and coral species at various point in time needs to be gathered and compiled. A wide array of modern acquisition technologies exists for gathering such data, and most can be combined with computer vision techniques to automate the biodiversity data extraction process. For instance, some methods take advantage of the fluorescence property of coral colonies, making them easy to locate underwater with a proper ultraviolet source. As shown in figure 0.1, the approach is demonstrated by Sasano *et al.* (2013), and allows for accurate distinction between six classes: live branch type coral, live table type coral, dead coral skeletons, sand, sea grass and algae. Similarly, multispectral imaging is used by Gleason *et al.* (2007) to distinguish three classes: algae, coral, background. They conclude that careful selection of narrow bands, in combination with texture information improves automated substrate mapping accuracy. Stereo vision is another popular approach that integrates three dimensional depth information for improved recognition accuracy, as demonstrated by Johnson-Roberson *et al.* (2006). Other methods based on satellite or aerial remote sensing exist, as presented by Roelfsema *et al.* (2002); Lesser and Mobley (2007); Lim *et al.* (2009).

While most of these imaging methods yield sufficiently accurate biodiversity data for research purposes within the marine ecology scientific community, they have three major drawbacks:

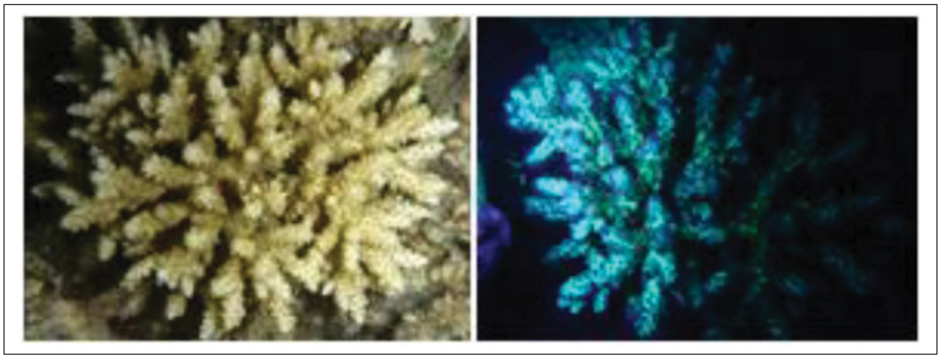


Figure 0.1 Coral response to visible (left) and 360-385 nm ultraviolet (right) light, observed in the visible spectrum. Adapted from Sasano *et al.* (2013).

a. They require access to expensive acquisition equipment, which is out of reach for smaller scientific groups.

b. They are able to distinguish only a limited number of unique classes (usually between three and nine). The number of classes depends on the taxonomic level used. A biological taxonomy contains many rankings useful to categorize organisms with different levels of detail. For research purpose, a finer ranking (*e.g.* species) as opposed to a coarser ranking (*e.g.* family) is desirable because it is useful for more specific research problematics, such as identifying the more vulnerable coral species in a given area.

c. They cannot be applied retroactively to benthic photographs acquired in the last decades, since the acquisition technology was unavailable. These are limited to mere RGB digital or analog photography, and analog VHS videos. However, these images contain important information for long-term studies.

For these reasons, simple underwater close range RGB photography remains a popular acquisition method, even if automatic annotation of these images remains an open problem. In order to extract biodiversity data from these images, researchers have adopted manual annotation protocols allowing image content sampling. For example, the Australian Institute of Marine Science (AIMS) have defined a strict protocol, as detailed by Jonker *et al.* (2008b), allowing multiple experts to collaboratively label the content of several images as objectively as possible. While manual expert annotation does produce biodiversity data, it has two main limitations. Firstly, manual expert annotation is extremely time consuming and given the available resources, it cannot be applied to the large volume of existing benthic images from the past decades. Secondly, manual expert annotation is a difficult error prone task which may lead to inconsistent results. A study presented by Ninio *et al.* (2003) provides statistics on the frequency at which multiples experts disagree in analog video labeling. In this study, experts are asked to label images at five different taxonomic ranks: group, life form, family, genus, species. The results, as presented in figure 0.2, show that experts disagree approximately on one out of ten labels. These results suggest that the annotation task is difficult to perform objectively. Results for a similar problem, marine microorganisms classification, are presented by Culverhouse *et al.* (2003). The author names short-term memory limitation; fatigue and boredom; recency effects (bias towards recently used labels); and positivity bias (bias towards one's expectations) as the causes for expert errors. We argue that methodological ambiguities also exist as source of errors at the annotation protocol level. These challenges will be explored in depth in chapter 2 on data.

This confirms the need for an automated annotation system capable of operating with little to no human intervention. Previous work, which will be reported in chapter 1, has attempted to address the problem of automatic benthic image annotation using texture recognition techniques, but the reported accuracy remains unsatisfactory compared to human performance for the same task. We estimate that an accuracy of roughly 80% would yield results useful to the scientific community, and 90% is close to the human expert performance. Any elevated level of performance above human accuracy would be insignificant, since these manual labels and their errors are used as a ground truth to design and measure the accuracy of automated systems.

In this thesis we will start by conducting a review of promising texture recognition techniques within the computer vision framework. Many of these cutting-edge techniques have been applied successfully to challenging natural scene problems and some even to benthic image labeling. The selected relevant techniques will then be applied to two distinct coral reef image datasets, as well as to standard texture recognition benchmark datasets to gain a better under-

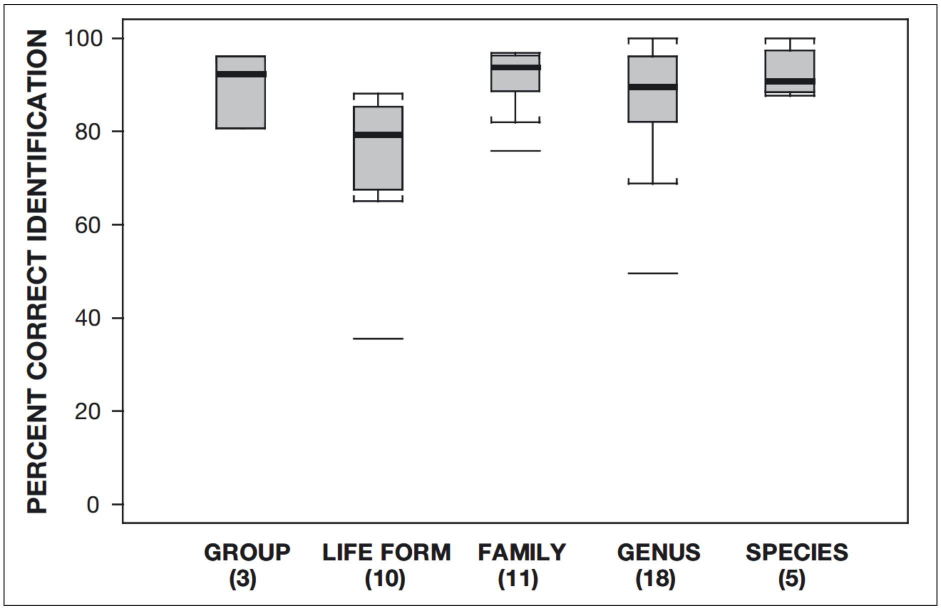


Figure 0.2 Disagreement between experts in manual annotation at five taxonomic ranks. The number of classes is shown in parenthesis. Taken from Ninio *et al.* (2003)

standing of their capabilities and limitations. Finally, a short conclusion will discuss potential applications of the proposed system.

CHAPTER 1

LITERATURE REVIEW

Benthic data acquired in the last few decades is limited to digital or analog photographs. Because no additionnal hyperspectral or depth information is available, the task of automatic annotation must be performed using RGB data. This points towards texture recognition methods from the field of computer vision known to be well suited for RGB data. In this chapter, we first present a generally accepted computer vision framework that will allow the subdivision of the problem into a set of smaller and well defined problems. We then explore, for each subproblem, methods that were applied to benthic image annotation as well as other similar texture recognition application in natural scene images.

1.1 Computer Vision Framework

Computer vision, as defined by Gonzalez *et al.* (2004), is a branch of artificial intelligence "whose ultimate goal is to use computers to emulate human vision, including learning and being able to make inferences and take actions based on visual inputs." This includes the problem of texture recognition in natural scenes. A typical computer vision classification system can be broken down into a pipeline of successive algorithms each dealing with one specific aspect of the problem. Because every problem is unique, the structure of the system may change significantly from one problem to another. However, as presented in figure 1.1, we settle on a fairly general definition of this pipeline that can be applied to many recognition problems from a wide array of fields.

The pipeline presented in figure 1.1 separates computer vision processing steps into two large groups of subproblems. The first one is associated with the field of image processing and deals with images and pixels representing the real world as seen by humans. At this stage, operations are performed to acquire, enhance, restore, segment and describe the image. The output is an abstract quantitative, yet highly descriptive representation of the observed object. The second group is associated with the field of pattern recognition, and deals with these much

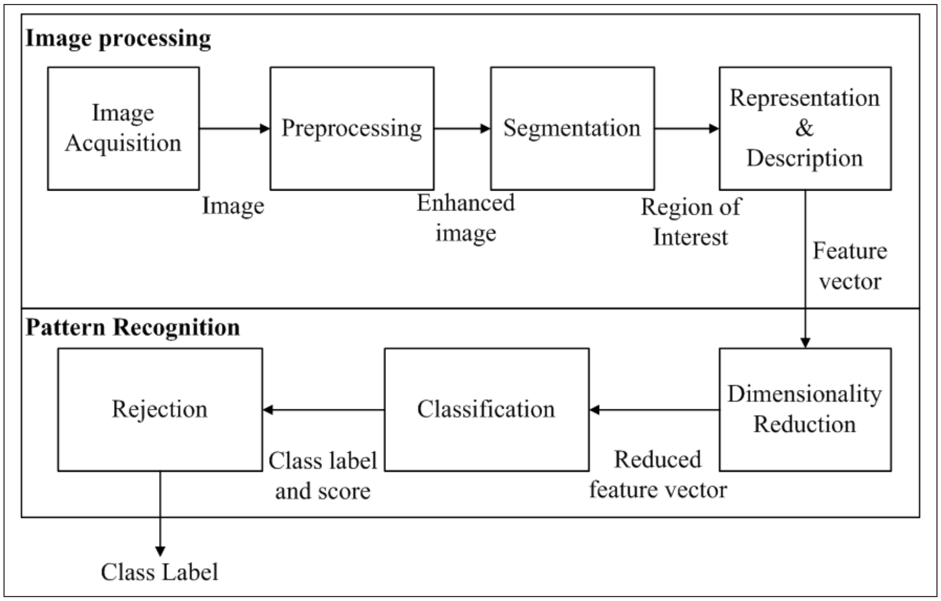


Figure 1.1 A pipeline for a simple computer vision recognition system. Based on Gonzalez *et al.* (2004); Szeliski (2010); Duda *et al.* (2012); Bernd (2002)

more abstract entities. The steps of dimensionality reduction, classification and rejection aim to output a correct decision given an input signal describing an observation. The following list presents a short definition of each one of these seven steps and introduces their roles in the context of benthic image annotation:

a. **Image acquisition** aims to convert energy waves (or particles) into a digital image. While visible light is a popular option, this definition can extend to any wavelength, and even to non electromagnetic sources, like ultrasounds, or beams of accelerated electrons. As it is true for all steps of this pipeline, high quality processing using appropriate methods and parameters is important to support the following processing steps. Bernd (2002); Gonzalez *et al.* (2004) explain some of these parameters: selection of an appropriate light source, sensor technology, wavelength, lens, illumination strategy, etc. While several studies have demonstrated the performance of various acquisition system specifically designed for benthic image annotation, the problem being studied does not allow any control over these parameters, as image acquisition has already been performed. It is therefore necessary to deal with the various challenges that come with the benthic image datasets through other means. These challenges will be explored in depth in the following chapter.

b. **Preprocessing** aims to *restore* the image by correcting acquisition artifacts such as noise, or *enhance* the image for further processing. This can include a wide array of image processing methods, such as intensity transformation, spatial or frequency filtering, geometric transformations, multi-resolution decomposition, lens distortion correction, etc. For simplicity, we'll refer to these as *image filtering methods*. Depending on the dataset, benthic images have specific acquisition flaws that can be addressed at the preprocessing level.

c. **Segmentation** is the process of finding and isolating one or more regions of interest (ROI) in the image. This step allow the system to focus on important objects without considering the irrelevant background information. Various segmentation methods exist depending on the complexity of the problem. For difficult problems, the segmentation step may be an entire computer vision processing pipeline including some recognition steps, like it is the case for face detection in biometric facial identification systems.

d. **Representation & description** is often referred to as descriptor extraction or feature extraction. This process takes a set of pixels representing an object of interest and attempts to extract a set of meaningful measures in the form of a feature vector that will allow mathematical models to manipulate the data and find patterns in the following steps. Because ROIs are defined by a set of many pixels, the colossal amount of information they contain usually makes them impractical to manipulate directly, as the smaller bits of meaningful information are diluted and hidden. While humans excel at inferring these patterns from few examples, it is very hard for a machine to find them, hence the necessity of feature extraction. Some modern methods like deep convolution neural networks, or sparse coding can be used to find these relevant patterns inside complex data, but these methods are considered beyond the scope of this work.

e. **Dimensionality reduction** is an optional step that takes a large feature vector, and increases its level of abstraction by further reducing its information. Because the previous step of representation & description has a similar goal, both steps are sometimes considered to be the same. However, because they are not mutually exclusive, we consider it to be a separate, but optional step. In some problems where there is a strong spatial-intensity relationship between pixels, like in object recognition, it is possible to use simple dimensionality reduction methods directly on the region of interest to extract meaningful features. Because it is not the case for texture recognition in natural scenes, dimensionality reduction is considered a separate step.

f. **Classification** can be defined as the statistical inference of the class associated with a given observation (or instance). Classification uses a mathematical or heuristic model previously trained on many labeled examples of the expected output given a specific input. A large variety of classification schemes exist, each having their own capabilities and limitations. But as stated by Wolpert and Macready (1997) in their famous *No Free Lunch Theorem*, "[...] for any [classification] algorithm, any elevated performance over one class of problems is offset by performance over another class", as different classes of problems have mutually exclusive properties. We focus our study on classification algorithms that have performed well in texture recognition problems as well as betnhic image annotation.

g. **Rejection** is another optional step that can further improve the reliability of the system. Some classifiers can be trained to output a score, or a certainty metric along with their class prediction. This score can be used to threshold acceptance of the prediction. Given a low enough score, the system can decide to ignore a particular sample reducing the misclassification frequency.

In the remaining sections of this chapter, we explore previous work relevant to each subproblem for the task of benthic image annotation.

1.2 Preprocessing

As previously defined, preprocessing includes every image filtering method aiming to *restore* or *enhance* image properties. Because underwater image acquisition is known to be difficult, much work has been done to study the quality loss of underwater photographs, and the applicable restoration methods. There are several physical phenomenons that contribute to degrading the quality of these images. Arnold-Bos *et al.* (2005) presents a short list of those factors:

- *A ray of light is exponentially attenuated as it travels in the water so the background of the scene will be poorly contrasted and hazy.*
- *Water will reflect a significant fraction of the light power towards the camera before it actually reaches the objects in the scene[, causing] [...] a characteristic glowing veil that super imposes itself on the image and hides the scene. [This effect is known as backscattering, and is mostly true with flash or strobe photography.]*
- *Macroscopic floating particles ("marine snow") [appear as bright white noisy dots. Again, this applies to flash or strobe photography].*

Bazeille *et al.* (2006) expands this list by adding the following two factors:

- *Floating particles highly variable in kind and concentration, increase absorption [and] blur [the] image features. [This is known as forward scattering].*
- *The non stability of the underwater vehicle [affects contrast, especially at low shutter speeds].*

The author then proceeds to summarize the observed effects of all these factors combined: "limited range, non uniform lighting, low contrast, color diminished (the wavelength corresponding to the red color disappears after only a few meters), important blur".

Because there are many different sources of artifacts in the underwater environment, having a strong understanding of the acquisition conditions is required to appropriately restore the image. Schettini and Corchs (2010) support this idea in their review article containing a thorough list of previous work on the matter. It becomes apparent that every method poses a set of characteristic assumptions on the acquisition conditions. Interestingly, because coral reef surveys are performed at most a few meters away from the marine substrate, and because the acquisition protocols enforce a camera angle perpendicular to the surface of the sea floor, some of the previously introduced artifacts and flaws may be uncommon or even never seen in bethic photograph, such as the limited range from the haze effect, as well as backward scatter caused by organic particles. Consequently it is difficult to evaluate the relevance of each method given the variable frequency and magnitude of the observed degradation effects. This also explains why previous work on benthic image annotation has often addressed the problem of image restoration in a more intuitive and empirical approach.

Pizarro *et al.* (2008) identify non-uniform lighting, backscatter and wavelength-dependent attenuation as problems. They propose using the "Comprehensive Normalization" method, which recursively normalizes RGB triplet magnitudes and channel lengths, as originally described by Finlayson *et al.* (1998) until convergence. While this methods was not originally developed for underwater images, it reportedly addresses the three previously identified problems. The recursive algorithm is defined as follow:

Given an N-pixel RGB image reshaped into a Nx3 matrix I, successive applications of the row-normalization R and column-normalization C can be applied recursively:

$$I_i = C(R(I_{i-1})) \tag{1.1}$$

where the row-normalization of image I is:

$$R(I)_{i,j} = \frac{I_{i,j}}{\sum\limits_{k=1}^{3} I_{i,k}} \tag{1.2}$$

and the column-normalization of image I is:

$$C(I)_{i,j} = \frac{N/3 I_{i,j}}{\sum\limits_{k=1}^{N} I_{j,k}} \tag{1.3}$$

In their work specific to benthic image annotation, Shihavuddin *et al.* (2013) take a different approach and attempt to empirically find a set of image restoration and image enhancement algorithms that yield more descriptive features and improve the global performance of the system for a given dataset. The algorithms tested are the following:

- **Color correction**: using color marker references (if available), colors are partially restored.
- **Contrast limited adaptive histogram specification (CLAHS)**: enhances the local contrast.
- **Comprehensive image normalization method**: as previously defined.
- **Color channel stretching**: the 1.5 and 98.5 percentile values are linearly stretched to the minimum and maximum values.

While this approach can easily be used to maximize the global performance of the system, it causes several issues. To avoid methodological bias, the set of preprocessing algorithms should be selected on a separate validation sample, which is an extremely time consuming process. Furthermore, because this selection process has little physical basis, the resulting selection is somewhat arbitrary, and may not generalize to other reefs in the same dataset, resulting in the necessity to reapply the filter selection on every reef. Finally, preprocessing for image restoration or enhancement is an lossy process: because values are changed and rounded, the amount of information in the resulting image is always equal or less. Successive application of many image filtering methods can therefore be problematic.

Work by Beijbom *et al.* (2012) present a similar approach, but search for a single preprocess method that tends to generalize better over different reefs. They identify the "color channel stretch" as single simple solution. While still empirical, this approach is in line with famous *Occam's razor* principle, that states that when multiple solutions compete for a same problem, the simplest one will tend to generalize better.

Likewise, Bouchard (2011) presents a system in which white balance followed by color channels stretch are systematically applied. This aims to enhance colors and contrasts to produce better texture features. White balance requires a white color marker, and is defined by equation 1.4. Given the intensity of the white color marker (R_w, G_w, B_w), the input intensity (R, G, B), and assuming a maximum intensity value of 255, the output intensity of the white balance correction is given by:

$$\begin{bmatrix} R' & G' & B' \end{bmatrix} = \begin{bmatrix} R\frac{255}{R_w} & G\frac{255}{G_w} & B\frac{255}{B_w} \end{bmatrix} \tag{1.4}$$

Despite the work done by these authors, only generic image enhancement algorithms have been applied to automated benthic image annotation, and popular image restoration algorithms specifically designed for underwater imagery have yet to be tested. While some of these methods require well parameterized deconvolutional filters and are quite challenging to implement correctly, much work has been done to develop parameterless methods applicable to images from different sources. Arnold-Bos *et al.* (2005) introduce the idea that simple algorithms can deal with many of the challenges of underwater image acquisition, and compete with advanced methods. They propose a set of two simple algorithms: contrast equalization, and self-tuning wavelet-based algorithm. Bazeille *et al.* (2006) propose using a pipeline of nine filters to overcome these problems: spectral peak thresholding (Fourier domain), mirror padding to obtain a square image, color space conversion from RGB to YCbCr, spectral homomorphic filtering, wavelet denoising, anisotropic filtering, intensity stretching, rgb color space conversion (and image cropping), mean equalization. Their method requires no parameters and is reportedly applicable to all underwater images. Carlevaris-Bianco *et al.* (2010) propose a method for dehazing based on the difference in attenuation between the three image color channels, which also generates a depth map as a by-product. Such a map could hold precious descriptive information usable by feature detection, similar to what stereovision would achieve. However, this requires hazy images. In more recent years, Prabhakar and Kumar (2012) have proposed using a combination of four filters: homomorphic filtering, wavelet denoising, bilateral filter and contrast equalization. The study shows that the proposed method improves feature detection.

1.3 Segmentation

Segmentation is the process of partitioning pixels of the image into contiguous sets representing unique objects. The resulting sets of pixels are called regions of interest (ROI). For the problem of texture recognition in natural scenes, segmentation could be used to isolate a homogeneous textured region, removing irrelevant background information and making it easier to describe the texture. In many computer vision problems, reliable segmentation can be performed by posing *a priori* assumptions about the scene. However, because coral colonies may take various shapes, sizes and colors, because illumination conditions may change from an image to another, and because the image quality is relatively variable, no simple assumptions can be made about the scene, which adds complexity to the task of segmentation. Todorovic and Ahuja (2009) have studied "texels", which are the smallest repeated elements in the pattern, and have explained the problem as follow: "Texels in natural scenes, in general, are not identical, and their spatial repetition is not strictly similar to one another, and their placement along a surface is only statistically uniform." This problem was encountered and described by Bouchard (2011) who investigated simple segmentation methods, such as watershed, and concluded these methods fail to adapt to the complexity of textures in underwater natural scenes.

Despite these difficulties, previous work has attempted to integrate various segmentation methods to benthic image annotation. These more advanced methods treat texture segmentation as a two step process: (1) pixels-wise (local) texture feature extraction, and (2) clustering or classification. Donate (2006) used Gabor wavelet response (which will be discussed in section 1.4) as a texture feature, in combination with k-means and expectation-maximization as clustering algorithms. While the examples of resulting segmentation are of remarkable quality, all these method requires fine tuning of at least one parameter for each new image. It would however be interesting to experiment further with adaptive clustering algorithms. Supervised approaches have also been proposed. Tusa *et al.* (2014) present an approach based on supervised learning. Using once again Gabor Wavelet response as a texture feature, small patches of the image are extracted and classified in two groups: coral and non-coral. This results in a label map that can be used as a set of segmented region. Though these approach are promising, it has not been demonstrated that they can be applied to large datasets of natural scenes.

In the absence of a viable solution to the problem of segmentation in benthic image annotation, authors have turned to a naive approach often used in texture recognition: a fixed size patch (or window) around the target pixel acts as a region of interest. While simple, this approach is problematic for obvious reasons: it does not adapt well to arbitrarily shaped organic content, and the resulting patch may contain multiple textures. Figure 1.2 presents an example from the AIMS dataset (Jonker *et al.*, 2008a) of this frequent problem with a square patches. Consequently, the following step of feature extraction yields lower quality features. Beijbom *et al.* (2012) recognized this problem and proposed a simple solution. In their system, four patches of unique size are extracted around the target point. Features detection is performed at four scales and resulting feature vectors are concatenated.

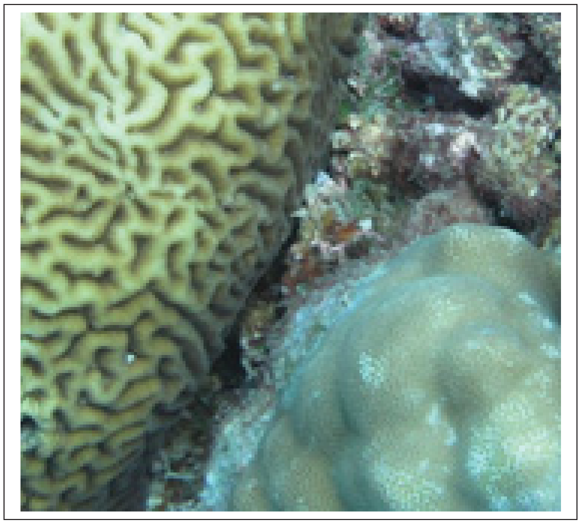


Figure 1.2 A patch from the AIMS dataset around a point of interest containing more than one texture.

While it is likely that proper image segmentation increases the performance of the system, its significance is unknown. We recently studied the matter in unpublished work. Appendix I presents the confusion matrices of the classification results on a subsets of approximately 1565 images from the MLC dataset (see 3.1). The first experiment was performed using fixed-size patch, while the second one used non-expert manually selected homogenous rectangle regions around the specified point. On this sample, average global classification rate over a 10-fold analysis showed an improvement from 63% to 76%. This suggest that there is a significant gain associated with the introduction of proper segmentation.

Although these results are promising, ambiguities with the manual annotation protocol cause disagreements between segmented regions and the ground truth, and have made segmentation very difficult to implement despite our efforts. These challenges will be further discussed in the following section. For this reason, segmentation is considered beyond the scope of this work, and will be addressed in future work. The proposed system will therefore disregard segmentation, and make use of simple fixed-size and multi-scale patches.

1.4 Representation & Description

In machine learning, features are key measurements extracted from an observation that describe an instance of a pattern (or an object of a class). As introduced previously, the processing pipeline behind a typical pattern recognition system within the computer vision framework is an information reduction process that turns a complex pixel map into a compact class prediction (a single bit for a two-class problem). As a result, the goal behind representation and description is to further reduce that information by providing a lower dimensionality representation of the object. This is why representation and description is also called feature extraction, which is sometimes associated with the separate, but not mutually exclusive dimensionality reduction step.

Gonzalez *et al.* (2004) and Duda *et al.* (2012) consider *representation* to be the process of encoding an instance of a pattern in the form of an appropriate data structure. For instance, a shape's border can be encoded as a string of symbols representing the border orientation. This is called a structural representation. On the other hand, *description* is associated with methods that compute descriptors, which measure a set number of texture, shape, color or intensity characteristics. If n measurements are taken from an object, they can be represented in a $n \times 1$ matrix, or, as a n-dimension vector. The description step takes an ROI in the image, and is said to output an n-dimension feature vector in Euclidean space $\mathbb{R}^N$ (also called feature space). Chosen descriptors are problem-specific, and target meaningful information in objects, making it possible to associate a new instance to its respective class. In the case of texture recognition, *description* methods are much more popular than *representation* methods.

Typically, "representation and description" is considered a mandatory step in computer vision classification problems, but newer methods have been developed to automatically find and extract the meaningful features. In fact, deep learning (convolutionnal neural networks) and sparse coding methods do not require this step, as they are able to use the raw input data directly. These modern methods are relevant for future work, but are considered beyond the scope of this work.

A large variety of texture descriptors have been proposed over the years, and the most popular ones seem promising for the problem of benthic image annotation. The following sections explores various descriptors that have been applied successfully to either coral photographs, or textures in natural scenes.

In the first few subsections, we discuss global features proposed by Bouchard (2011) as well as Shihavuddin *et al.* (2013), who present similar approaches. They propose computing a set of many popular global texture descriptors and combining them into a single large vector. These features include the following: local binary patterns (LBP), gray level cooccurrence matrix (GLCM), Gabor filters, channel histogram statistics, hue and opponent angle histograms.

1.4.1 Local Binary Patterns

Local binary patterns (LBP) originally described by Ojala *et al.* (2002) are powerful texture descriptors that have seen many applications in a variety of fields including biometrics and biomedical. LBP associates a pattern code to each pixel in the image, and create a histogram of the observed codes, serving as a feature vector. Equation 1.5 presents a simple mathematical definition of the LBP code.

Given a grey level image I, the LBP value of pixel *x* with sampled neighbors *S* (implicitly taken along a *d* pixel radius circle around *x* at a frequency *s*) is given by:

$$LBP(x,S) = \sum^{i \in S} H(I_x - I_i) \cdot 2^{i-1} \tag{1.5}$$

where H represents the discrete Heaviside function:

$$H(x) \begin{cases} 0, x < 0 \\ 1, x > 0 \end{cases} \tag{1.6}$$

Alternatively, it may be significantly more intuitive (and computationally efficient) to approach LBP as a binary algorithm, where sampled pixels S in the neighborhood of the pixel x in the image I are aggregated into a binary code:

```
**Algorithm 1.4.1:** LBP($I, x, S$)

  $code \leftarrow 0$
  **for each** $i \in S$
    **do** { $code \leftarrow code << 1$
           $code \leftarrow code \ OR \ (I_x > I_i)$
  **return** $(code)$
```

Ojala *et al.* (2002) explain that LBP can easily be adapted to become invariant to orientation, which is important in natural texture analysis. By consolidating rotated variations of binary codes, one can obtain identical results on rotated textures. For instance, if binary code 0001 was obtained for a pixel, it should be treated as part of the same pattern group as 0010, 0100, and 1000. This original LBP formulation was applied by Bouchard (2011) to benthic image annotation.

However, many versions of LBP have been proposed throughout the years. Here is a non-exhaustive list presenting a few variations:

- **Red green blue LBP** is extracted on each color channel.
- **Local gabor LBP** (Zhang *et al.*, 2005) is extracted on Gabor filter responses, and resulting histograms are concatenated.
- **Center-symmetric LBP** (Heikkilä *et al.*, 2006) combines another popular feature, SIFT, with the original LBP (see appendix II for more information on SIFT).
- **Multi-bloc LBP** (Zhang *et al.*, 2007) divides the image into blocs, and computes a unique histogram on each bloc. The resulting histograms are concatenated.
- **Volume LBP** (Zhao and Pietikainen, 2007) looks at time-wise neighbors in a video sequence.
- **Transition LBP** (Trefnỳ and Matas, 2010) uses the previous neighbor pixel instead of the center pixel as a reference.
- **Direction coded LBP** (Trefnỳ and Matas, 2010) adds additional gradient information in part of the LBP code.
- **Dominant LBP** (Liao *et al.*, 2009) considers only the most frequent pattern.
- **Completed LBP** (Guo *et al.*, 2010) adds additional information on the relative intensity of the center pixel and the magnitude of change.
- **Extended LBP** (Liu *et al.*, 2012) generates a code from four features: intensity of the central pixel, intensity of neighbors, radial-difference and angular difference.

Completed local binary pattern (CLBP) proposed by Guo *et al.* (2010) is among the most popular. CLBP was used by Shihavuddin *et al.* (2013) in benthic image annotation. The algorithm generates three separate feature maps:

- **The center pixel gray intensity map (LBP_C)** is a binary map obtained by thresholding the gray intensity (e.g. by using the mean intensity as a threshold).
- **The sign map (LBP_S)** is identical to the original LBP definition.
- **The magnitude map (LBP_M)** is generated by computing the difference between the center reference pixel and the neighboring pixels, and converting the result to binary format by using an adaptive threshold (e.g. the mean magnitude in the neighborhood). The resulting bits are concatenated as they are in the original LBP formulation.

The authors in Guo *et al.* (2010) propose two fusion methods applicable to any combinations of features maps that aim to integrate information from all three maps into a single feature vector. The first method is to compute and concatenate separate histograms. This was the method of choice in the work presented by Shihavuddin *et al.* (2013) on benthic image annotation, who concatenated the sign and magnitude maps. The second option is to generate a higher dimensionality histogram (2d or 3d), which reportedly outperform the first fusion method, but creates a much higher dimension feature vector. The author recommend using a simpler method, thus creating a much smaller histogram at the cost of a slight feature quality loss: the magnitude (LBP_M) and center (LBP_C) intensity map form a 2d joint histogram, and the result is concatenated with the sign map (LBP_S).

1.4.2 Grey Level Cooccurrence matrix

The grey level co-occurrence matrix (GLCM) was shown to yield great results in various fields, such as the biomedical field. It was applied to benthic image classification by both Bouchard (2011) as well as Shihavuddin *et al.* (2013). The method was originaly proposed by Haralick *et al.* (1973) and has been improved by many contributions over the years.

The GLCM observes the frequency at which every combination of intensity values are found as neighbors in the texture sample, resulting in a large $L \times L$ matrix, where $L-1$ is the maximum intensity value in a discrete range. This matrix can be visualized, and is quite descriptive of the texture. Figure 1.3 and 1.4 show an example of two different texture samples being compared: the blotchy texture presents a much larger variety of coocurrences than the structured texture, which presents higher peaks in smaller areas.

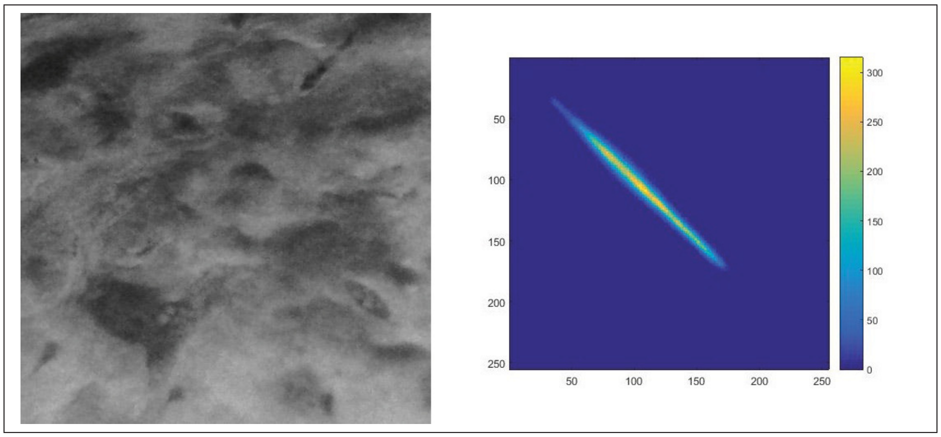


Figure 1.3 A blotchy texture and its GLCM. The transitions are smooth in the image, and are distributed over a large range in the GLCM.

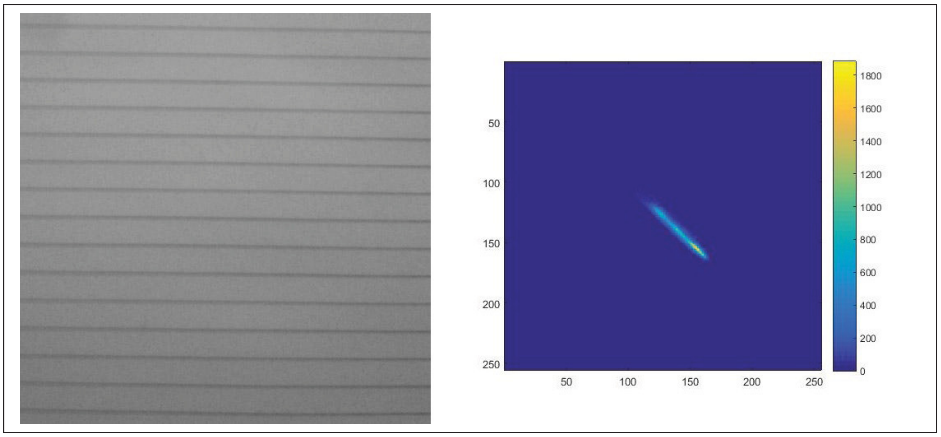


Figure 1.4 A structured texture and its GLCM. The transitions are similar and form a pattern that uses a small portion of the GLCM.

Because this matrix is large, the last step consists of greatly reducing the amount of information by computing key statistics from this large matrix. Gonzalez *et al.* (2004) presents six statistics: contrast, correlation, energy, homogeneity, maximum probability and entropy. Those six statistics were used as features by Bouchard (2011). Several authors have contributed by extending this list. Shihavuddin *et al.* (2013) used 16 additional features from various authors, as proposed by Uppuluri (2008): dissimilarity, inverse difference, inverse difference moment, inverse difference normalized, inverse difference moment normalized, sum of squared variance, sum average, sum entropy, sum variance, etc.

While GLCM statistics are relevant features for texture classification, they suffer from sensitivity to background information which may present unusual and sudden change in the texture pattern. Unfortunately, this tends to happen often in natural scenes, specifically if proper segmentation is not applied. While still useful, these features are expected to increase in quality once future work addresses the segmentation problematic.

1.4.3 Gabor Filter Response

Gabor Wavelets are a class of functions that can be described as modulated Gaussian functions. They have been commonly applied in computer vision as 2d convolution filters. The general form is given in equation 1.7, as described by Fogel and Sagi (1989). Figure 1.5 shows an example of a filter bank generated using various parameters

Given the coordinates of the pixel on the resulting filter x and y, the modulation wavelength λ, the rotation angle of the filter θ, the phase ψ, the variance of the Gaussian function σ, and the scale factor γ, gabor filters are defined as follow:

$$G(\lambda,\theta,\psi,\sigma,\gamma,x,y) = exp\{-\frac{(x\cdot cos\theta + y\cdot sin\theta) + \gamma^2\cdot(-xsin\theta + ycos\theta))}{2\sigma^2}\} + exp\{i(2\pi\frac{(x\cdot cos\theta + y\cdot sin\theta)}{\lambda} + \psi)\} \tag{1.7}$$

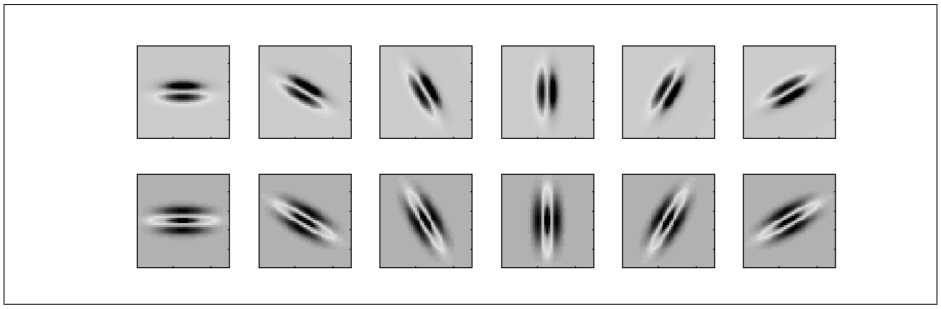


Figure 1.5 An example of a Gabor filter bank generated using a set of different parameters.

Typically, a filter bank is applied on the textured image in the form of successive 2d spatial convolution operations (see equation 1.8 for the definition of convolution), creating a series of response maps of size nearly equal to the input image. The size may be different, as convolution may ignore image borders to deal with incomplete information. The number of response maps is equal to the number of filters in the filter bank. As it is the case with GLCM, these response maps are generally too large to be useful for pattern recognition, and need to be reduced. This is why, as a final step, extraction of various statistical features on each response map is performed. These statistics, which can be used to form a feature vector, are often the following ones: mean, standard deviation, skewness, kurtosis.

Given an image f, and kernel k of size $S \times T$, convolution is defined as:

$$f(x,y) * k(x,y) = \sum_{s=1}^{S} \sum_{t=1}^{T} f(x-s, y-t) \cdot k(s,t) \tag{1.8}$$

It is worth noting that Gabor wavelets can be used in many other ways. For instance, they are used to create texton maps, which will be discussed later.

1.4.4 Other Global Descriptors

So far, we've covered three popular global descriptors (*i.e.* descriptors that aggregate information from every pixel in the image.) This section briefly introduces other global descriptors that have been applied to benthic image labeling.

Statical feature extracted from the intensity histogram were among the first features used for texture classification. Bouchard (2011) used seven feature extracted for each channel: mean, standard deviation, R-value of the second moment, skewness, kurtosis, uniformity and entropy. These features have shown to be powerful in simple cases. However in natural scenes, because variable illumination is a problem, these features may not perform as well as expected.

It has been shown numerous times that the introduction of color information in texture recognition increases significantly classification accuracy. For this purpose, Shihavuddin *et al.* (2013) used a hue histogram in combination with an opponent angle histogram, originally proposed by Van De Weijer and Schmid (2006). These features were developed to describe color in natural scenes, and respectively address the problems of photometric variations (shadows, shading, specularities) and geometric variations (change in viewpoint, zoom, object orientation). The concepts of hue and opponent angle are defined here by equation 1.9 and 1.10, where R, G, B represent the pixel in the original image, and R', B', G' represent the matching values of the first order derivative of each channel. In the implementation used, averages of the measured values are computed over small blocs of 20×20 px to smooth noise. Both histograms (hue and opponent angle) are weighted respectively by saturation and by a geometrical error factor given by equation 1.11, which is essentially the inverse of the magnitude of both terms in equation 1.10.

$$hue = \tan^{-1} \frac{\sqrt{3}(R-G)}{R+G-2B} \tag{1.9}$$

$$ang = \tan^{-1} \frac{\sqrt{3}(R'-G')}{R'+G'-2B'} \tag{1.10}$$

$$w = \frac{1}{\sqrt{3(R'-G')^2+(R'+G'-2B')^2}} \tag{1.11}$$

Bouchard (2011) also experimented with texture features from the Fourier spectral representation. As defined by Gonzalez *et al.* (2004), an image in the frequency domain can be flattened into a one dimension feature vector. Two ways of vectorizing the frequency domain are proposed, with respect to the angle (equation 1.12), and to the distance from the center (equation 1.13). Bouchard (2011) found that vectorizing in a fixed number of discrete bins using the latter method yielded discriminative features.

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \tag{1.12}$$

$$S(\theta) = \sum_{r=1}^{R_{max}} S_r(\theta) \tag{1.13}$$

1.4.5 Textons

Beijbom *et al.* (2012) used a texton histogram, which is a powerful method in the field of texture recognition that has been drawing much attention from the computer vision community since its introduction by Varma and Zisserman (2005). A texton histogram can be described as the frequency at which primitive texture prototypes are observed in the input image. These primitive prototypes are called "texton". Textons are usually listed in a simple matrix structure called a texton dictionary, where each line represents a unique texton in the form of a *n* dimensional vector in the texture feature space, also called texton space. Typically, the texton dictionary is generated using unsupervised clustering algorithms on a large number of texture feature vectors extracted from a sample of images depicting the various textures of interest. To summarize, textons can be used as follow: given a texton dictionary, and a set of pixels in a new image, one can compute each pixel's texture vector, vector-quantize each pixel's feature vector into a texton by associating it with the nearest known texton from the dictionary, and form a histogram describing the frequencies at which each texton is observed in the input image. Aside from the low level feature, this method is very similar to the popular Bag-of-Words method often used in computer vision. See Appendix III for additional information on Bag-of-Words.

Features used in the texton space were originally the Gabor wavelet reponses, where the dimensionality of the texton space is equal to the number of filter in the selected Gabor filter bank. Additional work has demonstrated that a large variety of features can be used instead, such as LBP or even small patches cropped directly from the texture.

Texton have proven to be powerful features in the context of coral reef image annotation. Among the reasons that could explain their impressive performance, we argue the texton histograms are capable of ignoring irrelevant background information by isolating most of it in a few separate bins. This is because the texton-pixel relationship is a one-to-one mapping, and background information will be associated with a few separate textons which has little impact on the description of the relevant textures. Because irrelevant background information is prominent in natural scenes, in particular in the absence of proper segmentation, it is not surprising to see textons perform better than other features. Also, the discriminative power of textons could be explained by the specificity of the texton dictionary. A dictionary can be created using coral reef images, which leads to textons highly representative of the expected textures for a given problem.

1.4.6 Describable Texture Dataset SVM Scores

While it has never been applied to benthic image annotation, Cimpoi *et al.* (2014) conducted recently a thorough study of texture classification in natural scenes. They introduce the idea that textures in natural scenes can be very different statistically while still being similar when observed at a semantic level. To illustrate this, they define a vocabulary of 47 texture description words humans use to describe various textures. They then design the "describable texture dataset" (DTD) consisting of 5640 images of 47 classes, each class corresponding to a very unique, yet typical example of a specific descriptive word. Using DTD, they densely extract the scale-invariant feature transform (SIFT, see appendix II for details on SIFT) descriptor at multiple scales. The resulting descriptor is a variable length set of histograms describing SIFT key points in the texture. This set of key points is then soft vector-quantized using the Improved Fisher Vector (IFV) method, which is, like the texton method, a way of pooling features into a fixed-length vector using quantization based on Gaussian Mixture Models (GMM) density estimation (see appendix III for more information on IFV). This proposed method based on IFV, while very powerful, leads to feature vectors of dimensionality above 40 000. To reduce dimensionality, the authors propose training a 47-class one-against-all SVM, and using the 47-value score vectors directly as features. They call this method DTD^{IFV}. They experimented with various SVM kernels and found the RBF kernel worked best. The resulting variation is called DTD^{IFV}_{RBF}. Intuitively, this method is similar to a semantic description of the texture using words.

Unlike previously introduced global descriptors, DTD^{IFV}_{RBF} relies on local features, and is extremely powerful when it comes to describing patterns at a much higher level of abstraction. As shown in figure 1.6, it was designed to find similarities in patterns that are semantically alike, but statistically very different. This may be applicable to benthic image annotation, where a high intra-class variability has been observed, which remains a challenge with traditional descriptors.

1.4.7 Deep Convolutional Activation Feature

Following the same trend, Donahue *et al.* (2013) proposed another feature to provide a description at a higher level of abstraction. They call their feature the deep convolutional activation feature (DeCAF). The objective of DeCAF is to use a fine tuned convolutionnal neural network (CNN) trained for object recognition on millions of images, and re-purpose it to a context specific classifier.

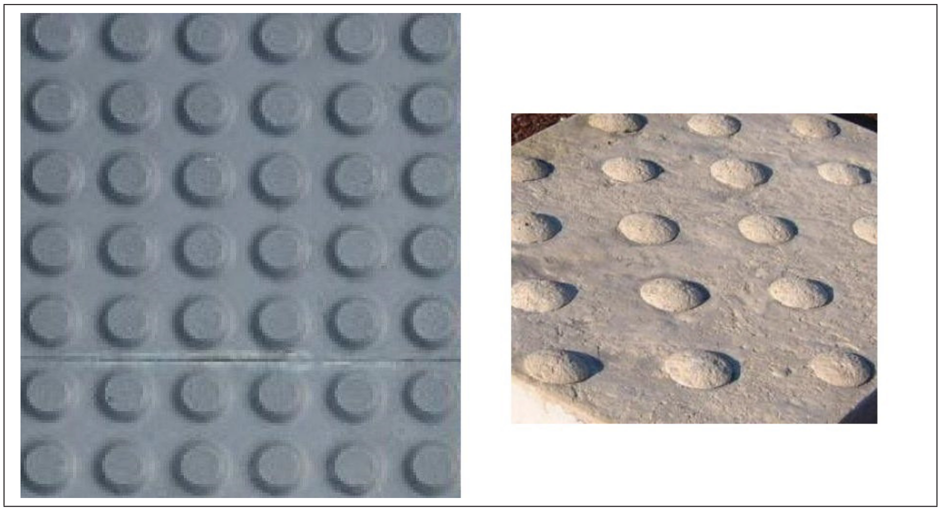


Figure 1.6 Two textures from the Describable Texture Dataset (DTD) representing the "bumpy" class. Both are quite statistically different, yet semantically representative of the bumpy class.

A typical CNN consists in roughly a dozen layers of neurons finely adjusted through back-propagation, and working together to correctly predict the object in an image (*e.g.* car, person, dog, flower, mountain, apple, etc). In order to do so, such networks are trained for extended periods of time on a massive amount of training examples, which makes them impractical to generate, hence the interest to re-purpose a publicly available pre-trained network.

The DeCAF method achieves this by trimming off the last two layers of the pre-trained network, which are respectively a fully connected and a softmax layer that both focus on object classification, and therefore turning the single value class output of the network into a much larger vector representing neuron activations of the last convolutional layer. Each one of these activation weights was originally trained to map input signals to specific classes of objects, but the vector as a whole can be used as a feature vector directly, which reportedly works very well. The rationale is that texture presenting distinctive semantic properties will tend to produce a similar activation pattern. This was first tested for textures recognition by Cimpoi *et al.* (2014).

CNNs can be very complex in practice, but they can be considered as a black box when extracting the DeCAF descriptor. As a result, it is not required to explore CNNs futher, and the high complexity of these networks is beyond the scope of this work.

To summarize, DeCAF shows promising results for situations where a texture can only be described at a higher level of abstraction. Though it has yet to be tested, DeCAF is certainly an interesting candidate for coral annotation, as it brings additional semantic information that seems different from the statistical global descriptors, such a LBP or textons. The information from DeCAF may be an interesting complement to previously introduced descriptors.

1.5 Dimensionality Reduction

Dimensionality reduction is the process of lowering the dimensionality of a feature vector. There are two main problems associated with high dimensionality. The first one is commonly referred to as the "curse of dimensionality". This phenomenon occurs when a small dataset is represented in a high dimensionality euclidean space. The small size of the dataset causes high sparsity in space, leading to the inability to correctly define decision boundaries for cases that have not yet been seen by the classifier. The second problem associated with high dimensionality is the computational cost.

Dimensionality reduction is sometimes considered to be a particular case of feature extraction, called feature space transformation, and uses statistical or heuristics techniques to eliminate the less descriptive portion of the information, as explained by Cheriet *et al.* (2007). A popular dimensionality reduction technique is the principal component analysis (PCA). PCA uses the covariance matrix to find the principal components in the feature space, and project the feature vectors into this new space. The method hypothesizes that the projected components with the highest variance present more representative features. It follows that the components with lower variance can be eliminated. While PCA is a fast and easy-to-apply method, it is often criticized because it may degrade the quality of the resulting features, and it should therefore be used carefully. I was applied to benthic image labeling by Bouchard (2011), but the work was inconclusive in regards to its usefulness.

Feature selection is a special case of dimensionality reduction that can often be a less radical alternative to feature space projection. Feature selection uses an objective function to compute a score, therefore heuristically rating the quality of each feature (or even feature set), and enabling the possibility of eliminating features which scored bellow a user specified threshold. A large variety of objective functions have been proposed. Work presented by Prévost (2015) makes extensive use of feature selection, and has shown promising result. The correlation based feature selection method was applied to reduce dimensionality from 18760 to 264. The empirical results suggested that high feature quality was maintained through dimensionality reduction.

In light of this, it becomes apparent that dimensionality reduction should be applied carefully, as it may destroy important information within the features. While the focus of this study will not be the impact dimensionality reduction has on classification accuracy, it may be an interesting option in future work to reduce computational complexity if required.

1.6 Classification

In pattern recognition, classification is the ability to predict the class associated with a new observation. For this task, a classification model needs to be trained on several examples. The process of training a model can be described as the search of a decision boundary or function that separates the $\mathbb{R}^N$ feature space, enabling class prediction of new instances. Cheriet *et al.* (2007) defines it as a search for a function f that can associate the correct class ω to the feature vector X, among all possible classes Ω, as represented in equation 1.14. Several methods have been developed to perform this task.

$$f : X \mapsto \omega | \omega \in \Omega \tag{1.14}$$

In the problem currently being studied, support vector machines (SVM) have been very popular. Preliminary work (Blanchet) brings additional evidence that RBF kernel SVMs (see section 1.6.2 for details on RBF kernel) do in fact consistently outperform other classifiers for the task of benthic image annotation. In addition, the "No Free Lunch Theorem" supports this idea:

because RBF kernel SVMs have been widely applied to texture classification, they are likely to perform well in a new problem of the same class such as texture-based benthic image annotation. However, this assumption may need to be verified more thoroughly, as recent progress in the field of machine learning has lead to the emergence of new classification approaches such as deep convolutional neural networks. Nonetheless, these are considered beyond the scope of the current work.

SVMs are highly flexible discriminative and non-parametric models that have been very popular since their recent introduction. SVMs use Lagrange multipliers as an optimization technique to find a hyperplane that can optimally separate samples from two classes represented by a set of training points in $\mathbb{R}^N$ space. By problem definition, the optimal hyperplane maximizes the margin between points of both classes. The rationale is that a hyperplane with a larger margin will tend to generalize better.

The remaining portion of this section presents various important aspects of the SVM. The objective is not to present a thorough mathematical explanation of the inner optimization process of the SVM model, as it has already been done so often, but rather to offer a conceptual and comprehensive introduction to the many aspects of this tool necessary to its proper usage. See work by Cortes and Vapnik (1995) for details on the SVM optimization problem.

1.6.1 Multi-class SVMs

Originally the SVM was designed for two-class (binary) problems but have since then been adapted to handle multi-class problems. Two particularly popular approaches exist for this: one-against-all and one-against-one. The one-against-all approach treats each class as a separate SVM problem with positive samples from the class of interest, and negative samples from every other class in the training set. Each SVM outputs a score, and the class selected is the one associated with the SVM leading to the highest score against all other classes. Alternatively, one-against-one is a much more popular approach that consist in training a SVM for each combination of classes. Each SVM votes for one of the two classes it is trained to distinguished. The winning class has the most votes.

Other solutions to the multiclass problem were proposed, such as the directed acyclic graph SVM, which is essentially a decision tree where each node is a binary SVM. These techniques are relevant, but they are considered beyond the scope of this work.

1.6.2 Kernel Trick

The kernel trick is a way to efficiently treat a distance-based (or similarity-based) problem as if it were projected in a higher dimensionality space, thereby increasing the separability of the data in cases where linear separation in features space cannot be achieved reasonably well. Because SVMs are based on point distances, it is possible to use alternative or modified distance functions. These functions are called kernels. Many different kernel functions have been proposed over time. One of the most popular is the radial basis function (RBF) which was used by Bouchard (2011); Shihavuddin *et al.* (2013); Beijbom *et al.* (2012); Prévost (2015) for benthic image annotation. Equation 1.15 represents the RBF kernel formulation between vector x and y. It is essentially the same as mapping the squared euclidean distance $||x-y||^2$ to a Gaussian-like distance metric. The kernel depends on a free parameter γ which defines the tolerance of the function, and needs to be optimized. This will be further discussed in section 1.6.3.

$$K_{RBF}(x,y) = e^{-\gamma||x-y||^2} \tag{1.15}$$

1.6.3 Model selection

The training of a RBF kernel SVM depends two free parameters:

- **The error cost parameter** C. The task of finding a support vector in $\mathbb{R}^N$ feature space that perfectly separates the two-class data cannot be done in most case due to the complexity of the data. This is why the problem formulation considers the cost of error in its optimization, in the form of a weighted loss function. The C parameter represents the weight, or the trade off between classification errors and the margin size.

- **The kernel function parameter γ.** As it is the case with most other kernel functions, the RBF kernel depends on the free parameter γ that represents $(2\sigma)^{-2}$, an exponential factor that controls the tolerance of the output Gaussian-like distance. As γ increases, points that are further away from each other are seen as closer by the kernel function.

Model selection, as defined by Cheriet *et al.* (2007), is the process of selecting a good model among a set of possible models, each one having a variable performance. In a context where one or more free parameters affect performance, model selection techniques can be applied to select values for those parameters. It becomes apparent that model selection is an empirical process: various models are generated, a performance metric evaluates each model, and the best one is retained. Various model selection methods exist, in the form of search algorithms. Arguably the simplest one, the grid search, is quite popular as it performs well.

In a typical grid search, various combination of (C,γ) parameters are generated across a reasonably bounded logarithmic domain. Models are trained on two thirds of the data, and validated on the remaining third. The classification error is then used as a performance metric, which allows the best (C,γ) to be found. It is critical that model selection be performed on data completely independent of the data later used to test the system, in order to avoid bias.

1.6.4 Multiple classifier fusion

Duda *et al.* (2012) describes a technique to enhance classification performance called "evidence pooling". The idea relies on the assumption that the advice from multiple experts is more likely to be accurate than the advice of a single expert. Multi-classifier fusion (MCF) methods are an example of evidence pooling that had many successful applications in the past decade. For instance, MCF methods can use a set of SVMs trained on different features. It is expected that each SVM will specialize in solving a portion of the problem, and the aggregation of all classification outputs, also called fusion, will yield better class predictions than any single SVM alone.

This fusion process is usually performed on a score metric, which a normal SVM does not output. Consequently, instead of using typical classification SVMs, it is preferable to train a regression SVMs to estimate the probability of belonging to each class, as explained by Chang and Lin (2011). These probabilities can be aggregated in various ways: sum, mean, maximum, product, vote, etc. The class with the highest aggregated score becomes the predicted class. Figure 1.7 illustrates the general MCF framework.

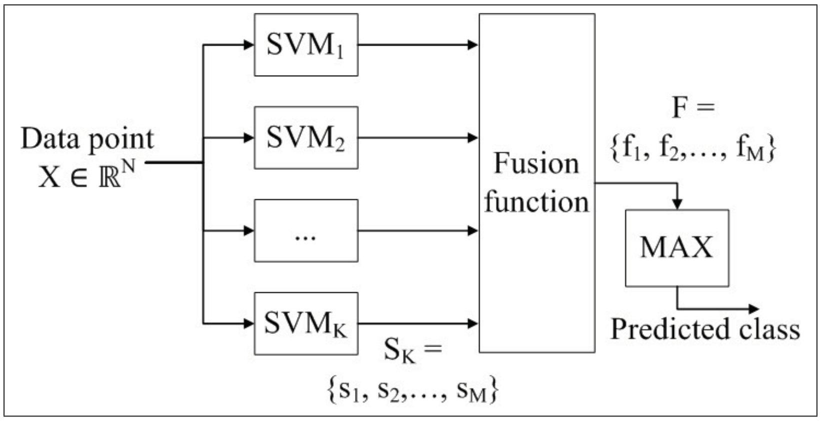


Figure 1.7 The multi-classifier fusion process of point X in N dimensions, using K SVMs, in a M-class problem. S represents the output scores and F the aggregated scores.

1.7 Rejection

Rejection is an optional step that consist in ignoring classifier outputs considered unreliable. Rejection is very popular in some computer vision applications, such as biometrics, where the cost of false positive is high. In the case of automated benthic image annotation, the system could reject a portion of the images. These could either be manually labeled afterwards, or simply ignored, therefore accepting the biodiversity estimation error associated with the rejection.

Applying rejection is as simple as thresholding the reliability score output of the classifier. The difficulty lies in the threshold selection. Several way of selecting a threshold exist. One popular approach, first proposed by Chow (1970), consists in estimating the classification error within the Bayesian framework, and selecting the threshold leading to the minimum error. Because the estimated probability is not perfect in practice, many heuristic methods were proposed afterwards. Dubuisson and Masson (1993) extends the method by proposing not only an ambiguity threshold, but also a distance-based threshold to handle novelty (previously unseen classes). However, regardless of the method, one must understand the context and the cost of rejection to correctly select a threshold.

As explained by Duda *et al.* (2012), selecting a rejection threshold is a tradeoff between precision and recall. Precision is defined as the correct prediction rate in the non-rejected set, while recall is the fraction of all correctly classified elements retained. In the case of benthic image annotation, as the threshold increases, predictions are more reliable, but consider a smaller sample, which is very likely to alters the biodiversity statistics. On the other hand, as the threshold decreases, the resulting labeled sample size is larger, but this comes at the cost of increased errors, which also alters the biodiversity statistics.

CHAPTER 2

DATA

Coral reef natural scenes are particularly complex from a computer vision perspective for multiple reasons. In this section on data, we explore the main challenges in available datasets. We then present two of these coral datasets: the AIMS dataset (Australian Institute of Marine Science) as well as the MLC dataset (Moorea Labeled Corals), we discuss their particularities in depth, and finally, we introduce other texture datasets that will be used to evaluate the ability of each descriptor to handle specific difficulties.

2.1 Challenges

The challenges encountered vary from one dataset to another as a result of differences in the acquisition protocols and environments.

- **Variable scale, orientation and illumination** is an expected challenge in natural scene images. Figure 2.1 presents six observations of a single coral species where this variability can be observed. This points towards texture features that are tolerant to this variability.

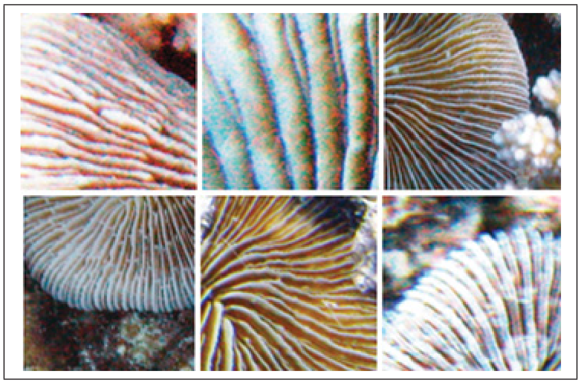


Figure 2.1 Six samples of the same species. The texture scale, orientation and illumination varies significantly.

- **Red channel information loss**. As previously discussed, the red wavelength tends to be absorbed by the water after just a few meters. This results in significantly less information in the red channel, and pictures sometimes appear to be blue. Figure 2.2 shows an example of an image with important red information loss.

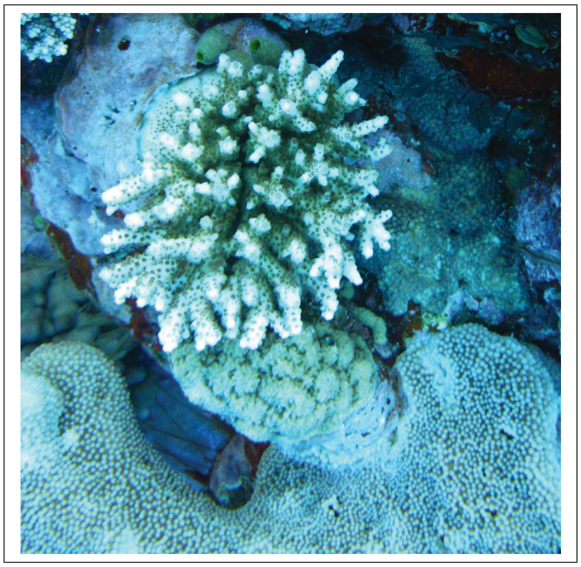


Figure 2.2 A example of red channel information loss from the AIMS dataset.

- **Chromatic aberration**: Light that passes through multiple translucent mediums cause a phenomenon called refraction. Refraction may cause light to decompose into its color spectrum. The effect on the resulting image is known as a chromatic aberration, as shown in figure 2.3. Special equipment to mitigate this effect exist such as dome ports designed for underwater acquisition, but unless these are used, the resulting images present radial distortions in the red and blue channel, which are respectively offset towards and away from the center of the lens. Being the middle wavelength, the green channel appears somewhat undistorted. This effect is consistently seen in the AIMS dataset, but may not be a concern with most datasets.
- **Imbalanced data**. Though it is expected, the non-uniform representation of every class can be a problem from a pattern recognition perspective. Even when a large sample of

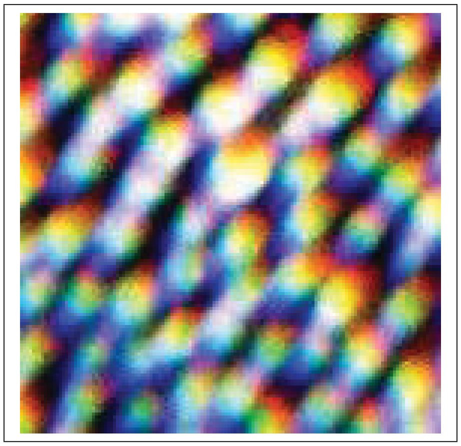


Figure 2.3 An example of a chromatic aberration from the AIMS dataset. Histogram equalization was applied to enhance the aberration.

data is taken, some classes will naturally be less frequent. In some cases, classes can be exceptionally rare. The few samples available for such a class may often not be enough to characterize the class in all of its natural variability.

- **Incorrect expert labeling** was discussed in the introduction as a problem with manual labeling. Because the output of manual labeling is used directly as a ground truth, the effects of this problem extend to computer vision as well. Errors in the dataset influence the classifier training process not just in terms of accuracy, but may even slow down the training process, as it is the case when using a SVM. Because the data is not easily separable, the optimization process does not converge quickly. Furthermore, incorrect labeling is a problem when assessing the accuracy of the system. Where the system made a correct prediction, perhaps the expert did not, leading to inaccurate performance measurements.

- **Sampling methods** proposed by expert annotation protocols are either systematic (*i.e.* always the same points used for annotation) or random (*i.e.* a fixed number of points are randomly selected). While these methods make sense from a statistical point of view, they ignore the complex organic shape of underwater content. This leads to confusion in labeling (for humans and machines), because points are often in between two or more classes. Ideally, the annotation protocol should allow the expert to lift all ambiguity, either by slightly moving the sampled point, specifying a vector pointing towards the region of interest, or sampling manually a few pixels in the region of interest. Obviously, this is not needed for most points, but would greatly increase the quality of the predictions. Results shown in Appendix I support this.

- **Complex environment**. Ideally, images should only present objects of classes known to the system. The reality is that sometimes, objects are irrelevant (*e.g.* fish) or impossible to distinguish given the lighting. This is dealt with in different ways. The AIMS dataset will use the "Other" class for this purpose (see figure 2.4), while the MLC dataset will usually ignore the ambiguity of these scenarios completely and instead label something of interest nearby in the image (see figure 2.5).

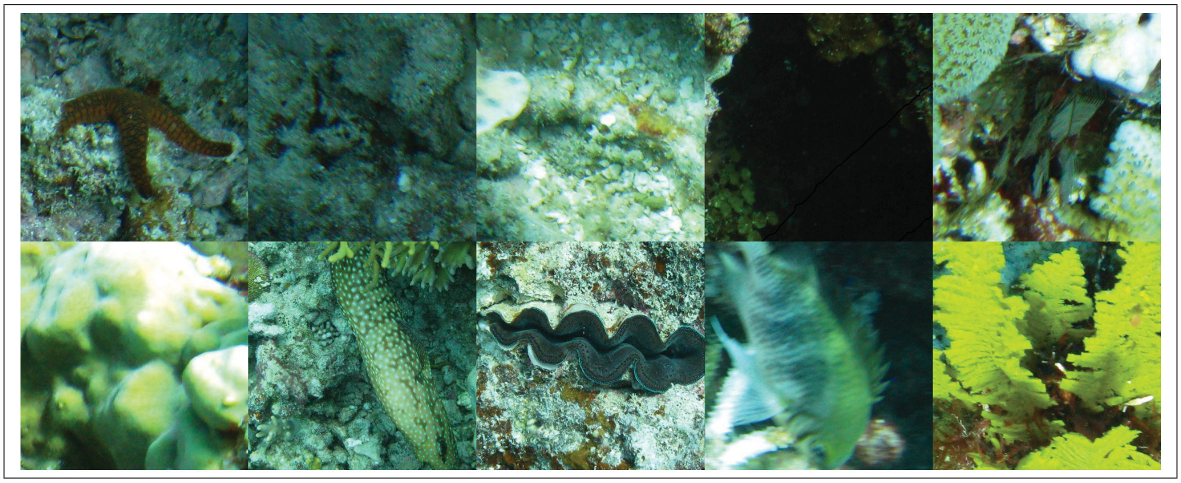


Figure 2.4 Examples of the "Others" class from the AIMS dataset.

- **High intra-class variance.** Coral colonies are complex. Their shape, color and texture all depend on many environmental factors, such as the time of the day, the month, the geographical position, etc. This is why a single class may be very heterogeneous. This causes classes to be scattered in the feature space.

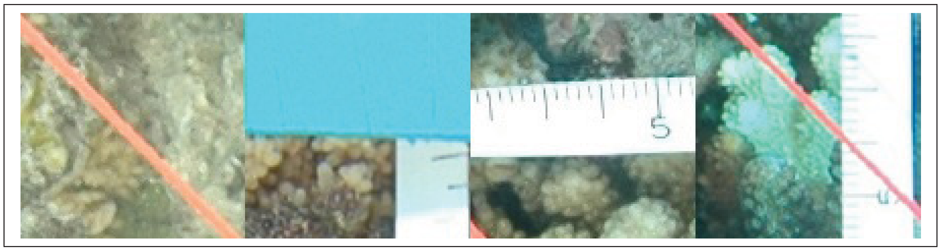


Figure 2.5 Examples from the MLC dataset where obstructing objects are labeled as *Acropora*.

- **Low inter-class variance.** In some cases, two different species of coral, algae or even sponges can present nearly identical texture patterns to the non expert eye. This causes overlapping between different classes in the feature space.

2.2 AIMS Dataset

The first dataset used in this work was provided by the Australian Institute of Marine Science (AIMS), which we'll refer to as the AIMS dataset (Jonker *et al.*, 2008a). The dataset contains 15 165 RGB 24-bit jpeg images taken in the area of the Great Barrier Reef in Australia between 2006 and 2012 for 90 transects each surveyed four times with two-year intervals in-between surveys. Figure 2.6 presents a typical image from the AIMS dataset. Image acquisition was performed underwater with a 6 mm lens at a distance of approximately 50 cm from the substrate, resulting in roughly 25 × 34 cm ground coverage per image. Two different resolutions are used: 3264 × 2448 pixels for images from 2006 to 2010 and 2112 × 2816 pixels for 2011 and 2012. The focal ratio setting varies between f/2.8 and f/9.0, the ISO speed setting between 80 and 3200, and the exposure time between 1/20 s and 1/2000 s. Because no artificial light sources were used, this is likely a result of manual and automatic adjustments to varying lighting conditions, resulting in images of variable quality. Figure 2.7 compares the quality of two images taken 2 seconds apart from the same reef with two exposure times of respectively 1/60 s and 1/250 s. While exposure is not the only factor at play, longer exposure times will often yield blurry images. Figure 2.8 presents the frequencies at which the various exposure times are found across the AIMS dataset. The high variance in the exposure time suggests variable levels of sharpness or brightness in the images.

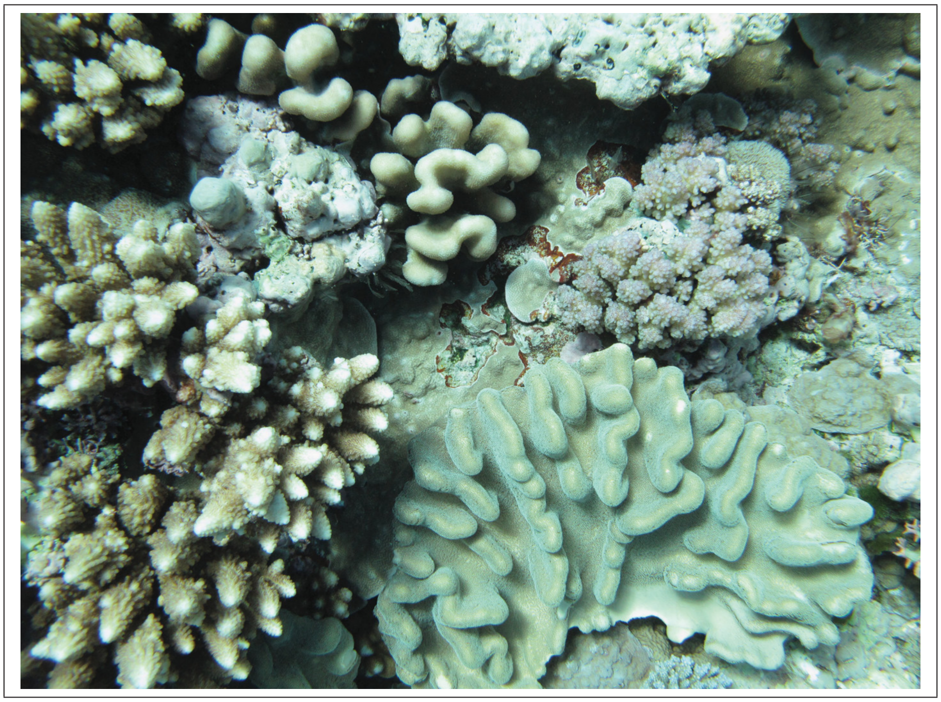


Figure 2.6 A typical image from the AIMS dataset.

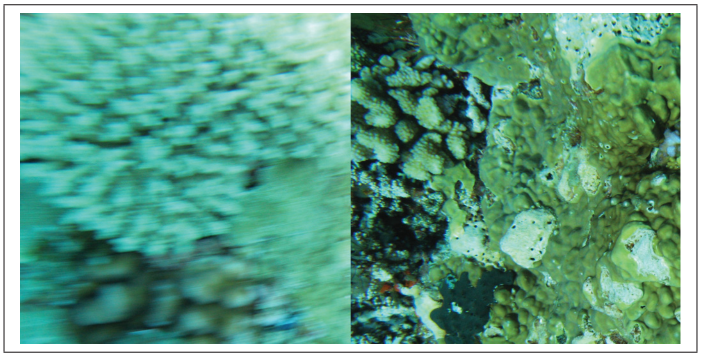


Figure 2.7 Samples from two images taken two seconds apart. The left and right one with exposure times of respectively 60 s^{-1} and 250 s^{-1}.

Each image was expertly hand labeled at five distinct points located at the following relative coordinates $(x,y) = (\frac{1}{4},\frac{1}{4}),(\frac{1}{4},\frac{3}{4}),(\frac{1}{2},\frac{1}{2}),(\frac{3}{4},\frac{1}{4}),(\frac{3}{4},\frac{3}{4})$ as it can be seen in figure 2.9. Labeling

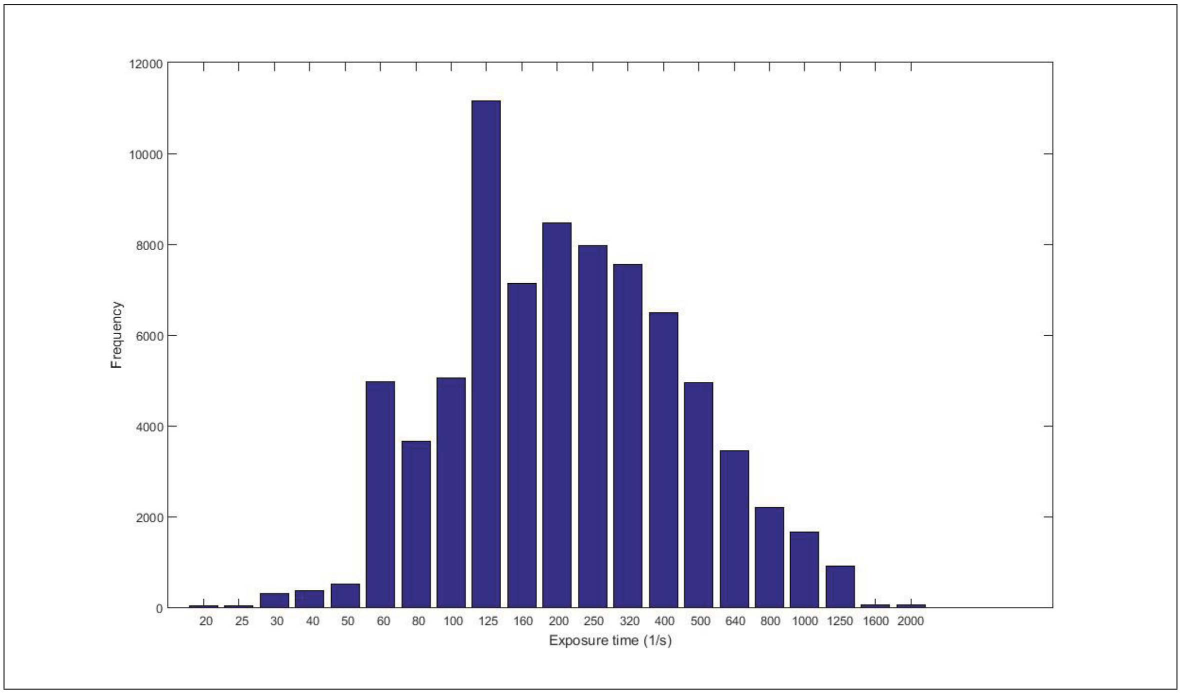


Figure 2.8 Exposure time frequencies in the AIMS dataset (in s^{-1}).

was performed at five different classification levels, from the broadest level to the finest: Group description, Benthos description, Family, Genus, Species description. For practicality as well as confidentiality, the exact classes and their frequency are not included. Instead, table 2.1 presents statistical data of each classification level. It can be seen that the number of samples per class is subject to an extreme variance. At the Genus level, 25% of the 118 classes have less than five samples and at the species description, half of the 225 classes have less than 19. Considering the high intra-class variance within each class, this poses a great challenge from a pattern recognition point of view.

When experts are unable to determine the label of a given point, they use special labels, similar to how rejection works in computer vision. At the group description level, the dataset contains three classes for such cases: "other", "N/A", "indeterminate". Manual inspection of those points revealed that the textures are too distinct to be classified using pattern recognition: unilluminated, light saturated or blurry areas, fishes, other life forms, etc. This problem will be further discussed in the methodology chapter. It is also worth noting that the goal of the

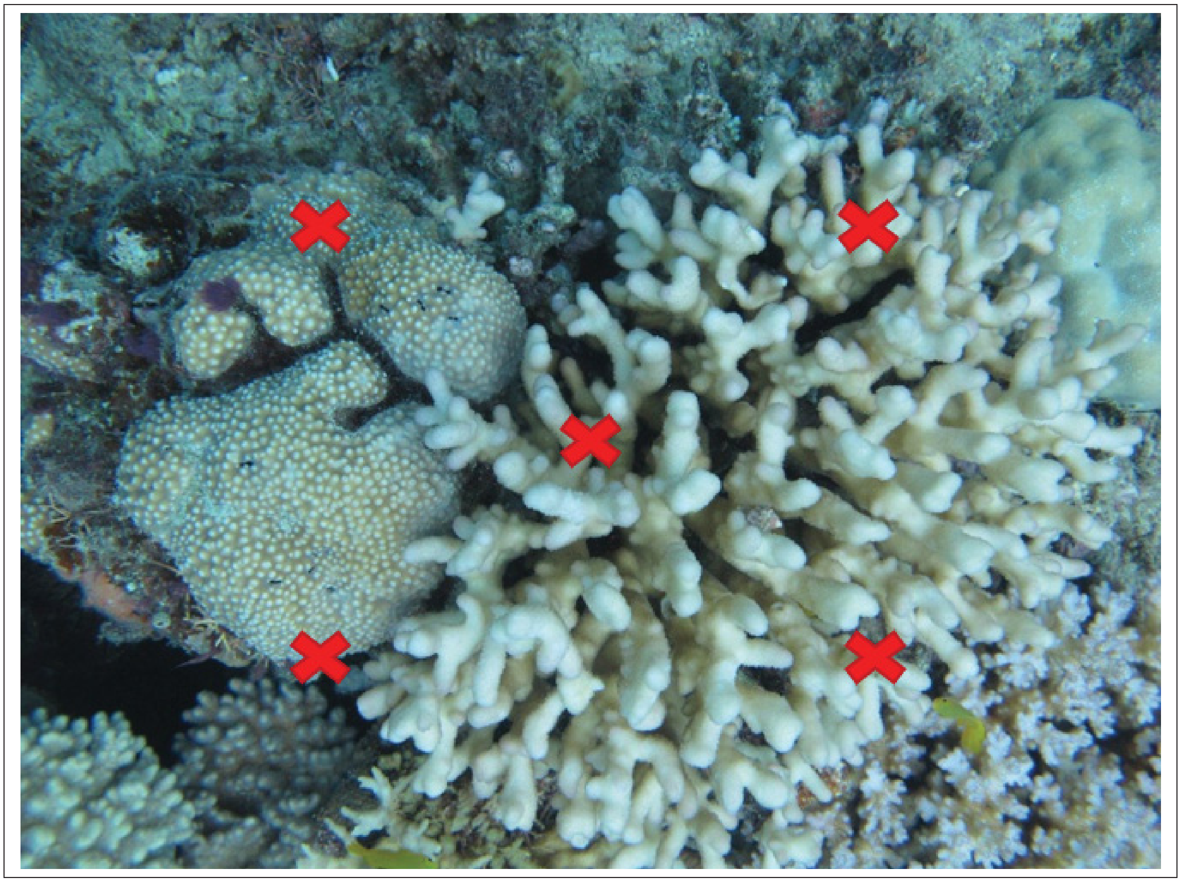


Figure 2.9 The five points sampled for each image in the AIMS dataset.

Table 2.1 Statistics on the number of labels for the five classification levels of the AIMS dataset.

	Group desc.	**Bethos desc.**	**Family**	**Genus**	**Species desc.**
Number of classes	7	45	42	117	224
Mean	8 608	1 648	1 761	657	344
St. dev.	14 085	4198	4 492	2 751	1 990
Max	39 662	26 025	26 021	26 021	26 021
Q3	15 072	1 200	737	250	91
Q2	1 500	625	226	40	19
Q1	508	49	37	5	3
Min	10	1	2	1	1

expert annotation protocol is to generate biodiversity data, and it was not designed to produce a dataset ideal for pattern recognition. Labeling single points instead of areas does not provide contextual information. Is it unknown what area around a point contains texture representative of the expertly labeled class. Furthermore, this affects the problem of segmentation by creating ambiguities for many points located in a transitional area between two textures. While this is a limitation for now, future work will investigate the problem of manually labeling a dataset for computer vision purposes, as well as the problem of applying segmentation to increase the texture sampling quality.

2.3 MLC Dataset

Edmunds *et al.* (2012) introduced a new dataset from the southern Pacific island of Moorea, which was first used by Beijbom *et al.* (2012). The publicly available Moorea Labeled Corals (MLC) dataset is part of the Moorea Coral Reef-Long Term Ecological Research (MCR-LTER) and contains for years 2008, 2009 and 2010 respectively 671, 695 and 689 jpeg and png images. The resolution used varies from an image to another, but averages at 1907 × 1915 pixels with standard deviations of respectively 106 and 103 pixels. Three lens of 18, 22 and 24 mm are used, and images are taken further away from the substrate compared to the AIMS dataset, which results in significantly more ground coverage per image. Unlike for the AIMS dataset, the aperture settings used for MLC yields much more consistent results. Focal ratio ranges from f/4.5 to f/13 and exposure time from 1/80 s to 1/500 s.

Each image was expertly labeled in 200 randomly selected points, and considers the circular region within a 15 pixel diameter around each point. A taxonomy consisting of nine dominant classes was used: *Crustos* coralline algae, turf algae, macroalgae, sand, *Acropora*, *Pavona*, *Montipora*, *Pocillopora*, and *Porites*. These classes represent 96% of the dataset. Work by Beijbom *et al.* (2012) proposed to simply ignore the other 4%, as it cannot be used for pattern recognition. Table 2.2 presents the frequency of each class.

Table 2.2 Representation of the nine classes across three years in the MLC dataset.

	2008	**2009**	**2010**
Crustos coralline algae	1.0%	0.3%	0.1%
Turf algae	48.8%	50.1%	77.5%
Macroalgae	7.0%	8.6%	2.7%
sand	2.7%	2.1%	1.7%
Acropora	1.4%	1.4%	1.3%
Pavona	5.8%	2.1%	0.7%
Montipora	10.5%	8.4%	2.8%
Pocillopora	10.6%	9.5%	9.3%
Porites	12.2%	17.4%	3.9%

The MLC dataset presents three main challenges:

a. As it is the case with systematic sampling, the random sampling used does not adapt well to the arbitrary shape of organic content. Generated points are often in transitional areas, which leads to ambiguity.

b. A quadrat (*i.e.* a frame used to sample a consistent area) is used for the underwater acquisition. This causes the appearance of a large frame, orange ropes, a white tape and some shadows, which modifies the texture. As an example, figure 2.10 presents a typical image from the MLC dataset.

c. As described by Beijbom *et al.* (2015), algae classes are highly prone to mislabeling, as experts often disagree while performing manual annotation.

2.4 Other Datasets

Because texture recognition techniques are applied in this work, it is interesting to measure the performance of the various features on well known texture benchmark datasets. The intent of such experiments is simply to gain insight on how a given set of features can adapt to constraints imposed by the dataset, such as scale, orientation or illumination changes.

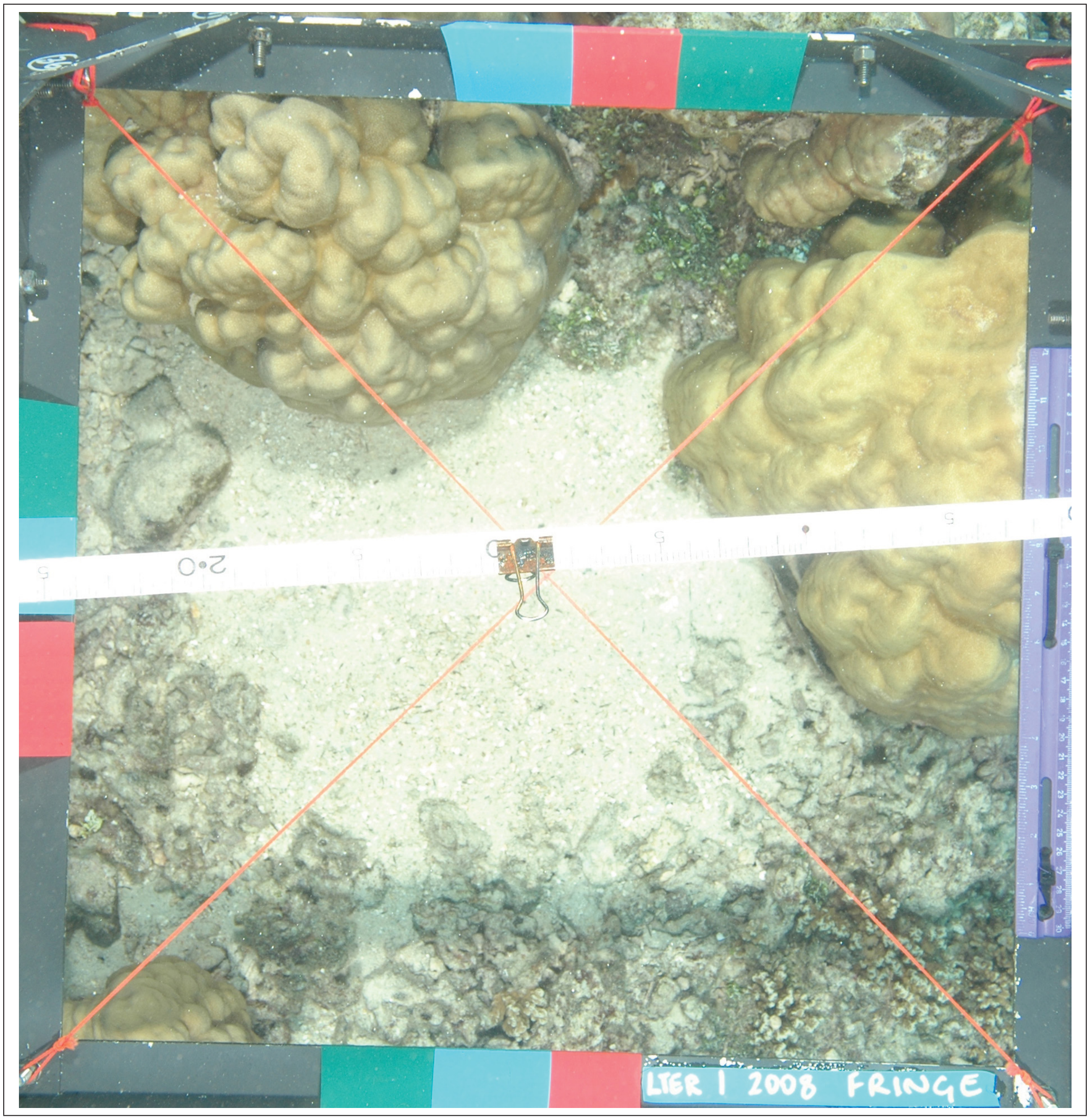


Figure 2.10 Typical image from the MLC dataset. The quadrat, orange rope, white tape and shadow modify the perceived texture around a given nearby point.

For this purpose, we also include two texture datasets that each present their own challenge: The texture dataset presented by Lazebnik *et al.* (2005), which we'll refer to as TDL and the Columbia-utrecht reflectance and texture database (CUReT) introduced by Dana *et al.* (1997).

2.4.1 Texture Dataset (Lazebnik)

TDL contains 40 images per class, for 25 classes: bark1, bark2, bark3, wood1, wood2, wood3, water, granite, marble, floor1, floor2, pebbles, wall, brick1, brick2, glass1, glass2, carpet1, carpet2, upholstery, wallpaper, fur, knit, corduroy, plaid. Each sample consists of a single-channel 640 × 480 px image. Figure 2.11 presents nine samples of three distinct classes.

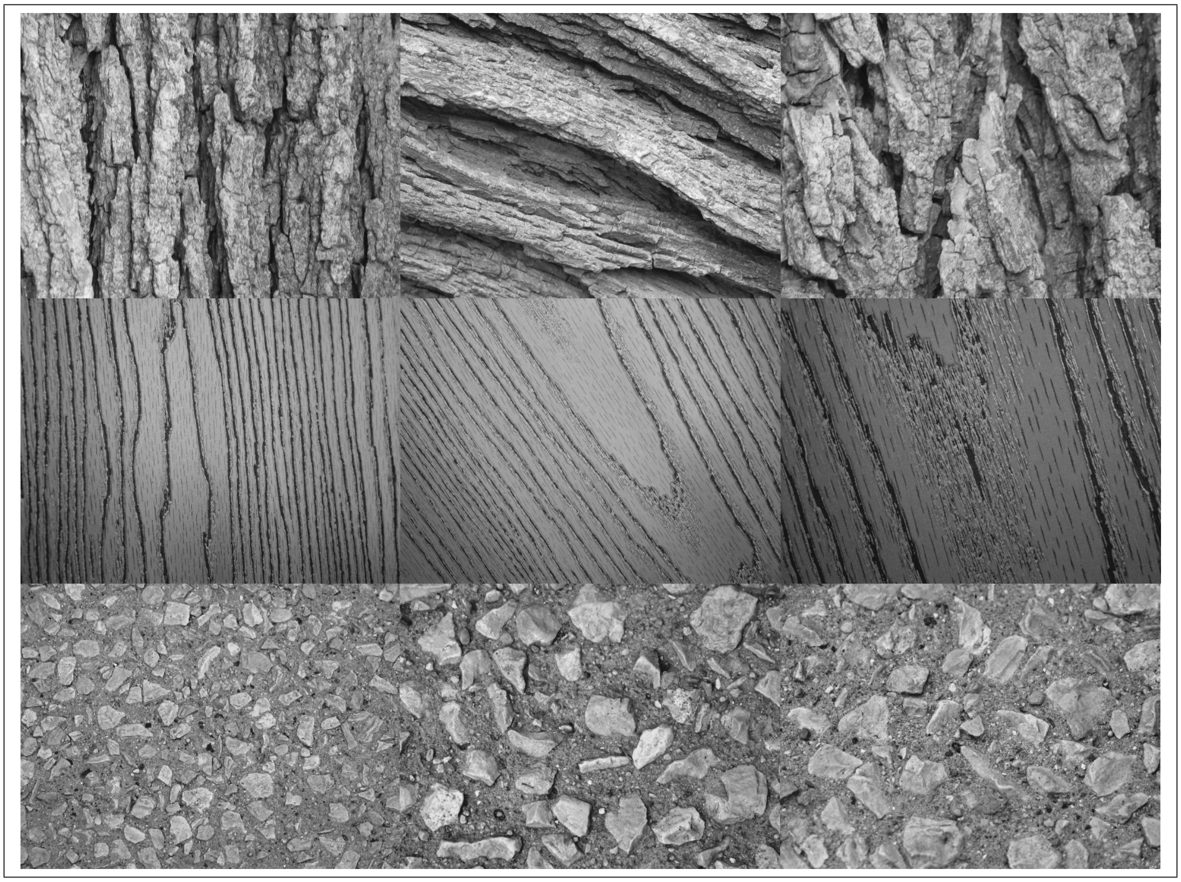


Figure 2.11 TDL dataset samples of the following three classes (top row to bottom row): bark1, wood1, pebbles

TDL brings characteristic challenges that test four desirable aspects of feature set:

a. Invariability to scale change.

b. Invariability to orientation change.

c. Tolerance to the absence of color information.

d. Tolerance to low, and moderately ambiguous inter-class variance.

2.4.2 Columbia-utrecht reflectance and texture database

CUReT presents 205 samples per class for 61 unnamed classes. Each image consists of a 640 × 480 px photograph of an object at a unique orientation relative to its lighting source. A commonly adopted methodology while working with this dataset consists in using only 92 images with enough surface to extract meaningful texture feature, as described by Varma and Zisserman (2005); Shihavuddin *et al.* (2013). Figure 2.12 presents sample images from two classes.

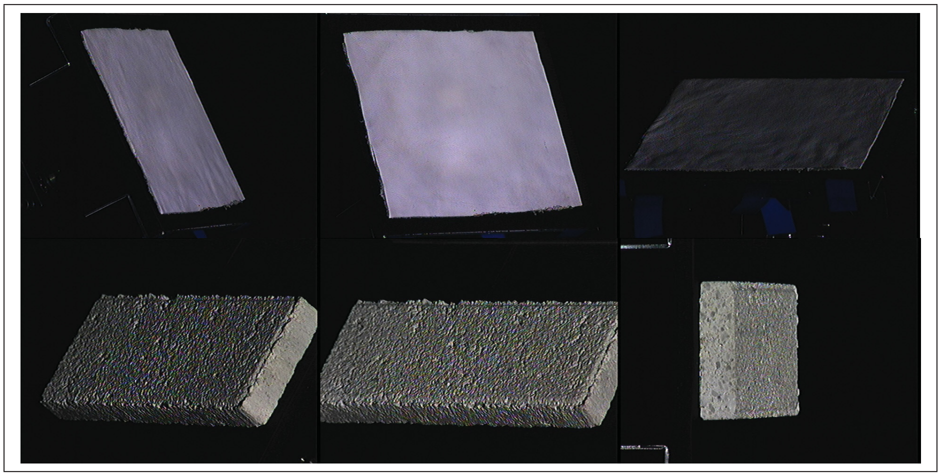


Figure 2.12 Samples from two classes from the CUReT dataset.

CUReT is useful for testing the following:

a. Invariability to illumination change.

b. Invariability to perspective orientation change.

c. Performance in a high class count problem.

d. Tolerance to high intra-class variance.

CHAPTER 3

METHOD

So far, a generally accepted framework for texture recognition in the field of computer vision was presented, and the relevant datasets were introduced. In this chapter, we bridge the two topics by training and testing a computer vision system that applies state-of-the-art texture recognition techniques to benthic photographs annotation. The proposed methodology will focus on each processing step individually, specifically: preprocessing, feature extraction, classification and rejection.

3.1 Preprocessing

Given the previously introduced challenges and particularities of each dataset, we present here a preprocessing method that can be applied to images from any dataset. Based on the results of previous work (Blanchet; Prévost, 2015), we propose a preprocessing method that was designed with respect to the following constraints:

a. Restoration filters should be targeted at a specific, well understood, image degradation phenomenons to limit potential image quality loss as a side effect.

b. Restoration filters should be applied adaptively. They should have little to no impact if the problem they aim to solve is not observed in the image. This is because every image is acquired under unique conditions, and quality varies significantly. This constraint enables the systematic filtering on every dataset without risk of major information loss.

c. Image enhancement filters that risk enhancing image flaws should not be applied.

Photographs from the AIMS and MLC datasets are both taken very close to the substrate and as a result, haze and marine snow are rarely seen in the images. Most underwater correction methods target at least one of the two phenomenons, hence the use of simpler and well targeted image restoration methods. Considering the previously introduced constraints, we propose using a two-step preprocess method that targets the following problematic phenomenons: chromatic aberration, and red channels information lost. The problems of motion blur and variable illumination conditions will be dealt with at the feature extraction step, by using robust features.

3.1.1 Chromatic aberration

Chromatic aberration, as defined in chapter 2, appears in nearly all AIMS images and causes two problems if ignored. Firstly, it has a significant impact on the result of other preprocessing algorithms since it causes unnatural RGB triplet values to occur in the image. For instance, instead of having two white pixels, the aberration may cause them to appear bright red and bright blue. Further processing is likely to enhance those unintended colors instead of the original white values. Secondly, by introducing color information that does not belong to the image, chromatic aberrations degrade the quality of the textures, and consequently of the extracted features. These may fail to describe the texture correctly as intended.

The chromatic aberration is caused by refraction, which is the deviation of the light's propagation direction due to the change of medium. This phenomenon in known as dispersion in optics physics, and is caused by the differences between the refractive properties of the water and of the lens. Red, green and blue lights have different wavelengths and phase velocities. The refraction index of a given medium is a function of the wavelength's phase velocity (ν). Consequently, red, green and blue are subject to different refraction indexes while passing from water to the lens, which causes light to divide itself in its colors spectrum, and results in the undesirable chromatic aberration. It is difficult to correctly model the refraction phenomenon, because the refraction index also varies according to the salt concentration in the water as well as its temperature. As a result, we treat the chromatic aberration correction as a simple optimization problem. We assume each image has a unique aberration, therefore this optimization is repeated for every image.

Given that the green wavelength (546.1 nm) is the middle wavelength between red (700 nm) and blue (435.8 nm), we can use the green channel as a reference of the expected solution, *i.e.* red and blue light should be refracted the same way the green light is. Or, in other words, the red and blue channels should be representative of the green channel. We then assume the red and blue channels were degraded respectively by a pincushion and a barrel radial distortion of unknown parameters. The problem then becomes a search for transformation parameters that minimize the error. We propose using the mean square error between the transformed channel and the green channel as a minimization function. Equation 3.1 and 3.2 formulate the proposed optimization for the red and blue channel, where R, G, B represent the red, green and blue channels, T is the corrective transformation function, s_R and s_B represent the transformation parameters for each channel and MSE is the mean square error function to minimize.

$$\begin{aligned} \underset{s_R}{min} \quad MSE_R(s_R) = \sum_x \sum_y (T(R,s_R)'_{x,y} - G_{x,y})^2 \\ s.t. \quad s_R \geq 1 \end{aligned} \tag{3.1}$$

$$\begin{aligned} \underset{s_B}{min} \quad MSE_B(s_B) = \sum_x \sum_y (T(B,s_B)'_{x,y} - G_{x,y})^2 \\ s.t. \quad 0 < s_B \leq 1 \end{aligned} \tag{3.2}$$

Furthermore, while a radial distortion is theoretically the appropriate transformation T for such corrections, we found that a simple bilinear image scaling lead to great results and offered a ten-fold computation time reduction. This could be because image acquisition is done at a very close range, which causes a near-linear radial distortion. Figure 3.1 presents examples of the proposed correction.

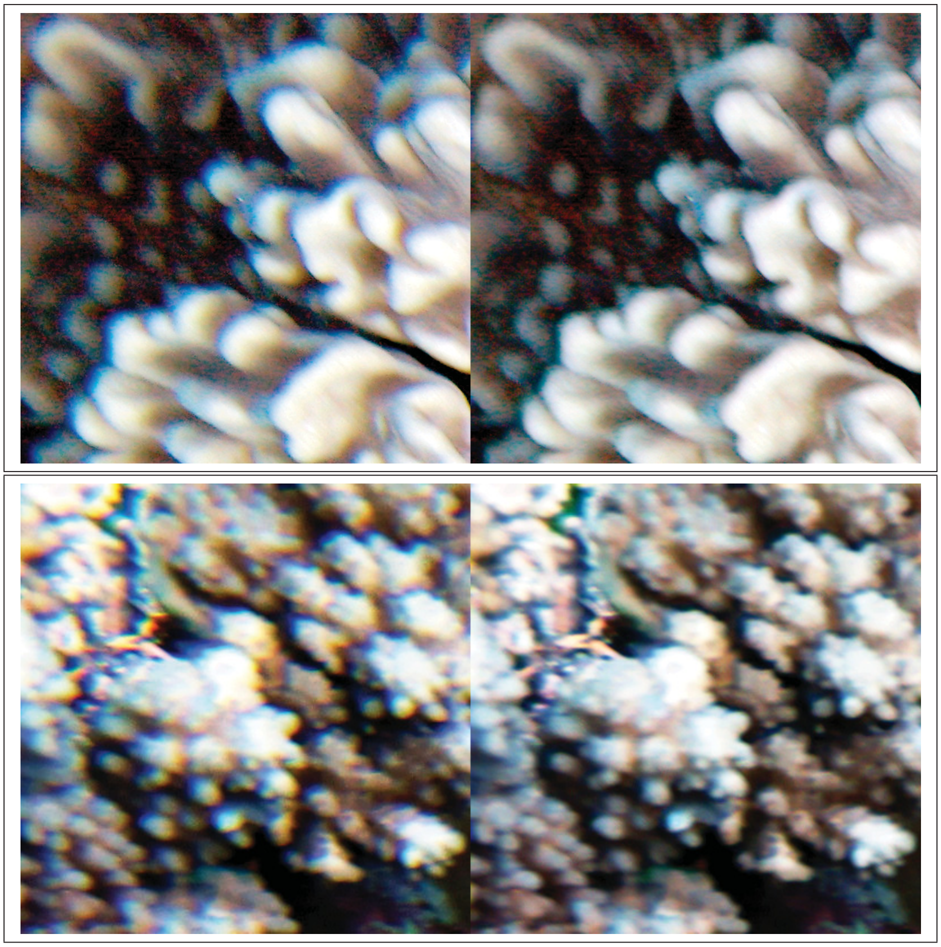


Figure 3.1 Examples from the AIMS dataset of the proposed chromatic aberration correction using simple linear optimization (original image on the left, corrected image on the right). Histogram equalization was applied to enhance visualization.

3.1.2 Channel information lost

Color information has been shown to be highly discriminative numerous times. Unfortunately, water absorbs a large portion of the red wavelength intensity, which alters the color. Previous work (Blanchet) has shown that simple channel stretch, histogram equalization or other simple enhancement methods lead to significant information loss. As an alternative, we propose using the Comprehensive Normalization method as a complementary source of information primarily aimed at describing the color. Features can be extracted a second time on the normalized image, and the resulting feature vector can be concatenated with the original one from the unchanged image. Figure 3.2 presents an enhanced example of the result following the application of the comprehensive normalization.

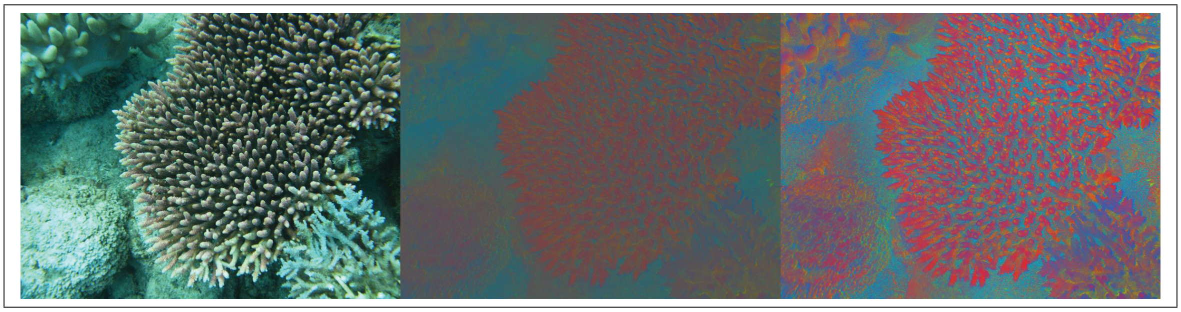


Figure 3.2 Comprehensive normalization preprocessing step example. From left to right: original image, comprehensive normalization output, additional adaptive histogram equalization for improved visualization.

3.2 Feature Extraction

As we've seen, feature extraction has been the focus of much work, and is arguably the most critical part of the processing pipeline. Many features from the field of texture recognition were introduced in section 1 each describing textures in a unique way. Because content of benthic images suffers from high intra-class and low inter-class variances, they are complex to describe. This is why we hypothesize that each previously proposed texture feature set brings additional unique information specifically useful in niche cases, and that the combination of many feature sets leads to greater performance. In this section, we first present a set of popular global texture descriptors. Secondly, we combine it with three other state of the art feature sets, each one having its own strengths. Thirdly, we briefly introduce part of our methodology to combine all these features, which will confirm our hypothesis. This will be further discussed in the next section on classification, fusion and rejection.

3.2.1 Proposed global feature vector

Combining popular global descriptors into a large vector has been done many times in benthic image annotation (Bouchard, 2011; Shihavuddin *et al.*, 2013; Blanchet; Prévost, 2015). Authors have proposed variations of combined features, often even disregarding the performance factor. Based on previous work, we combine and parametrize the same popular global texture descriptors and present a *global feature set*, which is designed on the following heuristic principles:

a. Performance is a constraint. It is unfair to assume time is a limitless resource. This will be further discussed in chapter 5, as we discuss potential applications of the system.

b. Redundancy in features is computationally costly, and too much of it may even affects accuracy by needlessly increasing dimensionality.

c. While multi-scale processing has been applied successfully to many problems, we found that it introduced much redundancy for little accuracy gain. Instead, we parametrize our feature extraction at the finest scale possible.

d. Previous work (Blanchet) applied feature selection, which lead to the pruning of some descriptors. These will not be included in the proposed feature set.

e. Concatenated descriptors should have roughly similar sizes. We found that the accuracy may drop if one vector is many times larger than the others.

f. Because image quality varies, pixel intensities can be quantized into larger bins when applicable, to compensate partly for the noise. This also reduces computational cost.

For the remaining portion of this section, we define every descriptor used within our proposed set of global features.

3.2.1.1 Intensity histogram features

The normalized histogram (probability histogram) from each color channel in addition to the gray level image is computed with 32 bins (8 bit channels, $\frac{256}{32} = 8$ pixel intensities per bin). Four features are extracted per histogram, for a total of 16 features: mean, standard deviation, uniformity and entropy. Given a 32-bin histogram H_c and normalized histogram P_c of the c channel, these features are defined by equations 3.3 to 3.6;

$$\bar{H_c} = \sum_{i=0}^{32-1} P_c(i) \cdot 8i \tag{3.3}$$

$$Std(H_c) = \sqrt{\sum_{i=0}^{32-1} (\bar{H_c} - H_c(i))^2 \cdot P_c(i)} \tag{3.4}$$

$$Uniformity(H_c) = \sum_{i=0}^{32-1} P_c(i)^2 \tag{3.5}$$

$$Entropy(H_c) = -\sum_{i=0}^{32-1} P_c(i) \cdot log_2(P_c(i) + \varepsilon) \tag{3.6}$$

3.2.1.2 Grey Level Cooccurrence matrix

The GLCM is extracted on the gray scale image quantized at 64 intensity levels. As for the neighboring pixel offset parameters, we consider both, the left and top pixels which are used to form a single matrix. By considering the immediate neighbors, we describe the texture at the finest scale. As previously introduced in section 1.4.2, 22 statistical measures are extracted from the resulting matrix. These are described thoroughly and made publicly available by Uppuluri (2008).

3.2.1.3 Completed local binary patterns

Because CLBP was shown to outperform the original LBP in the problem of texture recognition, we settle for CLBP as the finest scale, with a sampling of 8 neighbors at a distance of one pixel with a 10 bin uniform rotation-invariant mapping. The resulting center, magnitude and sign components of CLBP can be optimally combined in a 512 bin 3d histogram, but we opted for a fair accuracy-performance trade off by concatenating a 20 bin 2d center-magnitude joint histogram with a 10 bin 1d sign histogram. This results in 30 CLBP bins.

3.2.1.4 Gabor filtering

Our preliminary results suggested that Gabor filtering with a larger filter bank yielded very little relevant information. Consequently, we use a limited number of Gabor-based features compared to previous work. We extract two statistics from each filter response, with a filter bank at a single scale of $\sigma = 3$ pixels and six orientations, resulting in 12 statistics. Equation 3.7 and 3.8 present the statistics used for a Gabor response map G_r containing N pixels.

$$Std(G_r) = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(\bar{G}_r - G_r(i))^2} \tag{3.7}$$

$$Kurtosis(G_r) = \frac{\frac{1}{n}\sum_{i=1}^{N}(\bar{G}_r - G_r(i))^4}{(\frac{1}{n}\sum_{i=1}^{N}(\bar{G}_r - G_r(i))^2)^2} \tag{3.8}$$

3.2.1.5 Hue Histogram and opponent angle

Both hue and opponent angle color descriptor histograms are extracted each resulting in a 16 bin histogram. The features are computed exactly as proposed by Van De Weijer and Schmid (2006) and as applied by Shihavuddin *et al.* (2013) for a total of 32 bins.

3.2.1.6 Additional color information

The set presented so far contains a total of 112 features. We extract these a second time on the color-enhanced image resulting from the application of the comprehensive normalization method, which results in additional color information, for a total vector size of 224 features. While this introduces much redundancy, we found that the normalized image provided much useful color information, which resulted in a significant performance gain.

3.2.1.7 Comparison to previous work

To justify the introduction of our proposed global feature set, we also provide, in section 4.1.3, evidence of the much more reasonable computational cost compared to other methods, and show that it provides accuracy statistically identical to that of other proposed solutions. The former concern is addressed by measuring the executing time on samples of various sizes. While empirical complexity measurements are typically biased, it is a reasonably good alternative to the analytic approach in this case, because all implementations use the same descriptor extraction functions with different parameters. The analytic complexity would also be quite difficult to clearly interpret given the high number of variables. The latter concern on accuracy will be addressed simply by comparing the classification rates with those obtained using the other feature sets.

3.2.2 Combining features

Our proposed global feature set combines several computer vision descriptors and excels at describing textures from a statistical point of view. However, it becomes apparent that it also has limitations:

a. In the proposed feature space, textures are heavily affected by irrelevant background information, as most statistics represent a value obtained by aggregating the information from multiple or all pixels. This includes pixels from the intended region of interest as well as background pixels and is a result of poor segmentation, which greatly affects the quality of the features in many cases.

b. The features are sensitive to high intra-class variance. In the absence of a large number of samples, proper classification may be impossible.

To compensate these limitations, we propose using three additional state of the art feature sets: textons, DTD_{RBF}^{IFV} and DeCAF. Firstly, textons are extracted in the Lab color space exactly as proposed by Beijbom *et al.* (2012), using the same filter bank and dictionary of 135 textons, at four patch scales resulting in 540 texton bins. An additional channel stretch is applied for image enhancement. Because textons focus on describing each pixel individually, they are relatively more tolerant to background information. Secondly, DTD_{RBF}^{IFV} is extracted resulting in a 47 value feature vector. The RBF kernel SVM used for this is trained on the entire DTD dataset of roughly 5000 images of 47 classes, using a one-against-all strategy. DTD_{RBF}^{IFV} offers a very high level description of the aggregated local features: it is tolerant to limited background information as well as high intra-class variance. Thirdly, DeCAF is extracted resulting in a 4096 feature vector. The network used was trained on the ImageNet dataset by Simonyan and Zisserman (2014), and consists of 37 layers, 16 of which are convolutionnal layers of 64 to 4096 filters. Like DTD_{RBF}^{IFV}, DeCAF describes textures at a high level of abstraction, as if it described objects.

In order to confirm our hypothesis and show that these features can be complementary to each other, we propose a classification-level fusion of textons, DTD_{RBF}^{IFV}, DeCAF and our proposed feature set through evidence pooling, and show that it outperforms every feature set alone. We also include, as a comparison reference, results obtained using all other feature sets, including the variants of the popular global feature set described in previous work by Bouchard (2011); Shihavuddin *et al.* (2013); Prévost (2015). Evidence pooling has several advantages over the *naïve* alternative, *i.e.* feature vectors concatenation: significantly better accuracy (according to our preliminary results), greatly reduced computational complexity, curse of dimensionality mitigation, parallelization possibility. The classification methodology will be presented in depth in the following section.

3.2.2.1 Normalization

For classification purposes, all components of a vector in the feature space should have similar orders of magnitudes. For this reason, normalization is systematically applied. We combine two normalization methods:

a. For textons, DTD^{IFV}_{RBF} and DeCAF we apply L1 normalization. This is the simplest normalization, and is recommended by Chang and Lin (2011).

b. The propose feature set is normalized using the standard score function, as defined in equation 3.9. The mean and standard deviations are estimated from the training set. We use this function because our proposed vector suffers from high heterogeneity. Some statistics are computed using square or power functions, which causes extreme values that are several orders of magnitude below or above the average. If a typical L1 normalization was applied, features with extreme value occurrences would take an insignificantly low value resulting in mostly ignored features in the vector.

$$X'_i = \frac{X_i - \mu_i}{\sigma_i} \tag{3.9}$$

3.3 Classification, Fusion and Rejection

In this section, we first present our methodology for training and testing our SVM classifier. Secondly, we extend the single-feature set and single-classifier method by applying MCF to aggregate information from various feature sets. Thirdly, we present a rejection method and discuss its implications.

3.3.1 SVM Training and Testing Methodology

An RBF kernel SVM is used for classification. Grid search is performed systematically to identify good (C, γ) parameters. To avoid bias and maintain good computational efficiency, the grid search validation set is a stratified subset of the training data containing at most 600 samples per class. SVM training weights are used to compensate. We found that this approach was significantly faster than using the entire training set, while still maintaining good accuracy.

AIMS is separated by groups of two years, thus creating 4 datasets: 2006/07, 2008/09, 2010/11, 2012/13. Each of these dataset will be subject to a 10-fold analysis, as it is the case for the non coral datasets (TDL and CUReT). For MLC, we use one of the three experiments proposed by Beijbom *et al.* (2012): for images from year 2008, 2/3 of the data is used for training and 1/3 for testing, sampling for the testing set is done across the entire dataset, selecting one image out of three.

For coral datasets, the multiple labeled points on any single image are always kept together in either the training or the testing sample, this avoids possible bias. In addition, because TDL and CUReT are ordered by class, the data in shuffled before classification, using a deterministic seed for repeatability.

The AIMS dataset offers multiple taxonomic ranks. Experiments are done using the coarsest "group description" scale. Appendix IV extends results to other taxonomic ranks. We also ignore classes labeled at the "group description" level with one of the following labels: "other", "N/A", "indeterminate". Given the excessive intra-variability and few sample count, we believe that these cannot be classified. However, in an operational setting, given a case where one of these labels is theoretically encountered and the system it not aware of the existence of its expertly labeled class, all possible outcomes result in a disagreement, but are not necessarily detrimental to statistics:

- The area was too dark to be expertly labeled, but very low intensity texture revealed the presence of the correct class.

- Where the expert saw a fish, or another obstructing object, the system saw a large uniformly textured patch with a statistically insignificant fraction of it covered by a different texture. This results in a class prediction as if the obstructing object was not present.

- If the texture observed is significantly different from anything previously encountered, the subsequent rejection step is likely to ignore the point, which is essentially the same as the expert labeling "other".

- In the worst-case scenario, the patch is incorrectly labeled, in which case we can only accept the error. Given then small frequency of these labels, this is an insignificant offset.

We use the following patch size to sample the texture around a point:

- **AIMS, TDL, CUReT**: 300 × 300 px around the point of interest or the center.

- **MLC**: 440 × 440 px (with an additional image resize of factor 0.5)

Figure 3.3 shows a sample of 30 randomly selected patches per class from the AIMS dataset. The challenges that were previously discussed in chapter 2 can be clearly observed: high intra-class variance, low inter-class variance, ambiguous sampling methods, etc.

3.3.2 Multiple Classifier Fusion

We've previously stated our hypothesis that combined information from four different feature sets leads to increased accuracy. To demonstrate this, we propose a flexible classifier pooling scheme based on MCF. Because not all features are equally important to describe every class, MCF are an appropriate option as opposed to popular "Boosting" methods, or *naïve* vector concatenation, since each SVM will tend to become better than the average classifier for some specific classes, and will return higher probability scores for those cases. More accurate class predictions are therefore expected when results from all SVMs are aggregated. In our method, each one of the four feature sets is used to train a unique probability estimation regression

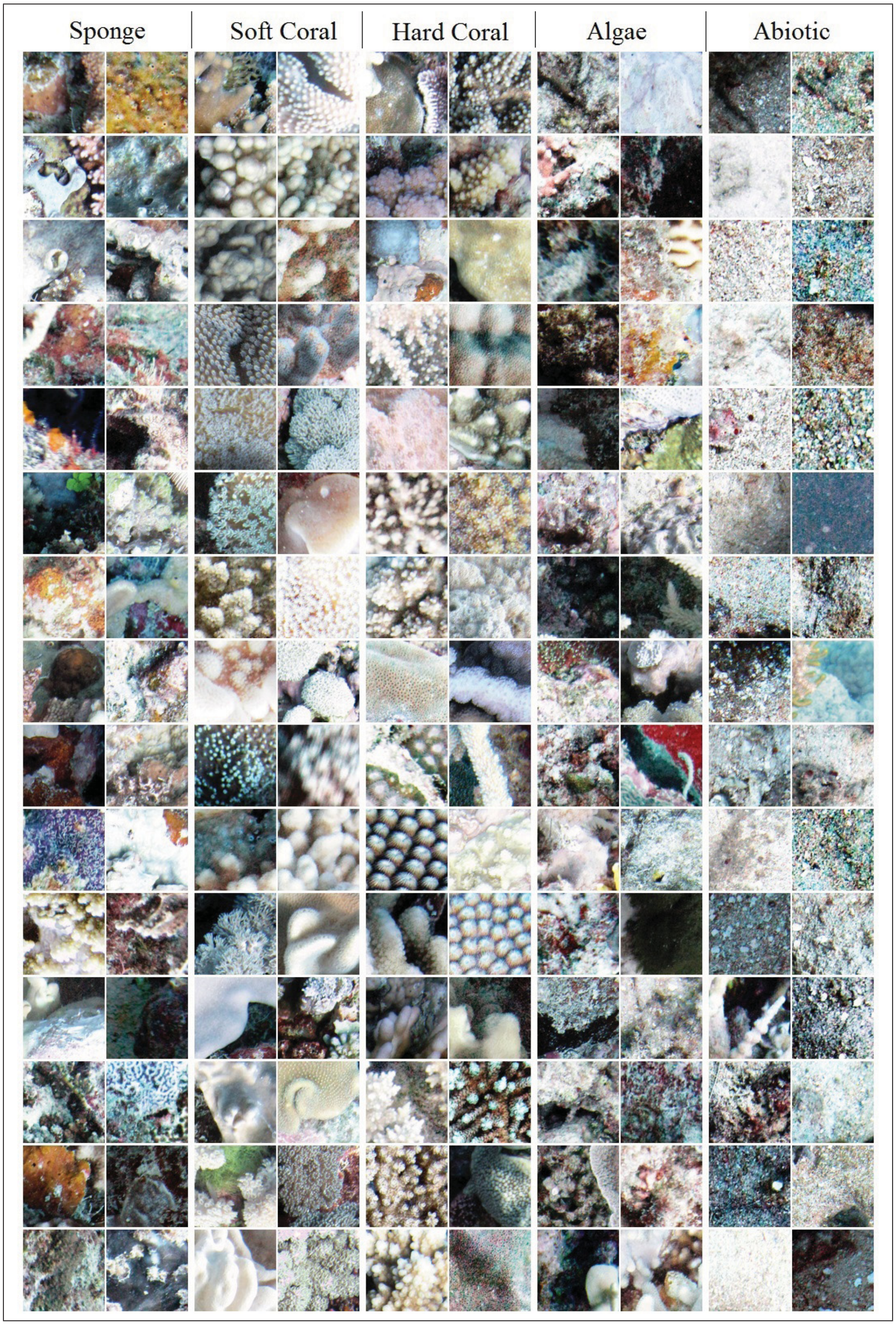


Figure 3.3 30 samples per class of the AIMS dataset at the group description level. Histogram equalization was applied to enhance visualization. From left to right: Sponge, Soft Coral, Hard Coral, Algae, Abiotic

SVM, as explained in section 3.3.1. As for the fusion function, the optimal function may be selected through additional validation. However, we found that the extra computational complexity was not worth the insignificant accuracy gain. Furthermore, we found that the class-wise product of SVM scores gave optimal results in every experiment we performed. This result is surprising, since a mean fusion function is the standard baseline choice in the literature. A product fusion is essentially the same as a sum of logarithms, and a mean fusion is the same as a sum fusion when the number of classifier is constant (as it is the case here). A mean fusion will consider all low scores to be almost equal to zero, whereas the product fusion will give much more weight to extremely low scores. We explain the relevance of the product fusion function by the way it handles extremely low scores, which are expected to occur more frequently given the high diversity of complex textures in the dataset. Figure 3.4 shows our proposed MCF pooling pipeline, and equation 3.10 defines our fusion function $\mathscr{F}$ of scores S_X of object X for an M class problem.

$$\begin{aligned} \mathscr{F}(S_x) = MAX_c \prod_i s_i^c \ \mid \ s_i^c \in S_X, \\ i \in \{Blanchet, Texton, DeCAF, DTD_{RBF}^{IFV}\} \\ c \in \{class_1, class_2, ..., class_M\} \end{aligned} \tag{3.10}$$

3.3.3 Rejection

Rejection is applied to eliminate low reliability class predictions, and increase classification rate. The MCF output is not just a class, but also a probability estimation obtained from the fusion-function-aggregated regression SVM scores. This probability score is used as a reliability measurement, and is thresholded to eliminate predictions which are likely to be wrong. This is not an error free process: some correct class predictions will be rejected (false rejection, or FR) and some classification errors will be accepted (false acceptance, or FA). However, rejection is flexible and can be tuned by setting a rejection threshold, which directly impacts

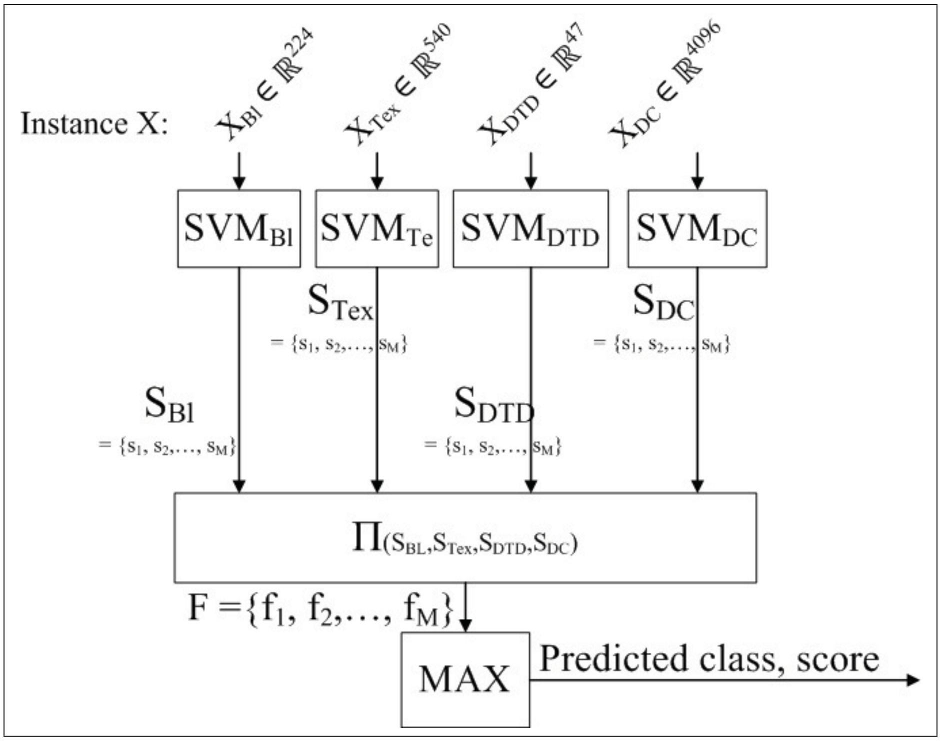

Instance X:
$X_{Bl} \in \mathbb{R}^{224}$
$X_{Tex} \in \mathbb{R}^{540}$
$X_{DTD} \in \mathbb{R}^{47}$
$X_{DC} \in \mathbb{R}^{4096}$
SVM_{Bl}
SVM_{Te}
SVM_{DTD}
SVM_{DC}
S_{Tex}
$= \{s_1, s_2, \ldots, s_M\}$
S_{DC}
$= \{s_1, s_2, \ldots, s_M\}$
S_{Bl}
$= \{s_1, s_2, \ldots, s_M\}$
S_{DTD}
$= \{s_1, s_2, \ldots, s_M\}$
$\Pi(S_{BL}, S_{Tex}, S_{DTD}, S_{DC})$
$F = \{f_1, f_2, \ldots, f_M\}$
MAX
Predicted class, score

Figure 3.4 MCF as proposed for an M class problem. Scores from the proposed features (Bl), textons (Tex), DTD_{RBF}^{IFV} (DTD) and DeCAF (DC) combined with the product fusion function.

the number of instances that are rejected: as it increases, false acceptance rate goes down, but at the cost of an increased false rejection rate (FRR). A high threshold will yield high classification rates, but the small sample size of the accepted group is unlikely to be representative of the biodiversity. While this threshold could realistically be set manually in operational mode, we settle for a simple threshold selection criterion: for each class, a constant fraction of the lowest scores predictions are rejected. One threshold is used for every class because their frequencies are highly variable and doing otherwise may eliminate completely a rare class. While rejecting a fixed fraction of every class has no impact on the resulting biodiversity data, the rationale is that these unreliable samples could then be manually inspected, which would yield high quality data. The rejection parameter should be selected accordingly with the availability of experts. Alternatively, if the intent is to generate a sample of labeled images regardless of overall biodiversity statistics, the resulting sample will be of much higher quality.

CHAPTER 4

ANALYSIS AND RESULTS

We've now presented in details our proposed computer vision system capable of predicting the benthic group of at point in a given input image. In this chapter, we start by presenting the results of the previously described experiments on feature sets aggregation using MCF in order to demonstrate the capabilities and limitations of our approach. We then explore and discuss the implications of using rejection in our system.

4.1 Features Comparison

In this first subsection, we present global classification rates averaged over ten-fold experiments involving every combination of feature sets and datasets introduced respectively in section 3.2 and chapter 2. Global classification rate is a simple overall performance metric defined by the ratio between the number of correct predictions and the number of total predictions. This is perhaps the most important metric given the problematic of biodiversity assessment. For each experiment, we also include the standard deviation observed across all folds to measure the consistency. These are all compared to the results obtained using the proposed multi-classifier fusion method, which considers information from our proposed global feature set, as well as from textons, DTD^{IFV}_{RBF}, and DeCAF.

Results are presented in three parts, which have different objectives:

a. **Popular texture benchmarks** focuses on texture datasets. The goal of these results is to provide insight on the capabilities and limitations of the different feature sets.

b. **Coral datasets** aims to demonstrate the performance of our proposed MCF method for the task of benthic image annotation.

c. **Proposed global feature set** provides evidence that justifies the introduction of our own version of the global feature set that combines popular descriptors.

4.1.1 Popular Texture Benchmarks

In section 2.4, we presented a few texture benchmark datasets and discussed their key challenges. Table 4.1 presents the average global classification rate on these two popular datasets. The proposed MCF method combines all four feature sets: Blanchet (proposed), Beijbom (textons), DTD_{RBF}^{IFV}, and DeCAF. The first four feature sets (Bouchard, Prévost, Shihavuddin, Blanchet) are all variation of the same global feature set.

Table 4.1 Global classification rate.

feature set	**TDL**	**CUReT**
Bouchard	91.9 ± 2.1	98.4 ± 0.5
Prévost	93.1 ± 1.9	99.4 ± 0.3
Shihavuddin	97.1 ± 1.8	99.7 ± 0.2
Blanchet	95.4 ± 2.5	99.6 ± 0.2
Beijbom (Textons)	74.4 ± 4.6	99.1 ± 0.3
DTD_{RBF}^{IFV}	95.0 ± 2.7	99.4 ± 0.3
DeCAF	98.4 ± 1.1	99.4 ± 0.2
Proposed MCF	**99.7** ± **0.5**	**99.96** ± **0.08**

TDL results are consistently high, meaning all features have reasonably good tolerance to challenging variances in scale and orientation. The only exception to this is the texton method, which clearly underperformed. However, this is only because the texton implementation used relies on the Lab color space, and is specifically designed for color rich coral images. Because the TDL dataset contains no color information, two of the three Lab color-space components provide no information, causing only a small fraction of all textons to be useful, and resulting in high feature space sparsity. If a new dictionary was created, textons would likely perform well. Interestingly, despite the introduction of an under-performing feature set, the proposed MCF method remained unaffected, and yielded improved results. This is a highly desirable aspect of a classifier pooling system.

CUReT results are also high in general, meaning all features are tolerant to illumination and perspective change. This is an important aspect for coral reef image annotation.

These results also support the hypothesis that the proposed MCF fusion consistently leads to better results than any feature set alone. It may be interesting to further study the effect of the proposed method on very hard state of the art datasets.

4.1.2 Coral Datasets

We now extend our experiments to coral reef datasets. In table 4.2 the global classification rates obtained on four samples of the AIMS dataset (grouped by two-year periods) are presented. As a reminder, the proposed MCF method combines all four feature sets: Blanchet (proposed), Beijbom (textons), DTD_{RBF}^{IFV}, and DeCAF. All experiments are done at the group description ranking using five classes: abiotic, algae, hard coral, soft coral, sponge.

Table 4.2 Global classification rates on the AIMS datasets for various periods.

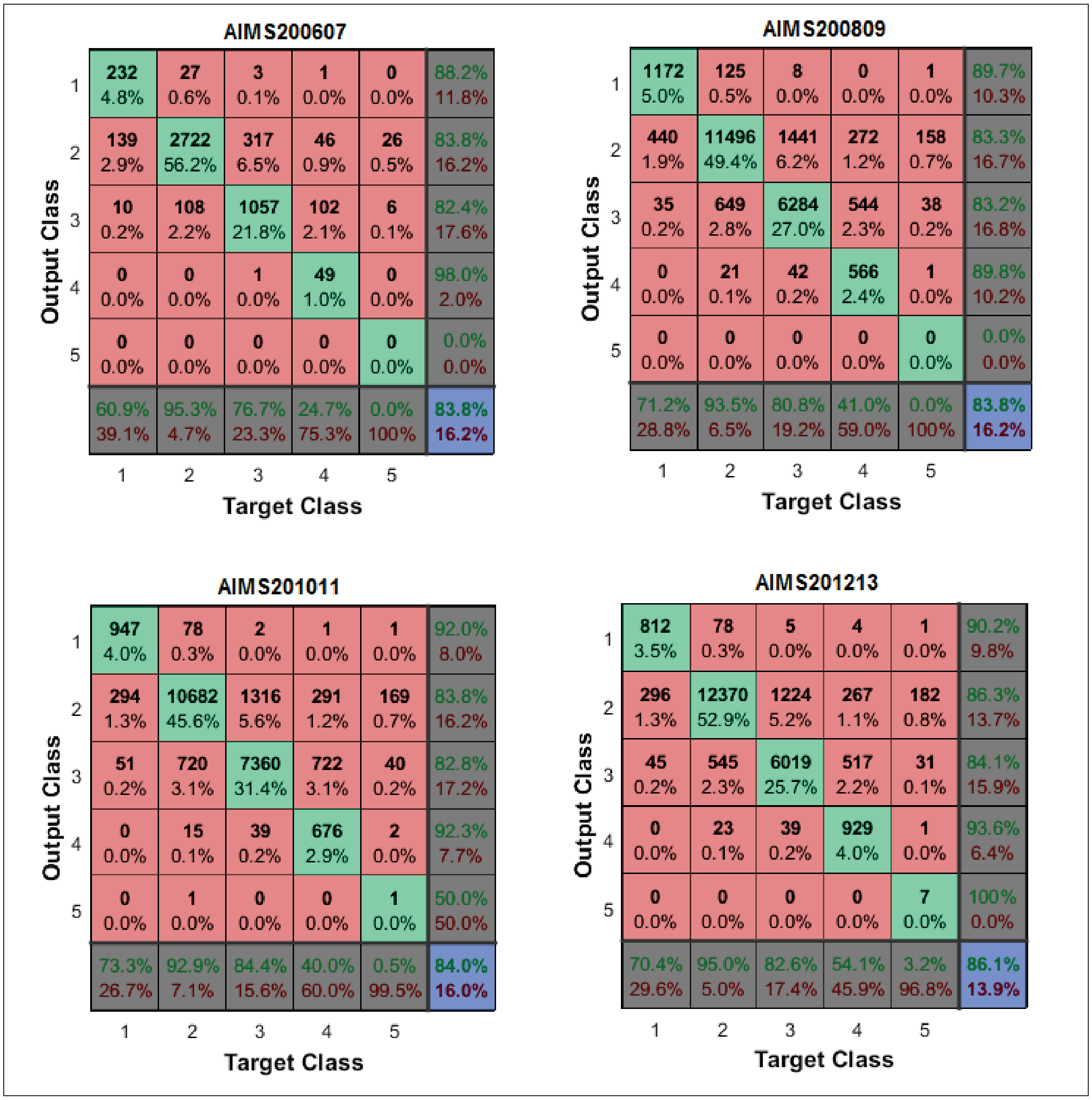
feature set	**AIMS 06-07**	**AIMS 08-09**	**AIMS 10-11**	**AIMS 12-13**
Bouchard	74.0 ± 2.8	72.1 ± 0.9	71.1 ± 1.1	73.3 ± 1.0
Prévost	80.4 ± 2.5	79.3 ± 0.8	80.5 ± 0.8	82.3 ± 0.8
Shihavuddin	80.3 ± 2.2	79.4 ± 0.7	79.2 ± 0.7	76.0 ± 0.9
Blanchet	78.5 ± 2.3	77.8 ± 0.6	77.9 ± 0.9	79.9 ± 0.8
Beijbom (Textons)	80.4 ± 2.1	82.0 ± 0.8	80.8 ± 0.8	83.8 ± 1.1
DTD_{RBF}^{IFV}	78.5 ± 1.6	78.6 ± 0.7	78.7 ± 0.9	80.5 ± 0.8
DeCAF	79.8 ± 1.3	80.7 ± 0.7	79.3 ± 0.5	82.5 ± 0.7
Proposed MCF	**83.8** ± **1**.2	**83.9** ± **0.5**	**84.2** ± **0.8**	**86.1** ± **0.8**

At first glance, while very close to other similar global feature sets, our proposed set seems to under-perform for the first few years. The slight performance gap is however within one standard deviation in the worst case. Furthermore, global classification rate only shows the absolute error, which is important when classifying using the feature set directly, but for fusion, only the SVM probabilities matter. Ambiguity should be explicit when encountered, allowing more decision weight to be given to the information provided by other feature sets. This is an important aspect that cannot be measured here, and will be addressed in the following subsection (4.1.3).

The performance of every feature set varies significantly across time, suggesting that the usefulness of each descriptor is not constant. For instance, DTD_{RBF}^{IFV}'s performance increases for the last observed period. This could be explained by variation in the acquired images' quality, or simply by the shift in biodiversity, which favors certain features. Regardless of the explanation, we've shown previously that our MCF pooling method is tolerant to information of variable reliability, and is therefore able to consistently perform well despite variable usefulness of each feature set. This demonstrates that our MCF method not only provides high accuracy, but also consistency. The low standard deviations across all folds for our MCF method support this observation.

Figure 4.1 presents the confusion matrices for the MCF experiments on all AIMS datasets. The class codes used are as follow: 1-Abiotic, 2-Algae, 3-Hard Coral, 4-Soft Coral, 5-Sponge. Each cell presents the relative and absolute number of samples from all aggregated folds for each combination of output (predicted label) and target (expert label, or ground truth) classes. The right column shows prediction accuracy, *i.e.* how likely is it that a prediction of this class is correct. The bottom row shows the target class accuracy, *i.e.* how likely is it that a sample of this class is classified correctly. The bottom right cell shows the global classification rate. Each accuracy percentage is also presented with its corresponding error. A few interesting observations can be made about these matrices:

a. The abiotic (1) class is reasonably well classified despite its low frequency. This means the features are well suited and capable of describing its various forms. It's worth noting that uncommon cases of dead coral are labeled as abiotic by expert, but will almost always be classified as hard coral by the system. From a pattern recognition point of view, it is futile to attempt the distinction between live and dead coral at this stage. This is a problem that should be addressed afterwards through further processing, but it is left to future work for now.



AIMS200607

	1	2	3	4	5	
1	232 4.8%	27 0.6%	3 0.1%	1 0.0%	0 0.0%	88.2% 11.8%
2	139 2.9%	2722 56.2%	317 6.5%	46 0.9%	26 0.5%	83.8% 16.2%
3	10 0.2%	108 2.2%	1057 21.8%	102 2.1%	6 0.1%	82.4% 17.6%
4	0 0.0%	0 0.0%	1 0.0%	49 1.0%	0 0.0%	98.0% 2.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 0.0%
	60.9% 39.1%	95.3% 4.7%	76.7% 23.3%	24.7% 75.3%	0.0% 100%	83.8% 16.2%

AIMS200809

	1	2	3	4	5	
1	1172 5.0%	125 0.5%	8 0.0%	0 0.0%	1 0.0%	89.7% 10.3%
2	440 1.9%	11496 49.4%	1441 6.2%	272 1.2%	158 0.7%	83.3% 16.7%
3	35 0.2%	649 2.8%	6284 27.0%	544 2.3%	38 0.2%	83.2% 16.8%
4	0 0.0%	21 0.1%	42 0.2%	566 2.4%	1 0.0%	89.8% 10.2%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 0.0%
	71.2% 28.8%	93.5% 6.5%	80.8% 19.2%	41.0% 59.0%	0.0% 100%	83.8% 16.2%

AIMS201011

	1	2	3	4	5	
1	947 4.0%	78 0.3%	2 0.0%	1 0.0%	1 0.0%	92.0% 8.0%
2	294 1.3%	10682 45.6%	1316 5.6%	291 1.2%	169 0.7%	83.8% 16.2%
3	51 0.2%	720 3.1%	7360 31.4%	722 3.1%	40 0.2%	82.8% 17.2%
4	0 0.0%	15 0.1%	39 0.2%	676 2.9%	2 0.0%	92.3% 7.7%
5	0 0.0%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	50.0% 50.0%
	73.3% 26.7%	92.9% 7.1%	84.4% 15.6%	40.0% 60.0%	0.5% 99.5%	84.0% 16.0%

AIMS201213

	1	2	3	4	5	
1	812 3.5%	78 0.3%	5 0.0%	4 0.0%	1 0.0%	90.2% 9.8%
2	296 1.3%	12370 52.9%	1224 5.2%	267 1.1%	182 0.8%	86.3% 13.7%
3	45 0.2%	545 2.3%	6019 25.7%	517 2.2%	31 0.1%	84.1% 15.9%
4	0 0.0%	23 0.1%	39 0.2%	929 4.0%	1 0.0%	93.6% 6.4%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.0%	100% 0.0%
	70.4% 29.6%	95.0% 5.0%	82.6% 17.4%	54.1% 45.9%	3.2% 96.8%	86.1% 13.9%

Figure 4.1 Confusion matrices using the proposed MCF method. Classes are: (1) Abiotic, (2) Algae, (3) Hard Coral, (4) Soft Coral, (5) Sponge

b. The abiotic (1) class is sometimes confused with algea (2). Part of this issue is greater than pattern recognition limitations: there are many cases where it is very difficult to separate both, even for humans, as they can look nearly identical.

c. There is also significant confusion between hard corals (3) and soft corals (4), which is not surprising as there is a high intra-class variance for both, and they tend to overlap in the feature space. However, for the last period, as more soft coral instances are available, the confusion error tends to go down. This suggests that with much more data, it would become possible to better separate these two classes.

d. Sponges (5) are in a similar situation as soft corals (4). They are uncommon and often confused with algae (2). Because of this, the classifier only makes sponge predictions for a very small region in the feature space, to ensure high reliability. However, for year 2010-2011, one of the two high-confidence sponge prediction turned out to be wrong. This indicates that the sponge class region of the feature space is highly dominated by other classes. The limitation lies within the features, as they cannot separate the sponge (5) class from the other classes. This could also be caused by the lack of segmentation: sponges can often be fairly small, and only cover part of the extracted patch, and therefore have their features mixed with the ones of surrounding algae and corals, hence the confusion in the feature space.

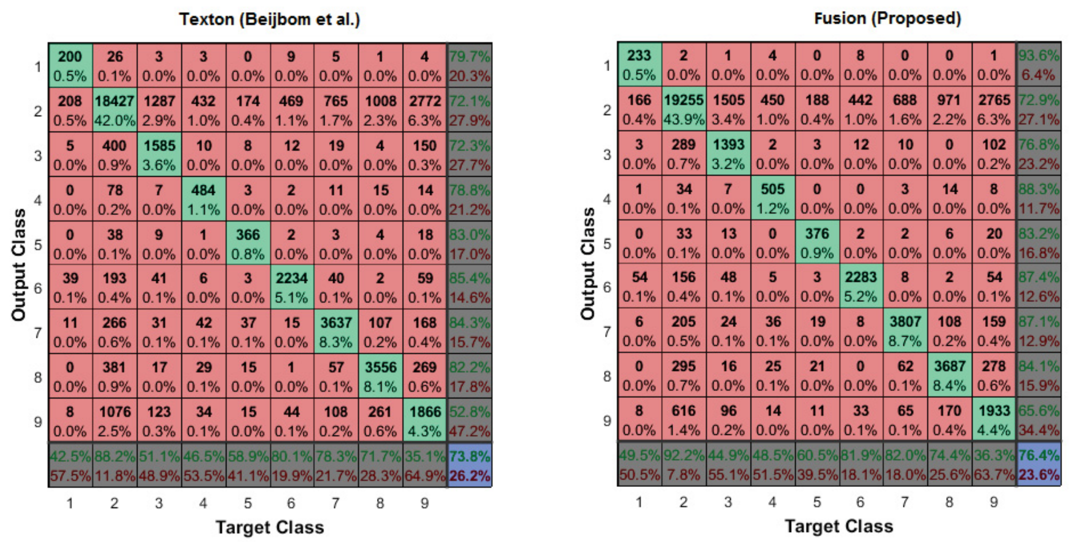
e. Algae (2) are very common and take many different forms, they cover a large area in the feature space, and as a result, all other classes are often confused with Algae (2). Segmentation would surely help better define the class.

Table 4.3 presents similar results obtained on the MLC dataset. Because of the great number of samples, only a single fold is performed, and no standard deviation is therefore available.

Table 4.3 Global classifications rate on the MLC datasets for year 2008, as proposed by Beijbom *et al.* (2012).

feature set	**MLC2008**
Bouchard	58.2
Blanchet	69.2
Beijbom (Textons)	73.8
DTD_{RBF}^{IFV}	65.3
DeCAF	62.1
Proposed MCF	**76.4**

We did not run two of the feature sets, due to high computational cost. Furthermore, we found that DeCAF and DTD_{RBF}^{IFV} were performing poorly on the MLC dataset. This could be explained by the fact that these images contain a significant amount of background information such as the orange rope, the quadrat and the white tape. Both of these feature sets are sensitive to such foreground objects. Perhaps proper segmentation would improve their performance. We also did not experiment with the window size, and instead followed the dataset's author's methodology. The patch size may be unsuitable for non texton features. For this reason, we eliminated the two under performing feature sets in our fusion, and kept only our global feature set in combination with textons. While this is a limitation for now, future work will investigate segmentation as a solution. Figure 4.2 presents the confusion matrices obtained using the state of the art method by Beijbom, and our proposed fusion method.



Texton (Beijbom et al.)

Output Class

	1	2	3	4	5	6	7	8	9	
1	200 0.5%	26 0.1%	3 0.0%	3 0.0%	0 0.0%	9 0.0%	5 0.0%	1 0.0%	4 0.0%	79.7% 20.3%
2	208 0.5%	18427 42.0%	1287 2.9%	432 1.0%	174 0.4%	469 1.1%	765 1.7%	1008 2.3%	2772 6.3%	72.1% 27.9%
3	5 0.0%	400 0.9%	1585 3.6%	10 0.0%	8 0.0%	12 0.0%	19 0.0%	4 0.0%	150 0.3%	72.3% 27.7%
4	0 0.0%	78 0.2%	7 0.0%	484 1.1%	3 0.0%	2 0.0%	11 0.0%	15 0.0%	14 0.0%	78.8% 21.2%
5	0 0.0%	38 0.1%	9 0.0%	1 0.0%	366 0.8%	2 0.0%	3 0.0%	4 0.0%	18 0.0%	83.0% 17.0%
6	39 0.1%	193 0.4%	41 0.1%	6 0.0%	3 0.0%	2234 5.1%	40 0.1%	2 0.0%	59 0.1%	85.4% 14.6%
7	11 0.0%	266 0.6%	31 0.1%	42 0.1%	37 0.1%	15 0.0%	3637 8.3%	107 0.2%	168 0.4%	84.3% 15.7%
8	0 0.0%	381 0.9%	17 0.0%	29 0.1%	15 0.0%	1 0.0%	57 0.1%	3556 8.1%	269 0.6%	82.2% 17.8%
9	8 0.0%	1076 2.5%	123 0.3%	34 0.1%	15 0.0%	44 0.1%	108 0.2%	261 0.6%	1866 4.3%	52.8% 47.2%
	42.5% 57.5%	88.2% 11.8%	51.1% 48.9%	46.5% 53.5%	58.9% 41.1%	80.1% 19.9%	78.3% 21.7%	71.7% 28.3%	35.1% 64.9%	73.8% 26.2%

Target Class

Fusion (Proposed)

Output Class

	1	2	3	4	5	6	7	8	9	
1	233 0.5%	2 0.0%	1 0.0%	4 0.0%	0 0.0%	8 0.0%	0 0.0%	0 0.0%	1 0.0%	93.6% 6.4%
2	166 0.4%	19255 43.9%	1505 3.4%	450 1.0%	188 0.4%	442 1.0%	688 1.6%	971 2.2%	2765 6.3%	72.9% 27.1%
3	3 0.0%	289 0.7%	1393 3.2%	2 0.0%	3 0.0%	12 0.0%	10 0.0%	0 0.0%	102 0.2%	76.8% 23.2%
4	1 0.0%	34 0.1%	7 0.0%	505 1.2%	0 0.0%	0 0.0%	3 0.0%	14 0.0%	8 0.0%	88.3% 11.7%
5	0 0.0%	33 0.1%	13 0.0%	0 0.0%	376 0.9%	2 0.0%	2 0.0%	6 0.0%	20 0.0%	83.2% 16.8%
6	54 0.1%	156 0.4%	48 0.1%	5 0.0%	3 0.0%	2283 5.2%	8 0.0%	2 0.0%	54 0.1%	87.4% 12.6%
7	6 0.0%	205 0.5%	24 0.1%	36 0.1%	19 0.0%	8 0.0%	3807 8.7%	108 0.2%	159 0.4%	87.1% 12.9%
8	0 0.0%	295 0.7%	16 0.0%	25 0.1%	21 0.0%	0 0.0%	62 0.1%	3687 8.4%	278 0.6%	84.1% 15.9%
9	8 0.0%	616 1.4%	96 0.2%	14 0.0%	11 0.0%	33 0.1%	65 0.1%	170 0.4%	1933 4.4%	65.6% 34.4%
	49.5% 50.5%	92.2% 7.8%	44.9% 55.1%	48.5% 51.5%	60.5% 39.5%	81.9% 18.1%	82.0% 18.0%	74.4% 25.6%	36.3% 63.7%	76.4% 23.6%

Target Class

Figure 4.2 Confusion matrices on MLC2008 using the state-of-the-art texton method by *Beijbom et al.*, and the extended MCF method.

4.1.3 Proposed Global Feature Set

We've previously proposed and tested a feature set based on popular global descriptors. Because simple global classification rates cannot justify the usefulness of the proposed features, we elaborate in this subsection on its purpose by demonstrating that it is more suited for MCF, and it is considerably more computationally efficient.

Any features used for fusion should return low confidence results when uncertain, because its goal is simply to provide a solid probability estimation, while the class output is irrelevant. To demonstrate that our proposed vector is suited for this purpose, table 4.4 presents the results with other global feature set variations from the literature.

Table 4.4 Global classification rate of the proposed MCF method using different versions of the global feature set.

feature set	**AIMS 06-07**	**AIMS 08-09**	**AIMS 10-11**	**AIMS 12-13**
No global feature set	83.6 ± 1.7	83.9 ± 0.6	84.0 ± 0.7	86.1 ± 0.8
Bouchard	83.2 ± 1.4	83.3 ± 0.5	83.2 ± 0.9	85.2 ± 0.8
Prévost	84.0 ± 1.3	84.0 ± 0.7	84.3 ± 0.7	86.3 ± 0.7
Shihavuddin	83.7 ± 1.5	83.8 ± 0.6	84.0 ± 0.7	85.8 ± 0.8
Blanchet	83.8 ± 1.2	83.9 ± 0.5	84.2 ± 0.8	86.1 ± 0.8

It becomes clear that the impact of the global feature set is relatively subtle: it provides a slight accuracy and consistency gain. While the proposed feature set seemed under performing earlier, it now compares well with other approaches. Furthermore, the only better alternative, Prévost, comes at the cost of unreasonable computation times. Figure 4.3 shows the computation time observed (averaged over 10 samples) for the extraction of various feature sets. We argue that the slight accuracy gain is not worth the efficiency trade-off. The importance of computational performance will be further discussed in the following chapter.

4.2 Rejection

Given that the proposed MCF method returns not only a class, but also a fusion score, rejection can be applied to discard lower reliability predictions. As we've explained earlier, rejection needs to be applied carefully: FA and FR both impact the biodiversity estimation. For practicality, we will limit our initial study of rejection to years 2012-2013 of the AIMS dataset. The ROC curve presented in figure 4.4 shows the false acceptance rate (FAR) and true acceptance

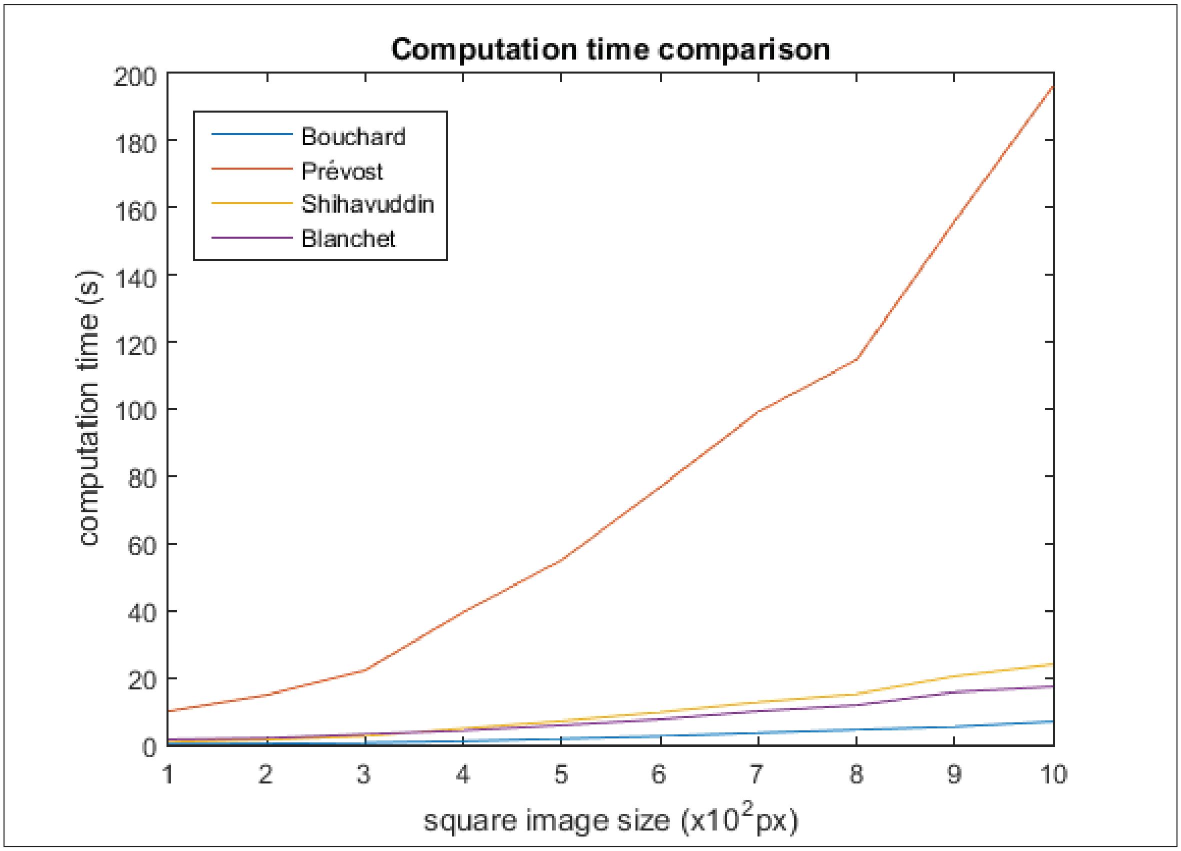


Figure 4.3 Execution time by patch size of the following global feature sets: Bouchard, Prévost, Shihavuddin, Blanchet.

rate (TAR) for all possible thresholds. Because each class comes with scores in a different value range, classes need to be studied separately, hence one ROC curve per class.

Rejection is slightly more difficult to apply correctly for the soft coral class. While there seems to be a weak correlation between the score output and the likelihood of a correct prediction, this could be explained simply by the fact that soft corals are still difficult to separate from hard corals. And because soft coral are much rarer, this leads to few cases of soft coral false predictions. In other words, there are few errors for the soft coral class, and these are hard to identify. This can be observed in figure 4.5. This suggests that it may be better not to apply rejection for classes that are rare and difficult to model. Errors within the algae and abiotic classes, on the other hand, can be effectively rejected. There seems to be a point around 25% FAR where every new true acceptance (TA) comes at the cost of a new error. For the

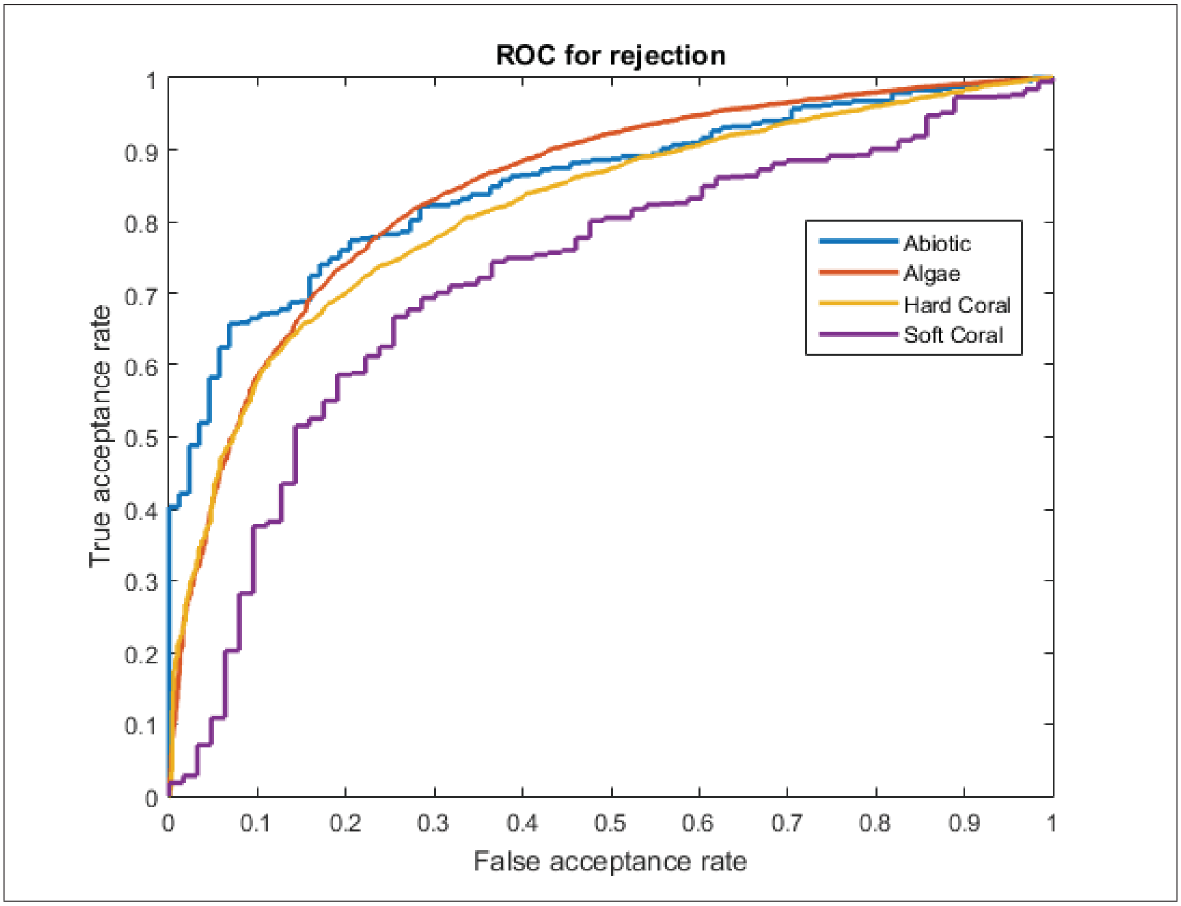


Figure 4.4 Class-wise ROC curves for rejection on MCF AIMS201213 experiment.

algae class, this could be explained by the high a priori probability of this class: because algae are dominant, an ambiguous sample is likely to be algae above everything else, while still remaining somewhat different. These ambiguous cases end up with the algae label, and can be isolated fairly well. In the case of the abiotic class, it is more homogeneous and easier to model than other classes, which unsurprisingly leads to better error rejection.

It is difficult to provide optimal thresholds, as rejection may affect the biodiversity statistics and should consider the problem at hand. However, given that experts are available to review, and manually correct the rejected data, we can reject a fraction of the lowest scores for each class. This fraction depends on the availability of experts, but this could be a very effective

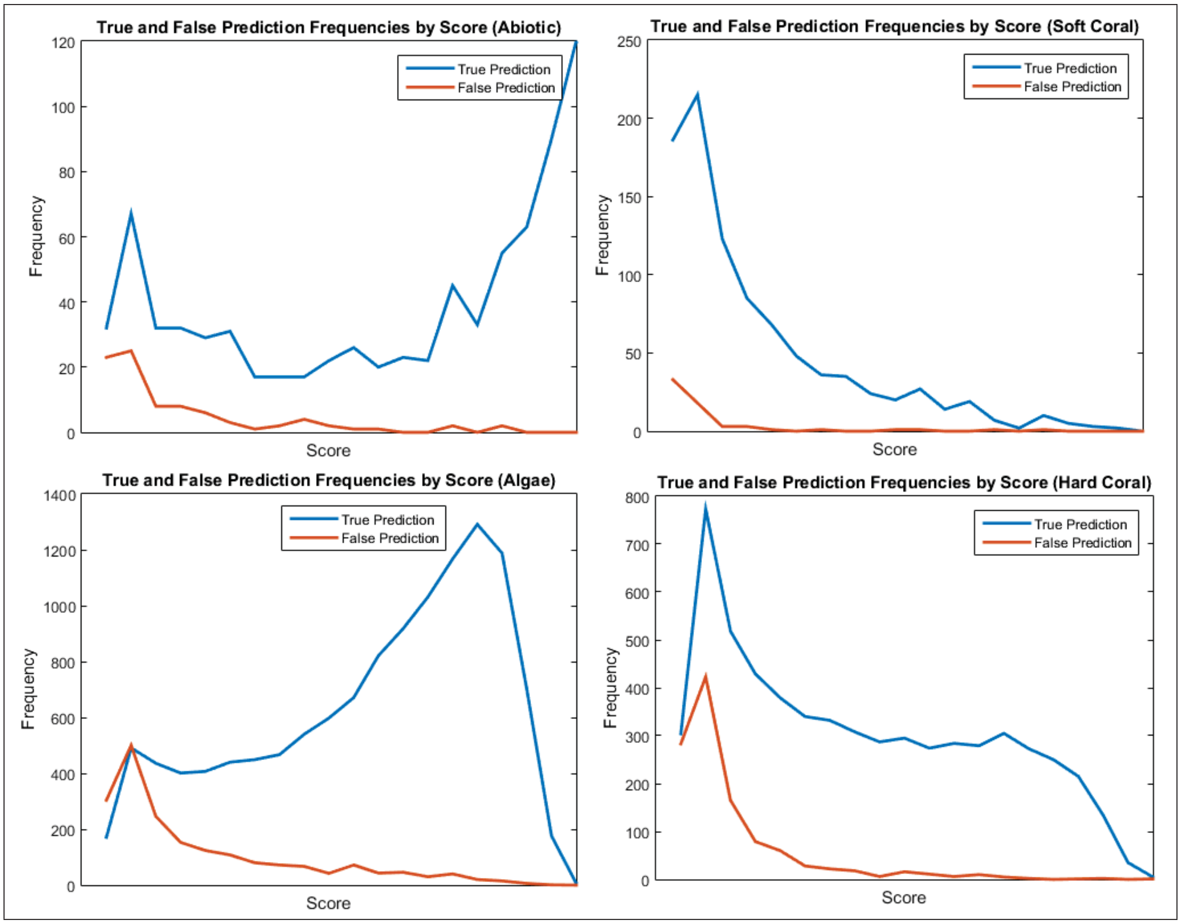


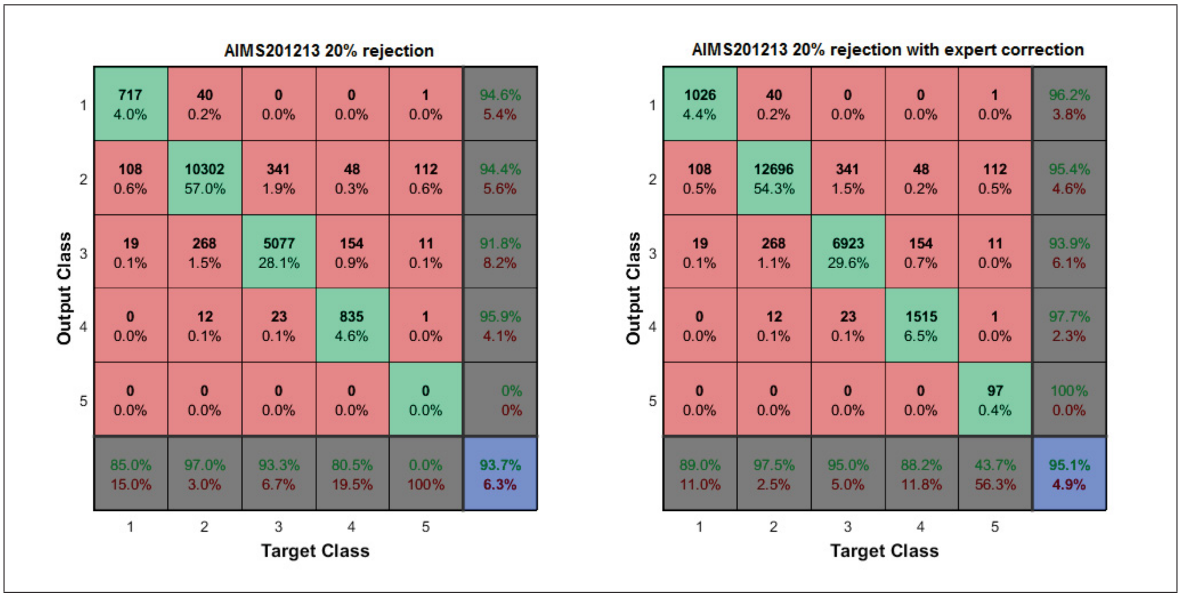
Figure 4.5 Frequency of all scores for the main four classes of the MCF AIMS201213 experiment. Sponges are not shown as there are not false sponge predictions for that experiment.

way to improve performance. Table 4.5 shows classification rates obtained after rejections of 5, 10 and 20%, and the results obtained after the introduction of expertly corrected data.

Table 4.5 Global classification rates obtained after rejection (R) and expert correction of the rejected samples (R+C).

Dataset	No rejection	5% R	5% R+C	10% R	10% R+C	20% R	20% R+C
AIMS200607	83.8	86.2	87.2	88.4	89.9	91.2	93.2
AIMS200809	83.9	86.2	87.1	88.4	89.4	91.5	93.5
AIMS201011	84.2	86.9	87.7	88.8	90.1	91.9	93.7
AIMS201213	86.1	89.0	89.7	91.0	92.1	93.7	95.1
MLC2008	76.4	78.6	79.9	80.5	82.8	83.9	87.6

Confusion matrices for AIMS2012-2013 with 20% rejection rate without and with expert correction are presented in figure 4.6. It can be seen that very high performance can be achieved by soliciting the expert for only one fifth of the original work load. In the following chapter, we elaborate on this idea as we discuss the possible applications in operational settings.



AIMS201213 20% rejection

Output Class	1	2	3	4	5	
1	717 4.0%	40 0.2%	0 0.0%	0 0.0%	1 0.0%	94.6% 5.4%
2	108 0.6%	10302 57.0%	341 1.9%	48 0.3%	112 0.6%	94.4% 5.6%
3	19 0.1%	268 1.5%	5077 28.1%	154 0.9%	11 0.1%	91.8% 8.2%
4	0 0.0%	12 0.1%	23 0.1%	835 4.6%	1 0.0%	95.9% 4.1%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0% 0%
	85.0% 15.0%	97.0% 3.0%	93.3% 6.7%	80.5% 19.5%	0.0% 100%	93.7% 6.3%

Target Class

AIMS201213 20% rejection with expert correction

Output Class	1	2	3	4	5	
1	1026 4.4%	40 0.2%	0 0.0%	0 0.0%	1 0.0%	96.2% 3.8%
2	108 0.5%	12696 54.3%	341 1.5%	48 0.2%	112 0.5%	95.4% 4.6%
3	19 0.1%	268 1.1%	6923 29.6%	154 0.7%	11 0.0%	93.9% 6.1%
4	0 0.0%	12 0.1%	23 0.1%	1515 6.5%	1 0.0%	97.7% 2.3%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97 0.4%	100% 0.0%
	89.0% 11.0%	97.5% 2.5%	95.0% 5.0%	88.2% 11.8%	43.7% 56.3%	95.1% 4.9%

Target Class

Figure 4.6 Confusion matrices obtained after rejecting the lowest 20% prediction scores (left) and after expert correction of the rejected samples (right) for AIMS2012-2013.

CHAPTER 5

DISCUSSION AND CONCLUSION

In this work, we've presented a system capable of predicting the benthic group of an image's content in any given point. The system used multiple support vector machines fusion to aggregate probability estimations obtained based on four different feature sets from the literature. We've shown that our proposed system outperforms any single feature set. We also applied rejection, and presented results suggesting that images can be labeled with an accuracy of over 95% when soliciting the expert to correct ambiguous labels. In this section, we expand on possible applications of our system in an operational setting, and on future work to improve this system.

5.1 Possible applications

While our proposed system is limited to simple class predictions, it can effectively be integrated into a much more complex system as a functionality to accomplish a variety of tasks. In this section, we discuss possible applications of the proposed system. Furthermore, we've shortly introduced the idea that performance is a desirable characteristic of the system. We argue that most of these proposed applications are much more practical with a reasonable computational times.

- **Fully automated biodiversity preliminary survey**. Assuming a model was trained *a priori* on many images, the system can be used to obtain preliminary biodiversity estimations at a very low cost. Any number of points per image can be automatically labeled. This could also serve the purpose of identifying reefs of interest among a very large database *e.g.* areas that suffer from extreme and sudden biodiversity loss, or areas where algae are dominating.
- **Fully automated species search.** The proposed system can be used to automatically find samples of a specific class. As shown in appendix IV, finer classification (lower taxonomic

rank) can still be performed with a reasonable accuracy. If one is interested is finding examples of a rare class (*i.e.* a rare species), the system can provide a list of the candidates images with the highest probability score where this class is likely to be found.

- **Semi-supervised biodiversity statistics**. As we've shown, collaboration between the expert and our system may be the best option to efficiently generate high quality biodiversity data. A system could be designed where automatic classification is performed, and the expert is solicited to correct a fraction of the most ambiguous predictions. Based on our experiments, this leads to high quality data.

- **Semi-supervised interactive image annotation tool**. Our system could also be used to enhance the current expert's user experience (UX) when performing annotation on thousands of images, therefore increasing productivity significantly. The system could handle a large portion of the points with high accuracy. The expert's opinion could be queried only when ambiguity is encountered. Furthermore, the system could learn in real time from the corrected samples, and deliver transect-specific class predictions based on the expert's corrected opinion. Also, because a good portion of the errors are due to point ambiguity (located in between two or more classes), the expert could be given the option of providing an explicitly segmented area, therefore eliminating any point related ambiguity, allowing the system to automatically output the corrected class, without solicitation from experts, as navigating through complex class-selection menus can be time-inefficient. This also brings up the point of performance: this promising approach can only be applied if computation times are reasonably low, otherwise defeating the purpose of an enhanced user experience.

- **Semi-supervised detailed surface estimation**. Though this would required further work, it is not unreasonable to say that the proposed system could be adapted to perform detailed surface estimation on any image (*i.e.* computing the area belonging to each species on a single image), given that the existing classes were all expertly identified in at least one point. With a few changes in the manual annotation protocol, this could lead to considerably more data for roughly the same work load.

5.2 Future work

In this work, we've discussed many aspects of the system that were not implemented, nor tested. In this section, we provide a short list of the most promising addition to our system:

- **Segmentation** is perhaps the most important single improvement to come. Appendix I provides evidence of the potential gain linked with the introduction of proper segmentation. Appendix V expands on the topic by showing our most promising attempt at fully automated segmentation. However, we were not able to improve global classification rate using any segmentation method. We argue this is a result of the single-point annotation methodology. Despite our segmentation method consistently isolating a single-textured area, it often fails to match the expert's intentions, rendering the ground truth unusable. A better way to assess the quality of segmentation would be to directly ask the expert to rate the correctness of the area as well as the predicted class associated with the segmented region. We believe improvement through segmentation is no longer a matter of developing the right segmentation algorithms, but of correctly validating the results.
- **Features** leave much room for improvement. During this study, we used only four feature sets which all turned out to work quite well. Our method can easily be extended to use additional features. This is likely to improve classification rates.
- **Deep learning** classification is a highly promising emerging technology. It is known to perform well for problems with large volumes of complex data.
- **Rejection** was applied using a simple class-wise threshold based on the score percentile ranks. As we've briefly discussed in the literature review, there are more advanced methods for detecting not only ambiguity, but also novelty, which could be very interesting in a setting where a complex taxonomy may change periodically.
- **Graphical processing unit (GPU) acceleration** would not only help significantly in an operational setting, but would also help speed up research, by allowing quick feedback on

the viability of new methods. Recent technologies have applied GPU acceleration to both image processing and SVM classification.

- **Scalable classification architecture**: in an operational setting, the number of class is likely to grow over time. Currently, the only way to support new classes is by regenerating the entire classifier. A good alternative to this would be to investigate multi-classifier architectures, similar to what is being done in biometrics areas, *i.e.* one classifier per class.

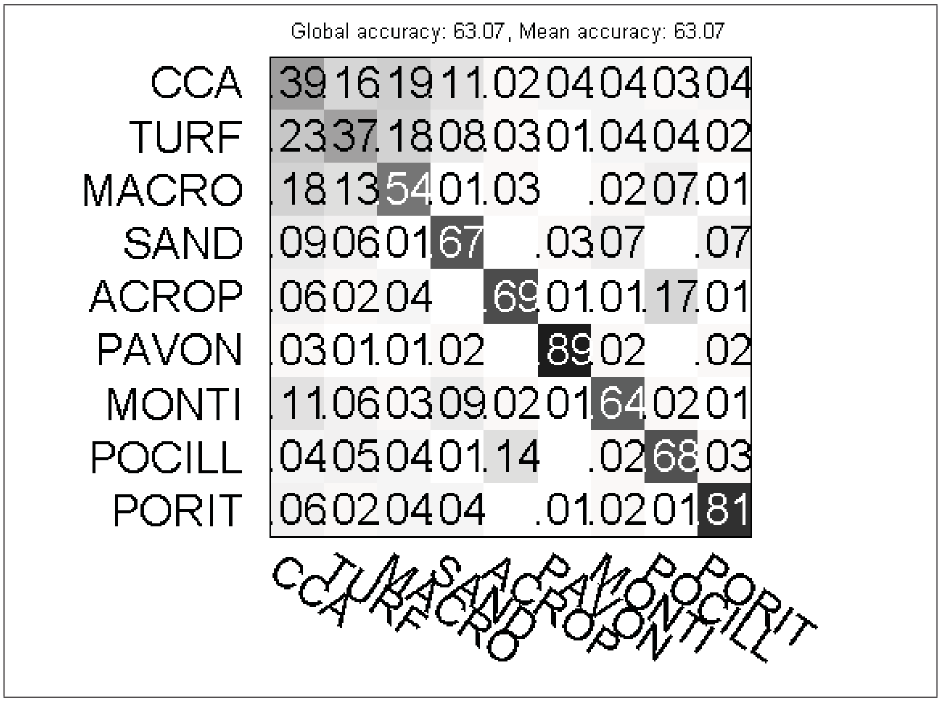
APPENDIX I

IMPROVING CLASSIFICATION RATES USING SEGMENTATION

Figure I-1 and I-2 show confusion matrices of a 10-fold analysis of 1665 randomly selected points across the Moorea Labeled Corals dataset (Edmunds *et al.*, 2012) for the 2008 year using an older version of our previously defined global feature set. Cluster sampling was used to select 200 instances of each one of the nine classes, and manual non-expert inspection of all the points lead to the rejection of 235 points which were considered unusable for machine learning (texture area too small, or non representative of the class). Figure I-3 shows the distribution per class of the resulting sample. The first confusion matrix presents the results when classifying fixed size patches, which typically contain considerable irrelevant background information, or multiple classes per patch. The second sample shows results with manual, non-expertly cropped homogeneous rectangle regions around a point. Figure I-4 shows a screen capture of the tool used to perform this task.

Though segmentation of corals, to the best of our knowledge, was never successfully performed in an automated way, this experiment leads to two conclusions: (1) it is possible to apply a segmentation to improve the classification accuracy, (2) coarse segmentation during the manual annotation task leads to a much better automated classification (likely due to the absence of ambiguity). Future work will attempt to automate this segmentation process. Appendix V introduces our most promising attempt at automated segmentation.

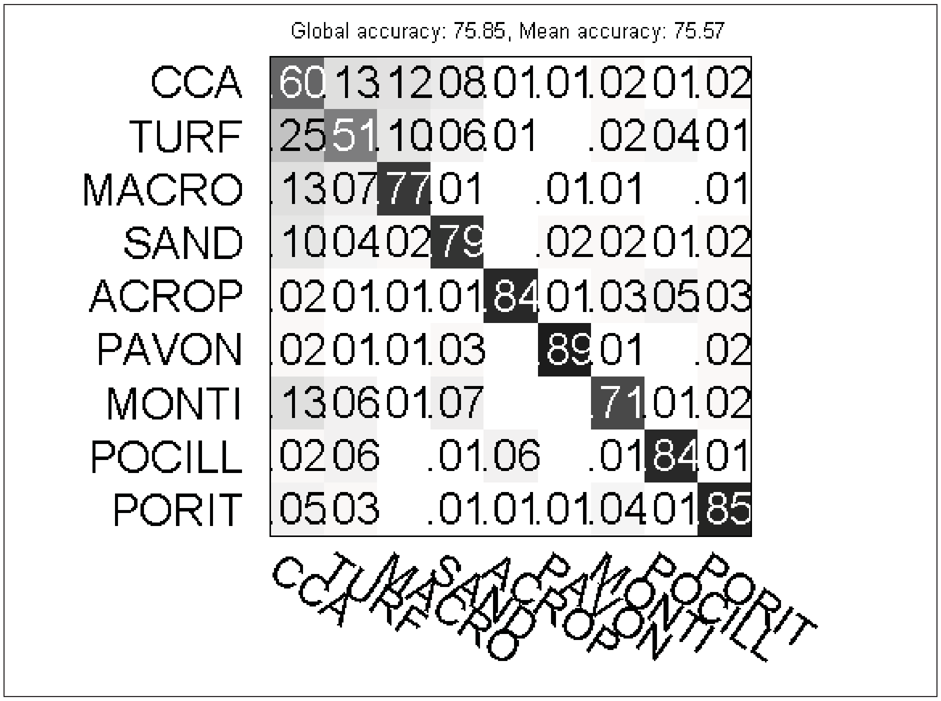
It is also important to note that none of the improvements discussed in this work are applied to this experiment (preprocessing, feature extraction, classification, rejection).



Global accuracy: 63.07, Mean accuracy: 63.07

	CCA	TURF	MACRO	SAND	ACROP	PAVON	MONTI	POCILL	PORIT
CCA	.39	.16	.19	.11	.02	.04	.04	.03	.04
TURF	.23	.37	.18	.08	.03	.01	.04	.04	.02
MACRO	.18	.13	.54	.01	.03		.02	.07	.01
SAND	.09	.06	.01	.67		.03	.07		.07
ACROP	.06	.02	.04		.69	.01	.01	.17	.01
PAVON	.03	.01	.01	.02		.89	.02		.02
MONTI	.11	.06	.03	.09	.02	.01	.64	.02	.01
POCILL	.04	.05	.04	.01	.14		.02	.68	.03
PORIT	.06	.02	.04	.04		.01	.02	.01	.81

Figure-A I-1 Confusion matrix for regular fixed-size square patches on a sample of MLC 2008



Global accuracy: 75.85, Mean accuracy: 75.57

	CCA	TURF	MACRO	SAND	ACROP	PAVON	MONTI	POCILL	PORIT
CCA	.60	.13	.12	.08	.01	.01	.02	.01	.02
TURF	.25	.51	.10	.06	.01		.02	.04	.01
MACRO	.13	.07	.77	.01		.01	.01		.01
SAND	.10	.04	.02	.79		.02	.02	.01	.02
ACROP	.02	.01	.01	.01	.84	.01	.03	.05	.03
PAVON	.02	.01	.01	.03		.89	.01		.02
MONTI	.13	.06	.01	.07			.71	.01	.02
POCILL	.02	.06		.01	.06		.01	.84	.01
PORIT	.05	.03		.01	.01	.01	.04	.01	.85

Figure-A I-2 Confusion matrix for manually cropped rectangle patches on a sample of MLC 2008

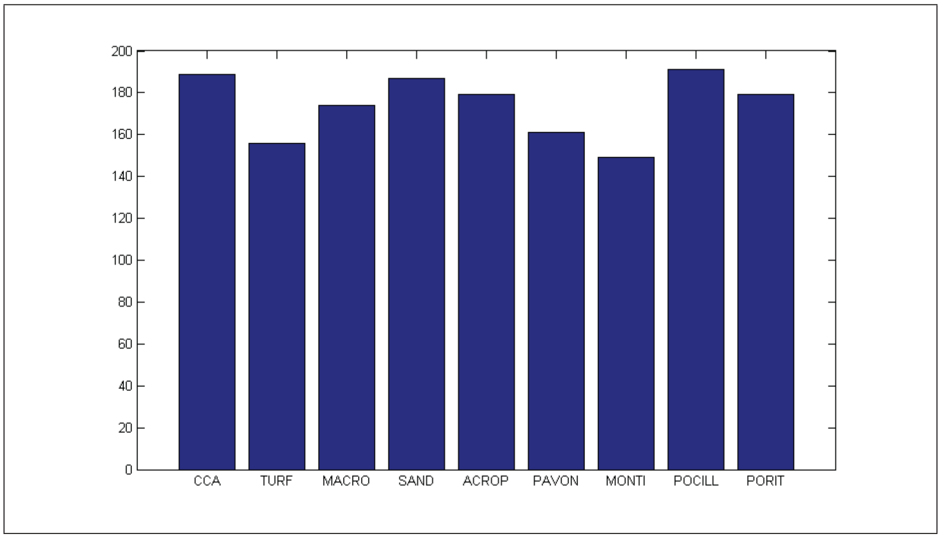
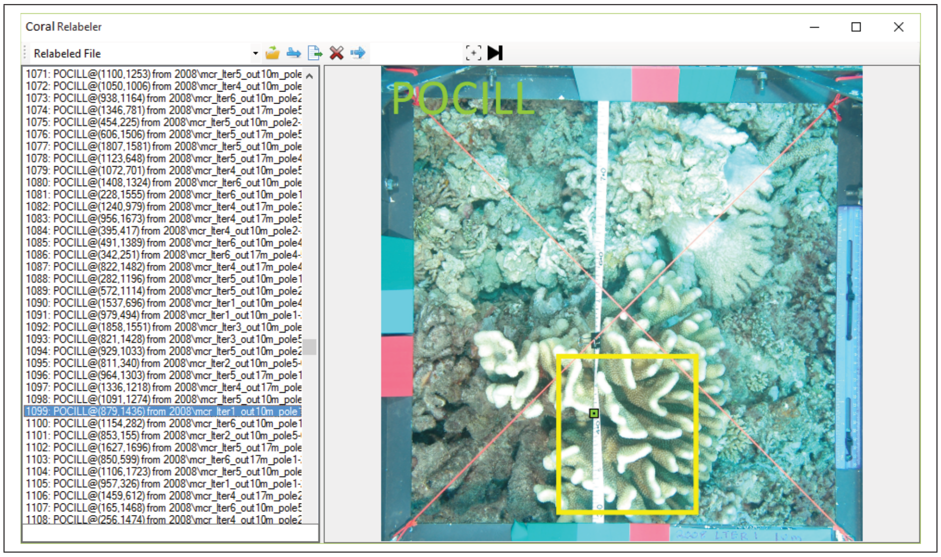


Figure-A I-3 Class distribution after manual rejection.



```
1071: POCILL@(1100,1253) from 2008\mcr_lter5_out10m_pole
1072: POCILL@(1050,1006) from 2008\mcr_lter4_out10m_pole
1073: POCILL@(938,1164) from 2008\mcr_lter6_out10m_pole2
1074: POCILL@(1346,781) from 2008\mcr_lter5_out17m_pole5
1075: POCILL@(454,225) from 2008\mcr_lter5_out10m_pole2-
1076: POCILL@(606,1506) from 2008\mcr_lter5_out17m_pole5
1077: POCILL@(1807,1581) from 2008\mcr_lter5_out10m_pole
1078: POCILL@(1123,648) from 2008\mcr_lter5_out17m_pole4
1079: POCILL@(1072,701) from 2008\mcr_lter4_out10m_pole5
1080: POCILL@(1408,1324) from 2008\mcr_lter6_out10m_pole
1081: POCILL@(228,1555) from 2008\mcr_lter6_out10m_pole1
1082: POCILL@(1240,979) from 2008\mcr_lter4_out17m_pole3
1083: POCILL@(956,1673) from 2008\mcr_lter4_out17m_pole5
1084: POCILL@(395,417) from 2008\mcr_lter4_out10m_pole2-
1085: POCILL@(491,1389) from 2008\mcr_lter6_out10m_pole4
1086: POCILL@(342,251) from 2008\mcr_lter6_out17m_pole4-
1087: POCILL@(822,1482) from 2008\mcr_lter4_out17m_pole4
1088: POCILL@(282,1196) from 2008\mcr_lter5_out10m_pole1
1089: POCILL@(572,1114) from 2008\mcr_lter5_out10m_pole2
1090: POCILL@(1537,696) from 2008\mcr_lter1_out10m_pole4
1091: POCILL@(979,494) from 2008\mcr_lter1_out10m_pole1-
1092: POCILL@(1858,1551) from 2008\mcr_lter3_out10m_pole
1093: POCILL@(821,1428) from 2008\mcr_lter3_out10m_pole5
1094: POCILL@(929,1033) from 2008\mcr_lter5_out10m_pole2
1095: POCILL@(811,340) from 2008\mcr_lter2_out10m_pole5-
1096: POCILL@(964,1303) from 2008\mcr_lter5_out17m_pole1
1097: POCILL@(1336,1218) from 2008\mcr_lter4_out17m_pole
1098: POCILL@(1091,1274) from 2008\mcr_lter5_out10m_pole
1099: POCILL@(879,1436) from 2008\mcr_lter1_out10m_pole1
1100: POCILL@(1154,282) from 2008\mcr_lter6_out10m_pole1
1101: POCILL@(853,155) from 2008\mcr_lter2_out10m_pole5-
1102: POCILL@(1627,1696) from 2008\mcr_lter5_out17m_pole
1103: POCILL@(850,599) from 2008\mcr_lter6_out17m_pole1-
1104: POCILL@(1106,1723) from 2008\mcr_lter5_out10m_pole
1105: POCILL@(957,326) from 2008\mcr_lter1_out10m_pole1-
1106: POCILL@(1459,612) from 2008\mcr_lter4_out17m_pole2
1107: POCILL@(165,1468) from 2008\mcr_lter6_out10m_pole5
1108: POCILL@(256,1474) from 2008\mcr_lter4_out10m_pole2
```

Figure-A I-4 Coral labeling tool used for this experiment.

APPENDIX II

AN INTRODUCTION TO THE SIFT DESCRIPTOR (SCALE INVARIANT FEATURE TRANSFORM)

The SIFT descriptor was introduced by Lowe (2004) and is a popular local image descriptor designed "for robustness to lighting variations and small positional shifts", as explained by Grauman and Leibe (2010). It has been widely applied in object recognition. The following appendix presents the SIFT extraction pipeline.

0.1 Key Point Detection

The first step is to detect key points in the image in a way that the same key points will be identified on many images of the same object, regardless of orientation, scale, noise, etc. Though there are more, two popular key point detection algorithms exist:

a. The **Hessian detector** applies a threshold on the second derivative. Local maxima are considered key points.

b. The **Harris detector** finds key points defined as "points that have locally maximal self-matching precision under translational least-squares template matching". Unlike the Hessian detector, it will typically be more sensitive to corners. Figure II-1 shows an example of the key points found using the Harris detector.

0.2 Scale-Invariant or Affine-Invariant Region Extraction

The second step aims to define regions, or more precisely a scale at which features will be extracted around each key point. Many methods exist for this, we introduce here three methods:

a. **Automatic scale selection** computes for each key point a signature representing the local neighborhood. The signature function is designed to preserve its general shape regardless of scale. The local maxima in the signature can be used as indicator of the scale.

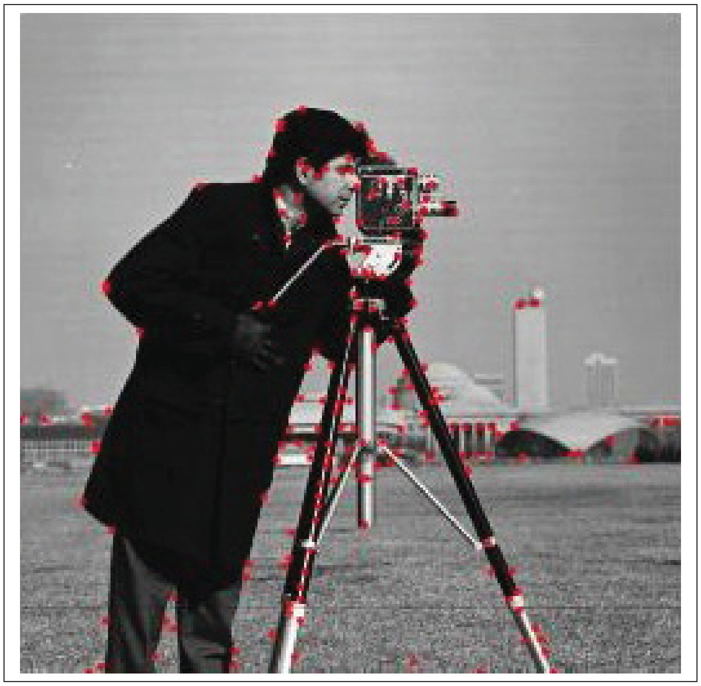


Figure-A II-1   Key points found using the Harris detector on the cameraman image.

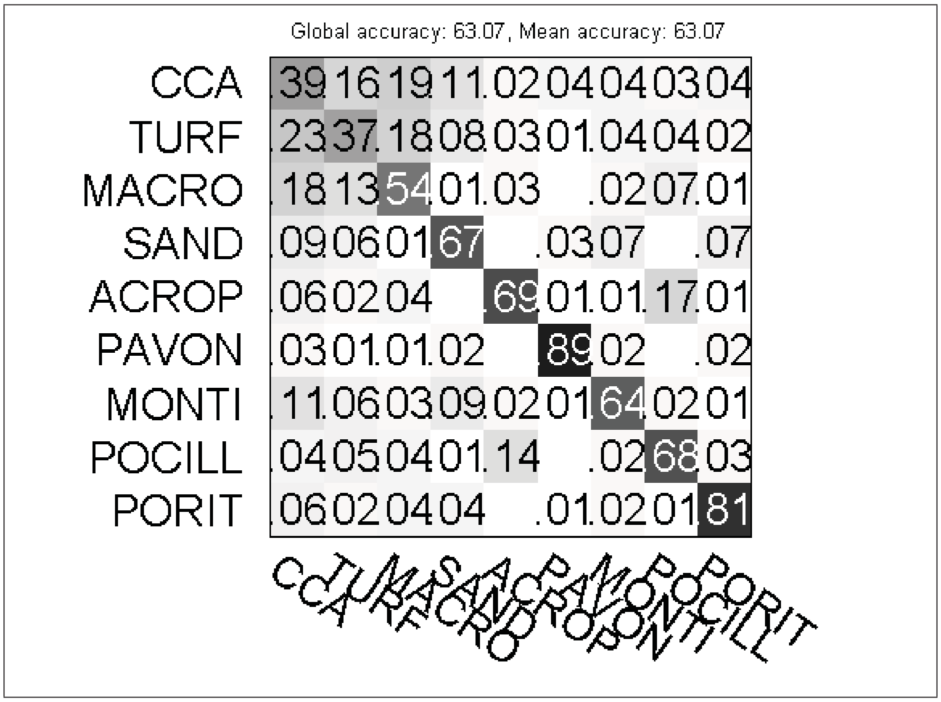
b. The Laplacian-of-Gaussian (LoG) Detector: LoG filters of various sizes are applied on the key point. The one with the maximal response corresponds to the scale of interest.

c. The Difference-of-Gaussian (DoG) Detector: a slight variant of the previous method that uses results from previously applied LoG filters to approximate and speed up the search of the best scale. This a very common approach used for SIFT.

0.3 Descriptor Extraction

As SIFT is a local descriptor, it considers each key point separately and generates a unique feature vector for each one. The SIFT features are magnitude weighted histograms of the gradient orientations for blocs within the region of interest.

The gradient in the regions of interests (ROIs) around the key point at the selected scale is sampled along a 16×16 grid, yielding 256 magnitudes and 256 orientations. The 256 samples are aggregated in groups of 16 ($4 \times 4 blocs$) in the form of 16 8-bin orientation histograms weighted with a combination of the magnitude component as well as a Gaussian function to give more importance to pixels closer to the key point. All 16 8-bin histograms are then concatenated to form a 128 values feature vector describing the key point. Each keypoint has its own feature vector.



Global accuracy: 63.07, Mean accuracy: 63.07

	CCA	TURF	MACRO	SAND	ACROP	PAVON	MONTI	POCILL	PORIT
CCA	.39	.16	.19	.11	.02	.04	.04	.03	.04
TURF	.23	.37	.18	.08	.03	.01	.04	.04	.02
MACRO	.18	.13	.54	.01	.03		.02	.07	.01
SAND	.09	.06	.01	.67		.03	.07		.07
ACROP	.06	.02	.04		.69	.01	.01	.17	.01
PAVON	.03	.01	.01	.02		.89	.02		.02
MONTI	.11	.06	.03	.09	.02	.01	.64	.02	.01
POCILL	.04	.05	.04	.01	.14		.02	.68	.03
PORIT	.06	.02	.04	.04		.01	.02	.01	.81

Figure-A II-2 Confusion matrix for regular fixed-size square patches on a sample of MLC 2008

0.4 Matching or Vectorizing

The resulting key point histograms describe local ROIs around the object, and can be used in many ways:

a. Object matching: the key points can be matched between two objects to generate a similarity metric, which can be thresholded to classify between match or non-match.

b. Vectorizing the key points: many methods have been developed to turn a set of key point features into a fixed size vector usable by common classifiers. Appendix III explores two of these methods.

APPENDIX III

A COMPARISON BETWEEN BAG-OF-WORDS (BOW), IMPROVED FISHER VECTOR (IFV), AND TEXTONS

BoW and IFV are pooling methods applicable to any local features. These build a dictionary of commonly observed patterns, and use it to turn a set of local feature vectors into a fixed length global feature vector that can be used by common pattern recognition techniques. These methods all require the same steps:

a. **Local feature extraction** outputs a set of feature vectors describing local areas across the image. This step is performed not only in operational mode when vectorizing local features, but also a priori, for defining a dictionary.

b. **Dictionary creation** uses unsupervised learning methods to find common patterns in the given problem. These common patterns form a set of primitives used for vectorization. The output vector is a histogram of the frequency at which every primitive pattern is observed in the input image.

c. **Quantization** takes a local feature vector and associates it with one (or more) primitives. This is necessary to create a histogram.

Table III-1 presents how these steps are commonly implemented for the BoW, IFV and even the texton methods, which is very similar as well.

Soft-quantization used along with GMM density estimation refers to the association of each local feature vector with all known bins, but with a variable weight based on the GMM probability.

The *improved* version of the Fisher Vector method uses signed square rooting and L2 normalization, which reportedly improves performance.

Table-A III-1 A comparison between Bag-of-Words, Improved Fisher Vector, and the texton method. Note that any local feature can be used for any method, but the following ones are most commonly used. The following acronyms are used: Nearest neighbor (NN), Gaussian Mixture Model (GMM), Scale Invariant Feature Transform (SIFT)

Processing step	**BoW**	**IFV**	**Textons**
Local feature	SIFT	SIFT	Gabor filter responses
Dictionary creation	K-Means	GMM density estimation	K-Means
Quantization method	Euclidean distance NN	Soft-quantization	Euclidean distance NN

APPENDIX IV

CLASSIFICATION RESULTS EXTENDED TO OTHER TAXONOMIC LEVELS

When tackling the coral annotation problem at a lower taxonomic rank, additional classes are integrated, many of which have extremely few samples. The low representation of these classes can be somewhat offset by using SVM training weights. However weights only move the decision boundary in favor of one class and at the cost of a higher error rate for others. The problem them becomes finding the appropriate weights given the error cost for each class in the problem domain. This larger problem is beyond the scope of this work.

As an alternative, we demonstrate the capabilities of our proposed system on a smaller sample built using 50 randomly selected samples per classes across the entire AIMS dataset. Classes with less than 50 samples are discarded. In total, 4 samples are generated, one at each lower taxonomic rank: Benthos, Family, Genus and Species. Figures IV-1 to IV-4 present the confusion matrices where the cells represent the number of samples corresponding to the real class (vertically) and the predicted class (horizontally). The maximum value is 50, as there are 50 samples per class. The mean classification rate (class-wise) is included in the title. Because there are 50 samples for all classes, this is also the global classification rate. Table IV-1 presents the classification rate obtained by each SVM individually, and shows that our MCF method improves performance consistently. Figure IV-5 to IV-11 illustrate the difficulty of the problem by showing 15 samples per class for each one for the four lower taxonomic rank datasets used here. Histogram equalization was applied for enhanced visualization. The three major challenges can easily be observed: high intra-class variance, low inter-class variance, ambiguous labeling.

An interesting observation was made during these experiments: Even if DTD_{RBF}^{IFV} performs well below 5% in some cases, removing it does not improve MCF performance. To our great surprise, it even slightly decreased classification rates in some cases (still within one standard deviation). This suggests that even if the features are not appropriate to solve the entire problem, they provide additional information that can help discriminate between two classes in some niche and difficult cases. It would be interesting to study those difficult cases individually and design features that are specifically useful in those cases. The cases in question are easily identified by looking at the confusion matrices.

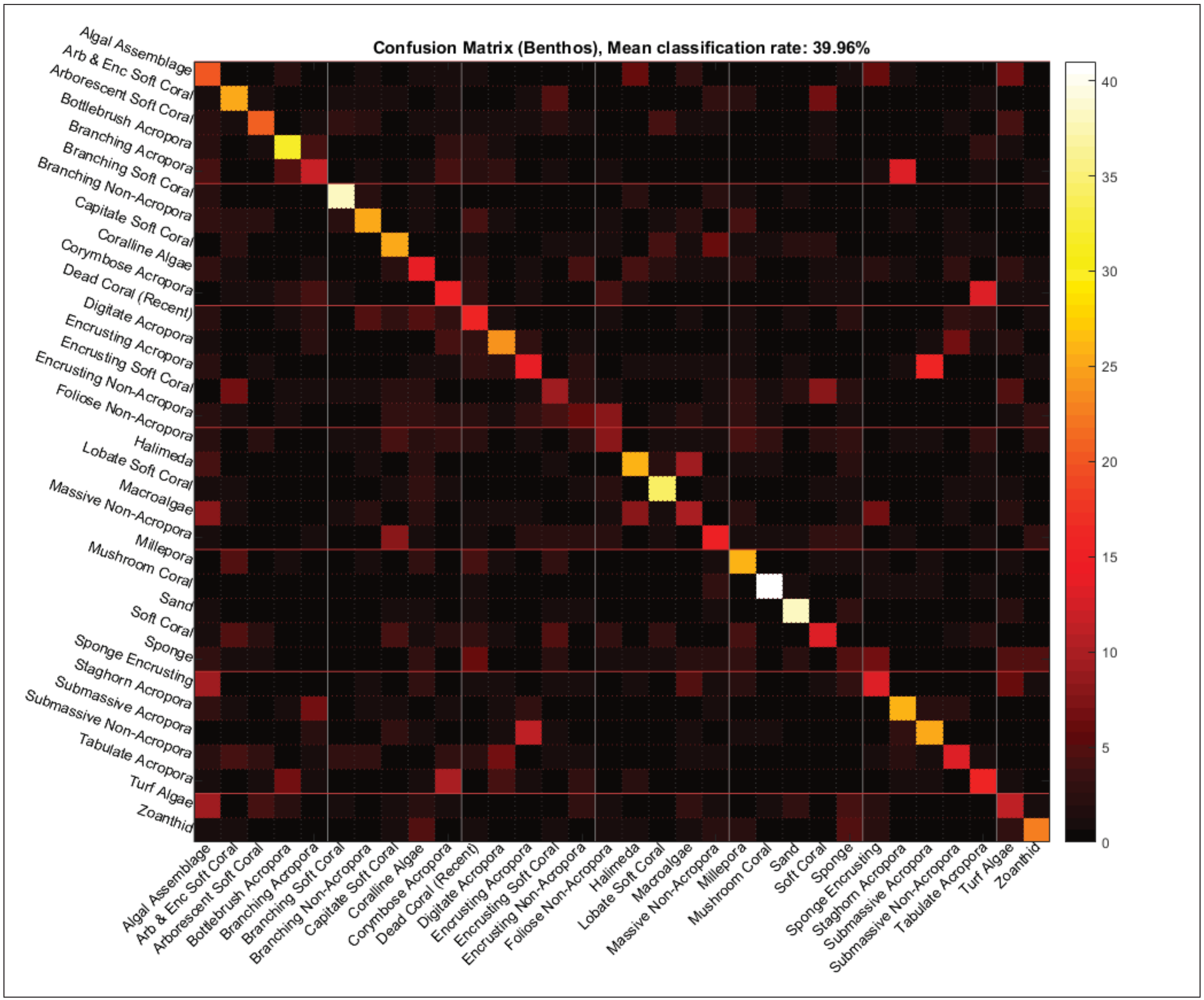


Figure-A IV-1 Confusion matrix at the benthos level, 50 samples per class are used.

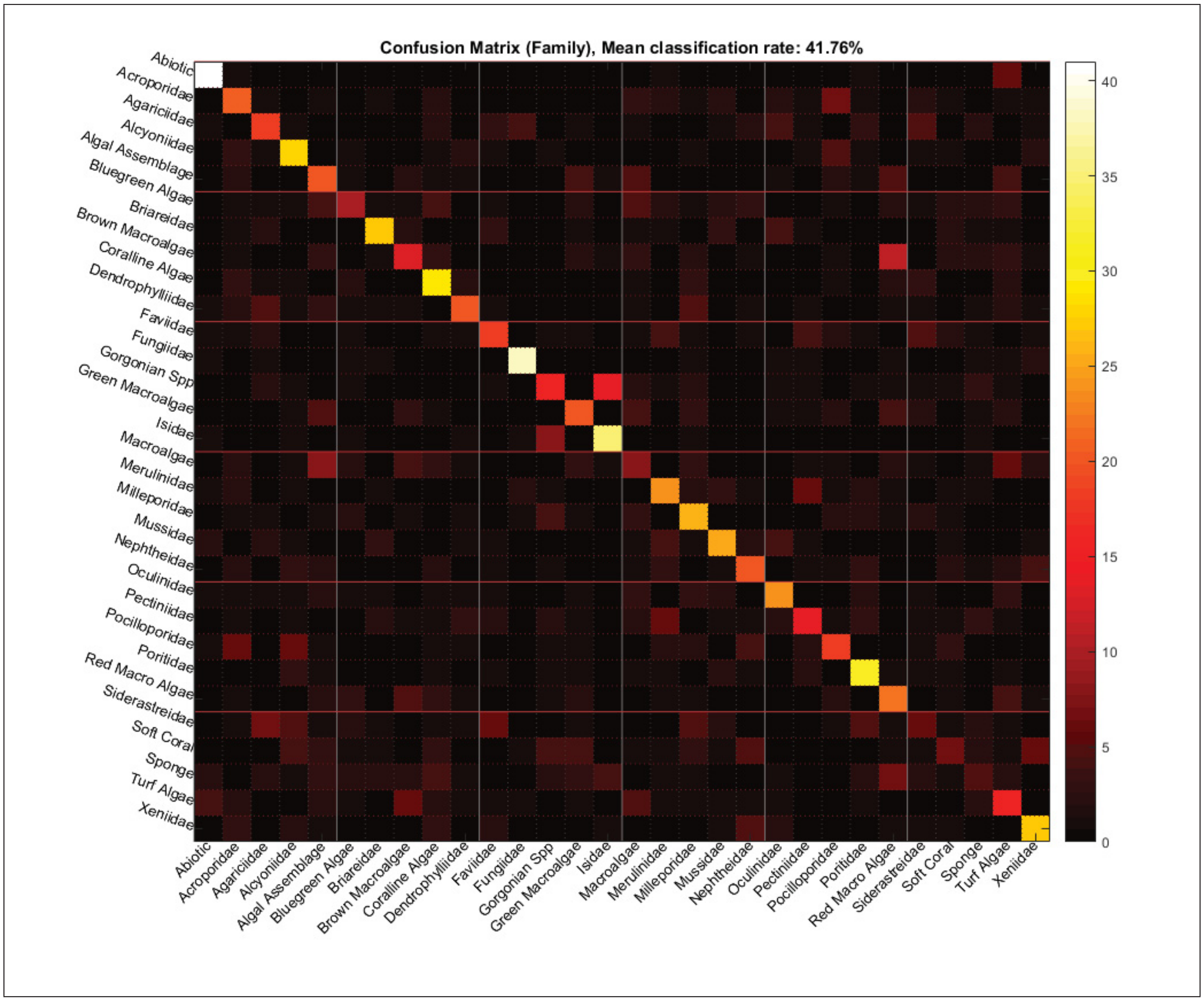


Figure-A IV-2 Confusion matrix at the family level, 50 samples per class are used.

Table-A IV-1 Mean classification rates obtained for each SVM individually as well as with our proposed MCF method (ten-folds averages)

	Family	Benthos	Genus	Species
Blanchet	22.4 ± 3.0 %	21.3 ± 3.6 %	17.4 ± 2.9 %	18.1 ± 2.1 %
Beijbom (Textons)	27.7 ± 3.6 %	27.1 ± 4.9 %	23.8 ± 2.7 %	24.0 ± 2.2 %
DTD_{RBF}^{IFV}	4.9 ± 0.6 %	4.1 ± 1.1 %	2.8 ± 0.2 %	1.9 ± 0.4 %
DeCAF	34.2 ± 3.2 %	34.5 ± 4.3 %	32.3 ± 3.2 %	31.0 ± 2.3 %
Proposed MCF	**41.8** ± **4.5** %	**40.0** ± **5.3** %	**38.1** ± **2.6** %	**38.4** ± **2.6** %

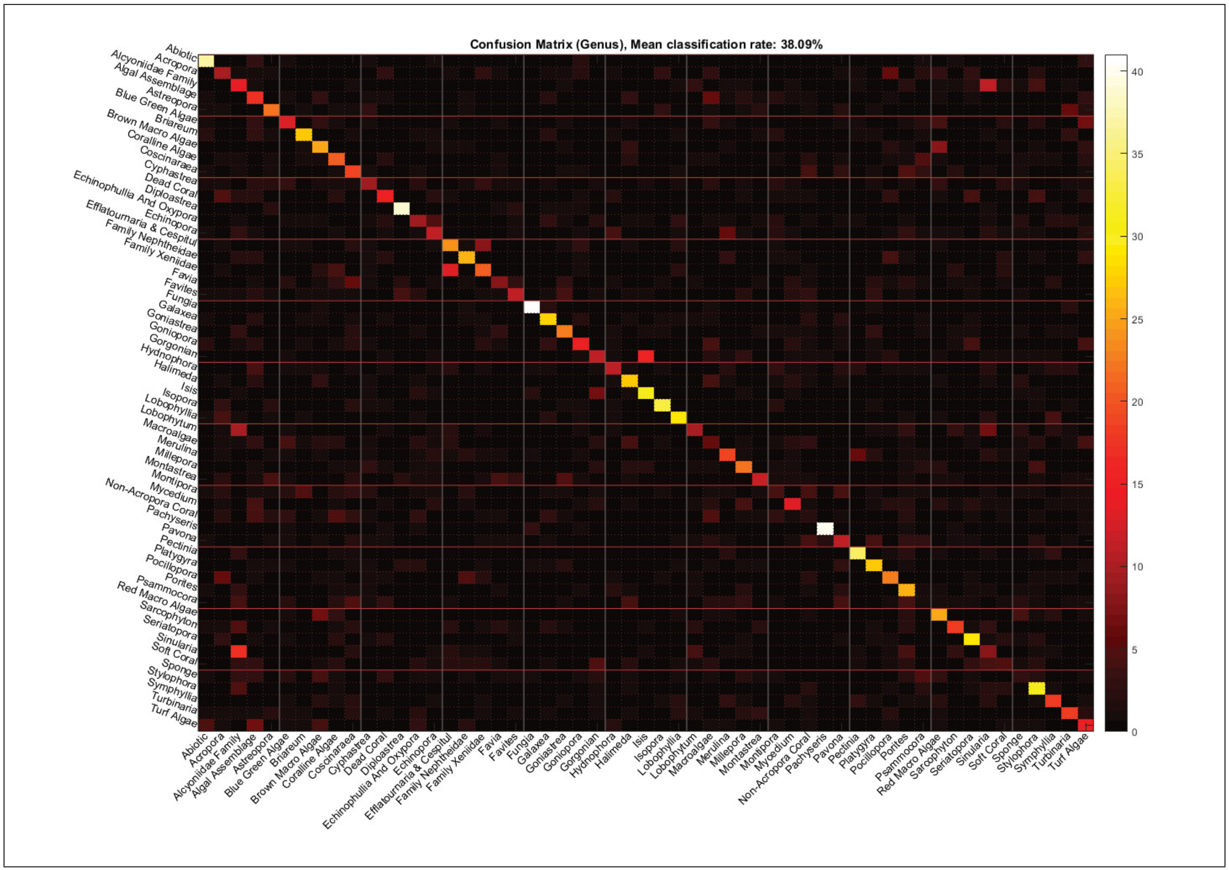


Figure-A IV-3 Confusion matrix at the genus level, 50 samples per class are used.

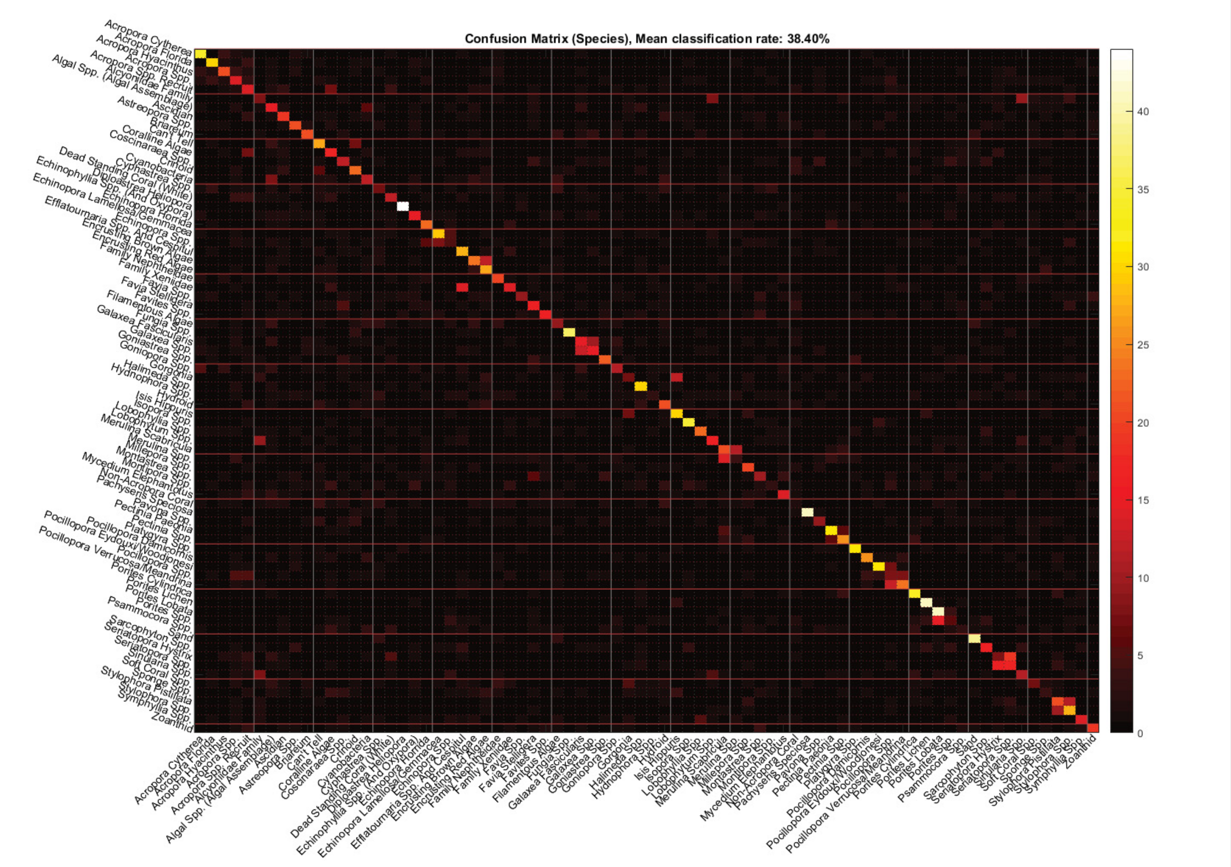


Figure-A IV-4 Confusion matrix at the species level, 50 samples per class are used.

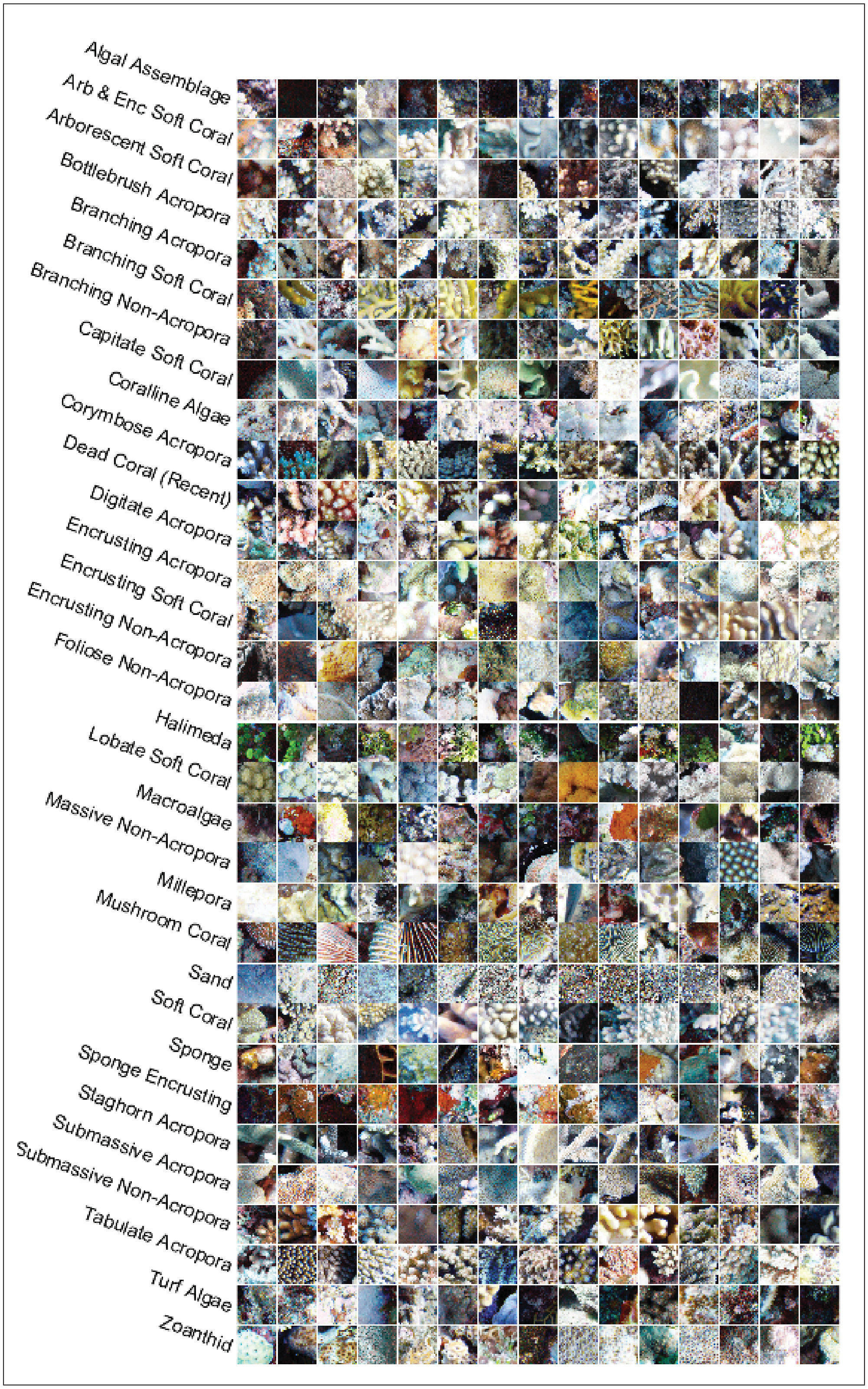


Figure-A IV-5   15 samples per class at the benthos level

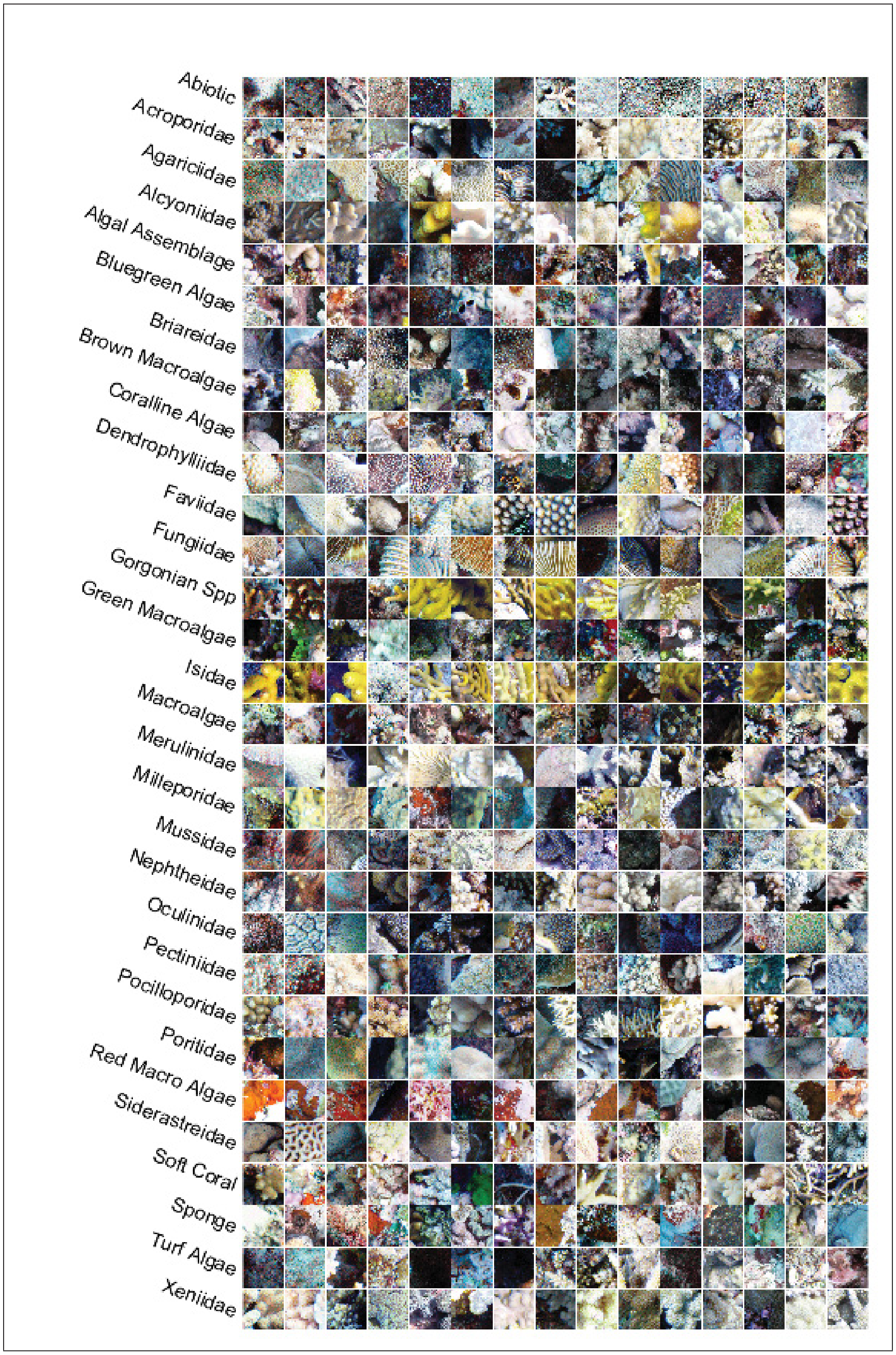


Figure-A IV-6 15 samples per class at the family level

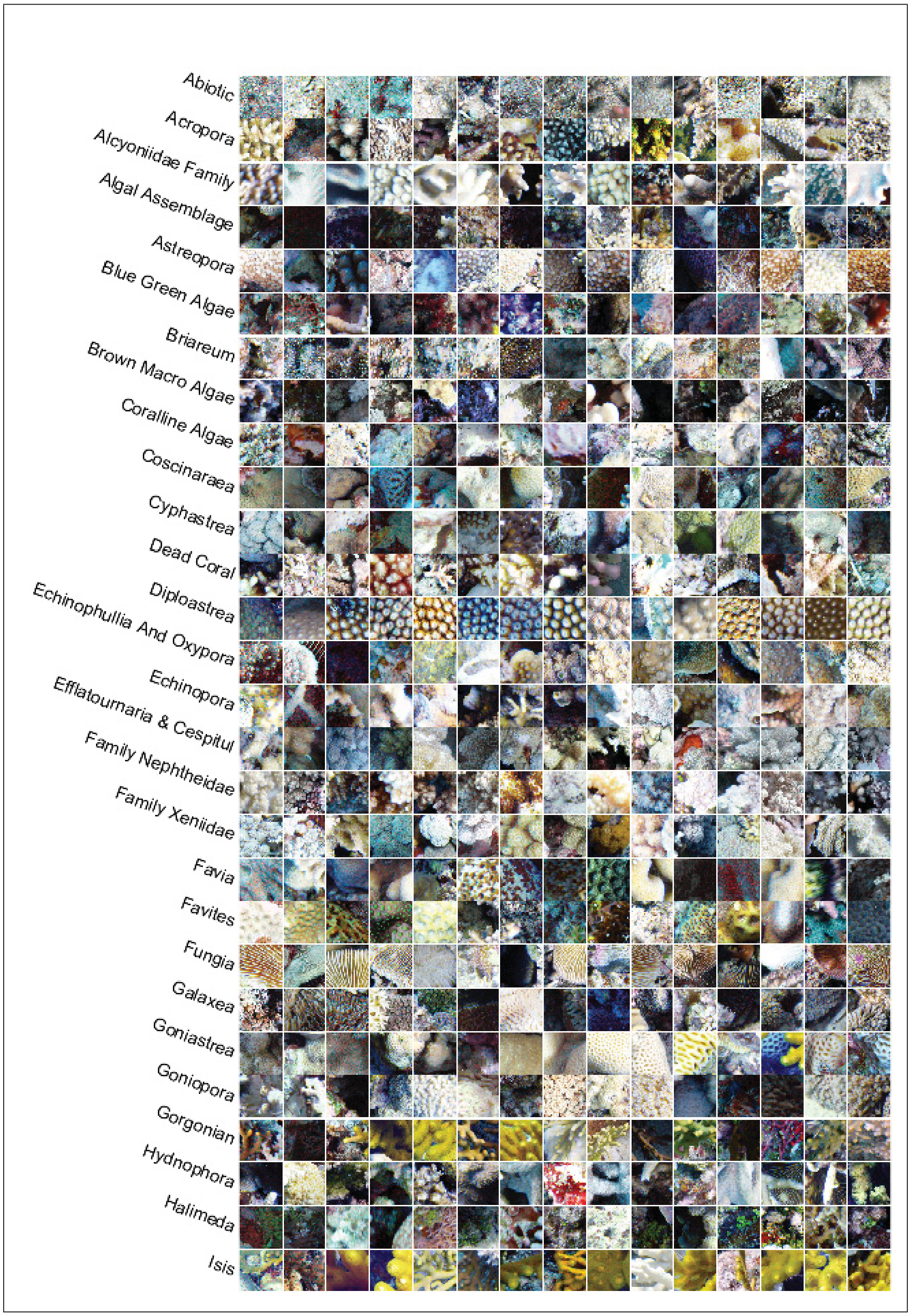


Figure-A IV-7 15 samples per class at the genus level (1 of 2)

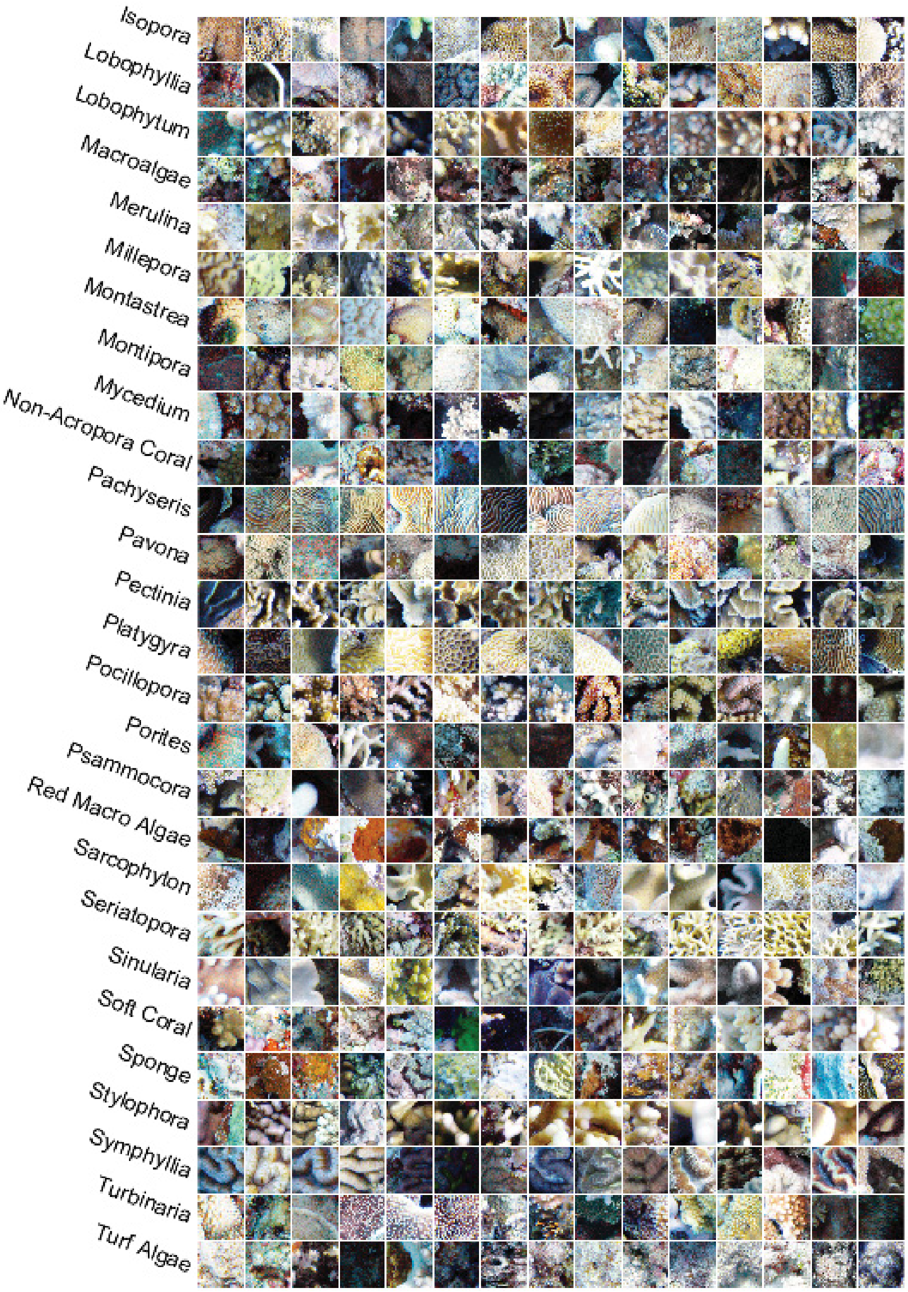


Figure-A IV-8 15 samples per class at the genus level (2 of 2)

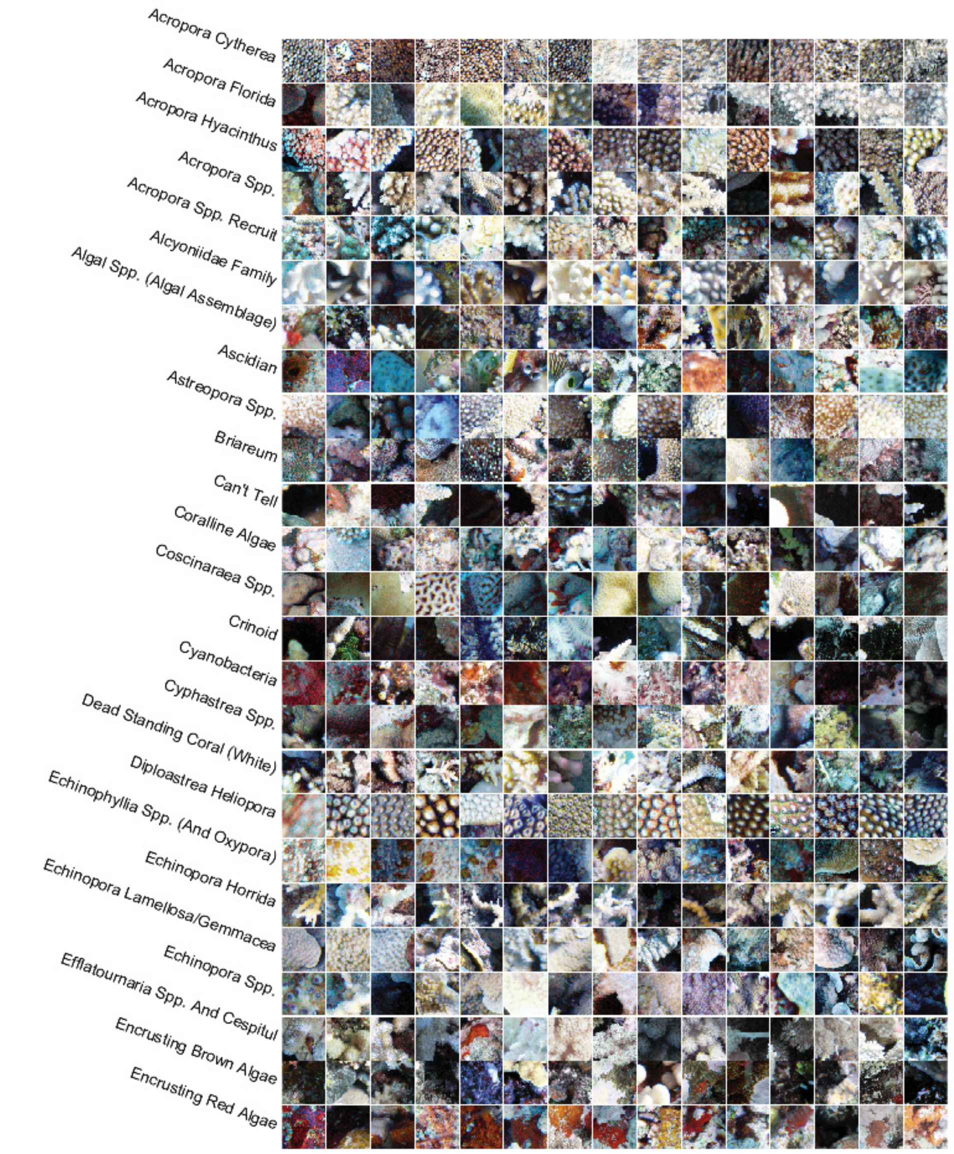


Figure-A IV-9 15 samples per class at the species level (1 of 3)

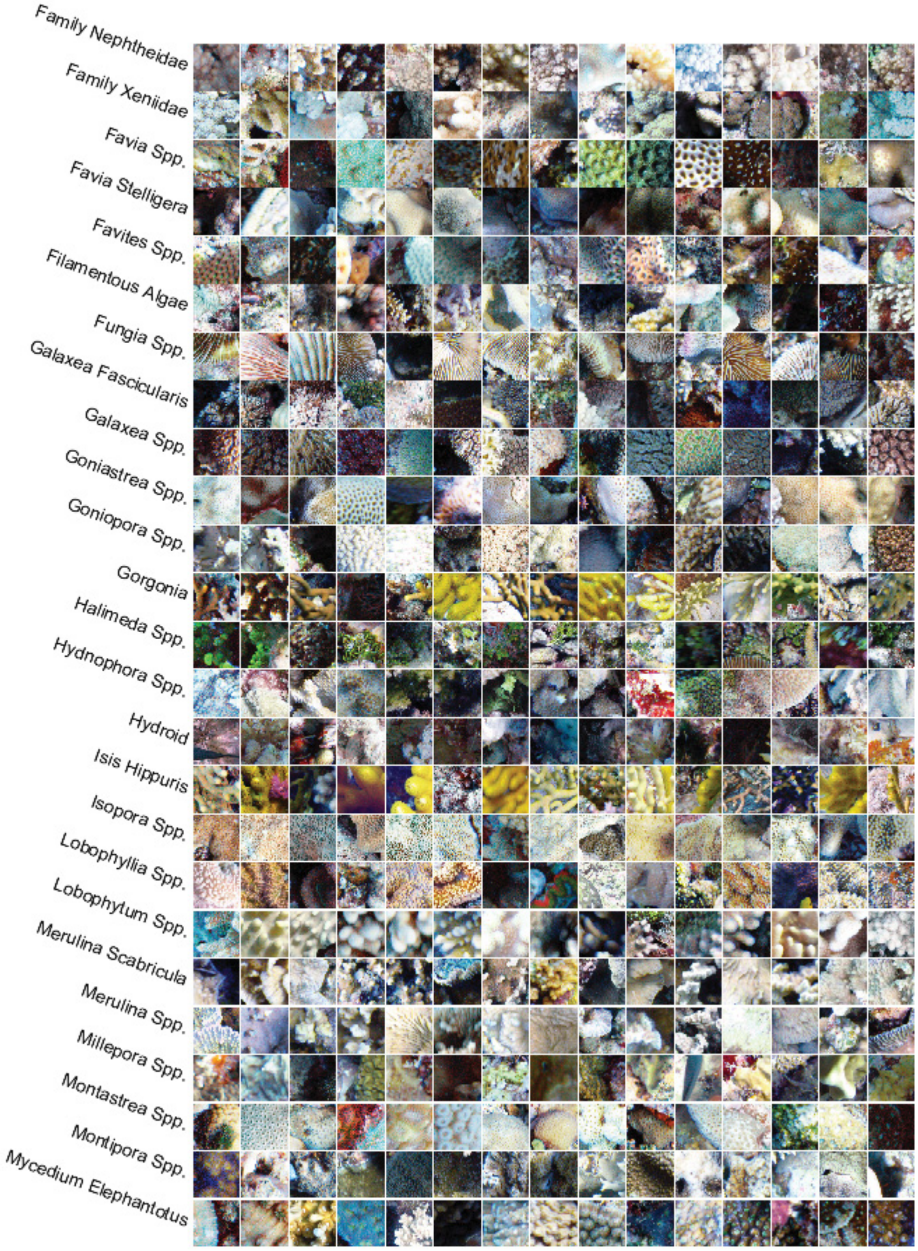


Figure-A IV-10 15 samples per class at the species level (2 of 3)

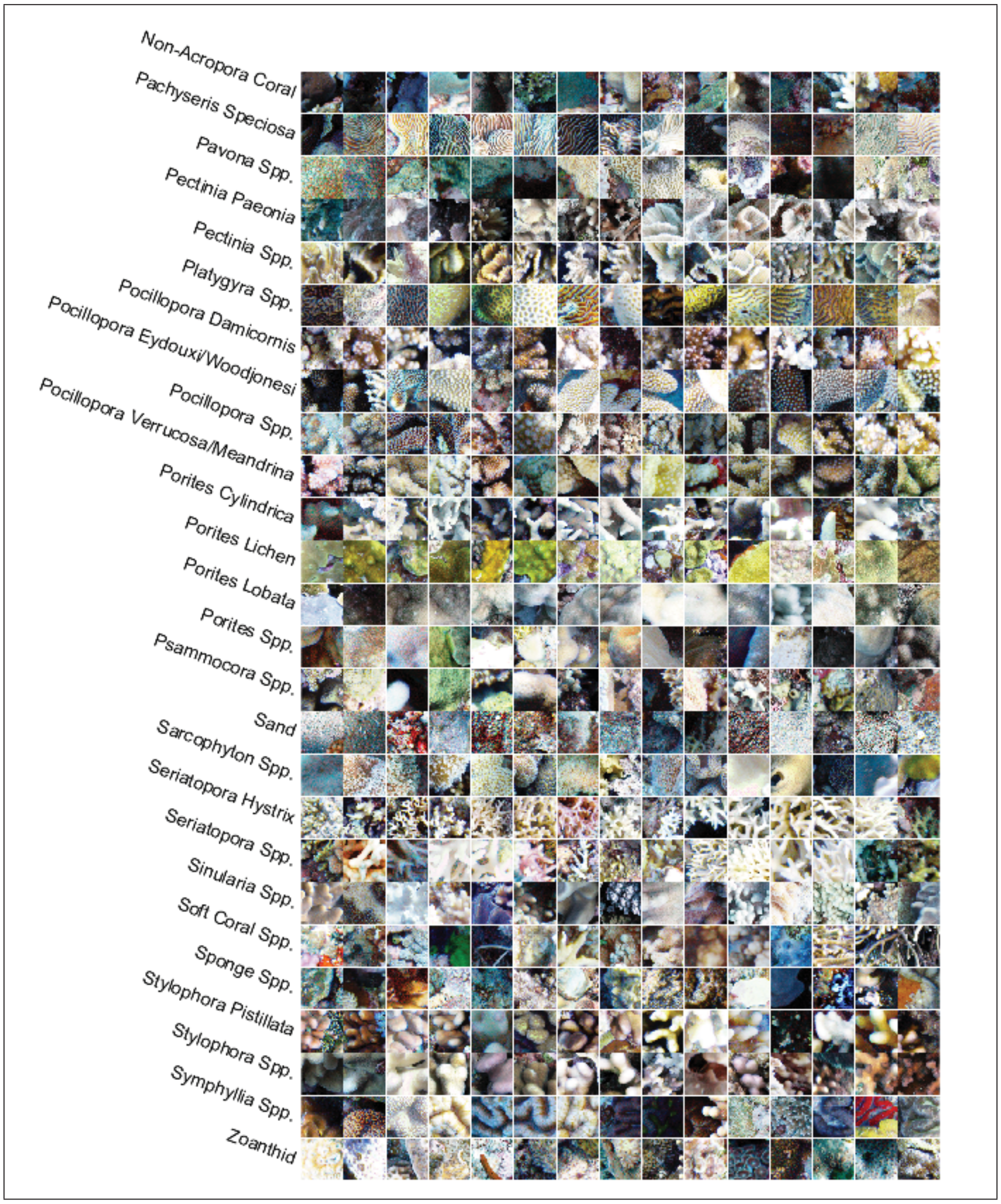


Figure-A IV-11 15 samples per class at the species level (3 of 3)

APPENDIX V

PRELIMINARY WORK ON SEGMENTATION

Throughout this work, we've experimented extensively with segmentation. Segmentation quality is difficult to measure for several reasons:

a. Only single points are available as a ground truth. Popular segmentation quality metrics cannot be applied.

b. Even if a few images were expertly segmented, it would be hard to measure how well the segmentation approach generalize to other images, as the range of observable textures and the variability of a single texture on an image can be quite significant. Furthermore, perfect segmentation of the coral is somewhat irrelevant. A reasonably sized sample should suffice, similarly to what is being done with fixed size patches. The objective is simply to eliminate ambiguity when multiple classes are observed around a point.

c. While a good metric could be the classification rate improvement, the single point ground truth does not allow good improvement measurement: where two classes meet near the sampled point, the expert will chose one, and the system may chose the other, or worse, may treat the transitional texture in between the two as the texture of interest. This leads to segmentations results unrepresentative of the expert's intentions, and therefore to disagreement between the ground truth and the system predictions.

The following images show samples of our unsupervised adaptive thresholding method designed to find similarly-textured pixels around the pixel of interest.

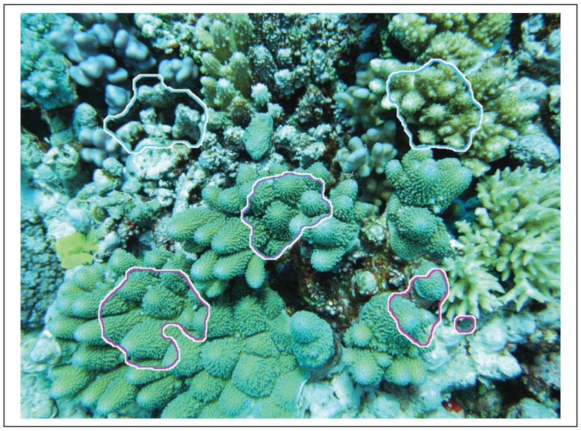


Figure-A V-1 Results from an experimental segmentation algorithm.

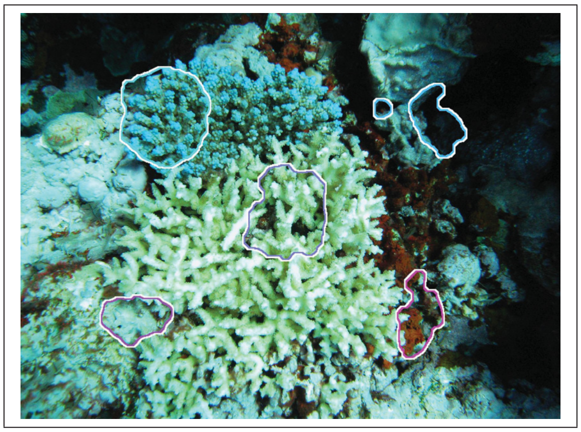


Figure-A V-2 Results from an experimental segmentation algorithm.

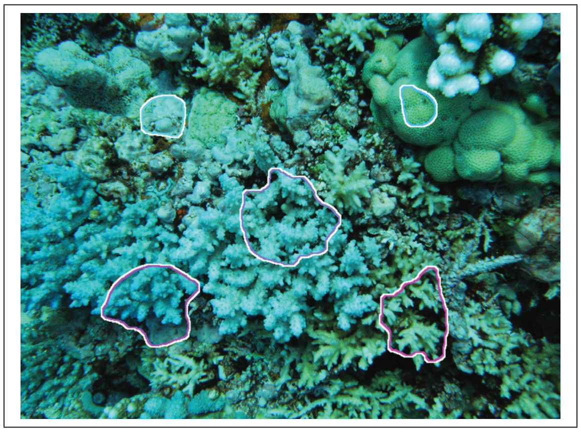


Figure-A V-3 Results from an experimental segmentation algorithm.

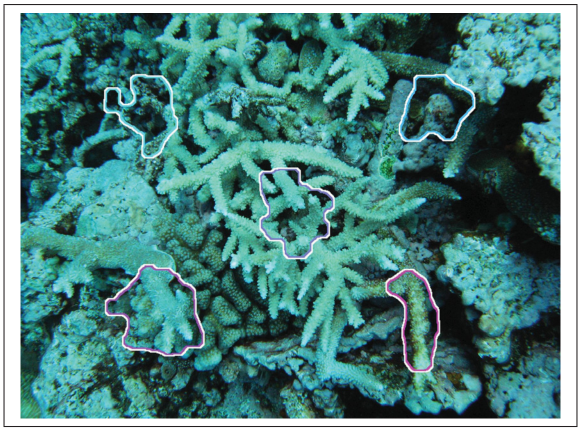


Figure-A V-4 Results from an experimental segmentation algorithm.

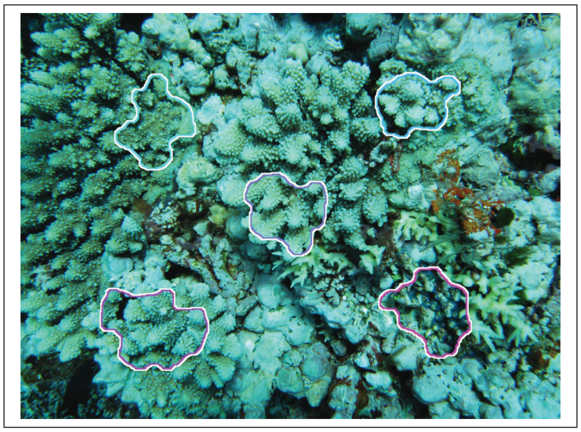


Figure-A V-5 Results from an experimental segmentation algorithm.

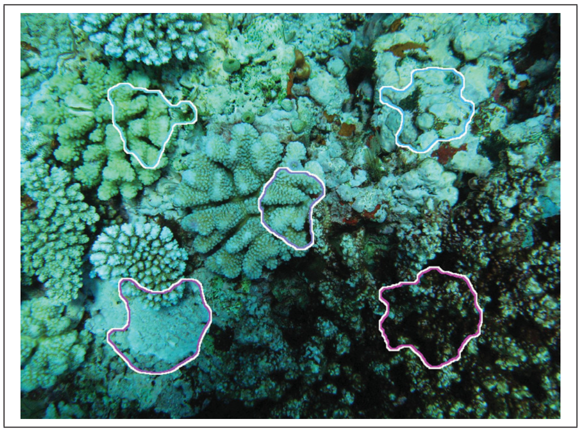


Figure-A V-6 Results from an experimental segmentation algorithm.

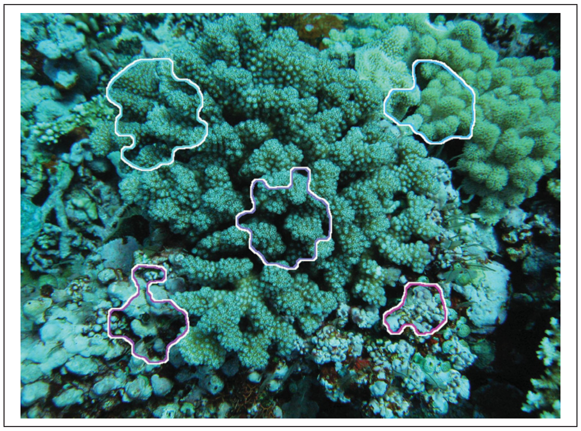


Figure-A V-7 Results from an experimental segmentation algorithm.

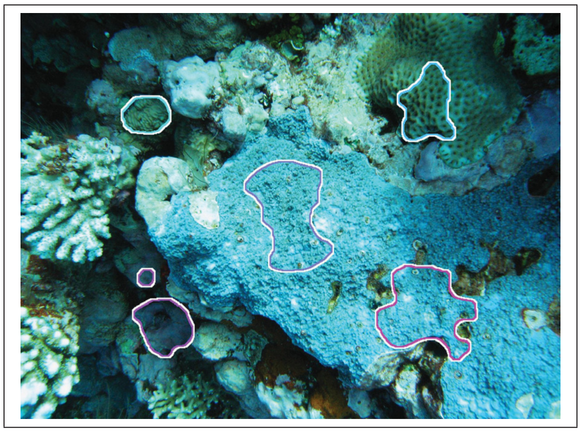


Figure-A V-8 Results from an experimental segmentation algorithm.

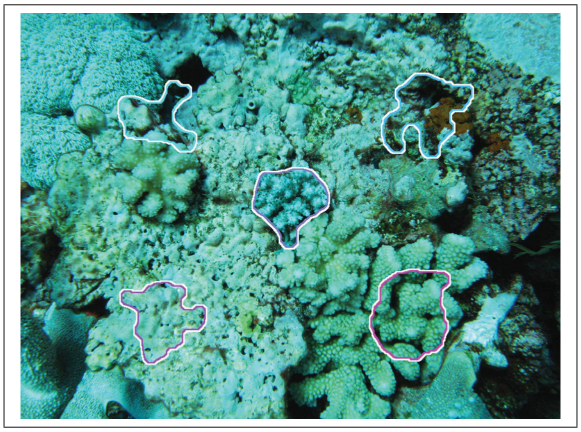


Figure-A V-9 Results from an experimental segmentation algorithm.

BIBLIOGRAPHY

Arnold-Bos, A., J.-P. Malkasse, and G. Kervern. 2005. "Towards a model-free denoising of underwater optical images". In *Oceans 2005-Europe*. p. 527–532. IEEE.

Bazeille, S., I. Quidu, L. Jaulin, and J.-P. Malkasse. 2006. "Automatic underwater image pre-processing". In *CMM'06*.

Beijbom, O., P. J. Edmunds, D. Kline, B. G. Mitchell, D. Kriegman, et al. 2012. "Automated annotation of coral reef survey images". In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. p. 1170–1177. IEEE.

Beijbom, O., P. J. Edmunds, C. Roelfsema, J. Smith, D. I. Kline, B. P. Neal, M. J. Dunlap, V. Moriarty, T.-Y. Fan, C.-J. Tan, et al. 2015. "Towards Automated Annotation of Benthic Survey Images: Variability of Human Experts and Operational Modes of Automation". *PloS one*, vol. 10, n° 7.

Bernd, J., 2002. *Digital image processing: electronic version of the book, exercises, additional images and runtime version of the heurisko image processing software*.

Blanchet, J.-N. "Sélection et extraction de caractéristiques pour l'annotation automatisée d'images des récifs coralliens".

Bouchard, J. 2011. "Méthodes de vision et d'intelligence artificielles pour la reconnaissance de spécimens coralliens". *École de Technologie Supérieure*.

Carlevaris-Bianco, N., A. Mohan, and R. M. Eustice. 2010. "Initial results in underwater single image dehazing". In *OCEANS 2010*. p. 1–8. IEEE.

Chang, C.-C. and C.-J. Lin. 2011. "LIBSVM: A library for support vector machines". *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, n° 3, p. 27.

Cheriet, M., N. Kharma, C.-L. Liu, and C. Suen, 2007. *Character recognition systems: a guide for students and practitioners*.

Chow, C. K. 1970. "On optimum recognition error and reject tradeoff". *Information Theory, IEEE Transactions on*, vol. 16, n° 1, p. 41–46.

Cimpoi, M., S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. 2014. "Describing textures in the wild". In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. p. 3606–3613. IEEE.

Cortes, C. and V. Vapnik. 1995. "Support-vector networks". *Machine learning*, vol. 20, n° 3, p. 273–297.

Culverhouse, P. F., R. Williams, B. Reguera, V. Herry, and S. González-Gil. 2003. "Do experts make mistakes? A comparison of human and machine identification of dinoflagellates". *Marine Ecology Progress Series*, vol. 247, n° 17-25, p. 5.

Dana, K., B. Van Ginneken, S. Nayar, and J. Koenderink. 1997. "Columbia-utrecht reflectance and texture database".

Donahue, J., Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. 2013. "Decaf: A deep convolutional activation feature for generic visual recognition". *arXiv preprint arXiv:1310.1531*.

Donate, A. 2006. "Texture Classification and Segmentation of Coral Reef Images".

Dubuisson, B. and M. Masson. 1993. "A statistical decision rule with incomplete knowledge about classes". *Pattern recognition*, vol. 26, n° 1, p. 155–165.

Duda, R. O., P. E. Hart, and D. G. Stork, 2012. *Pattern classification*.

Edmunds, P., V. Moriarty, and O. Beijbom. 2012. "MCR LTER: Coral Reef: Computer Vision: Multi-annotator Comparison of Coral Photo Quadrat Analysis". knb-lter-mcr.5013.3.

Finlayson, G. D., B. Schiele, and J. L. Crowley. 1998. Comprehensive colour image normalization. *Computer Vision—ECCV'98*, p. 475–490. Springer.

Fogel, I. and D. Sagi. 1989. "Gabor filters as texture discriminator". *Biological cybernetics*, vol. 61, n° 2, p. 103–113.

Gleason, A., R. Reid, and K. Voss. 2007. "Automated classification of underwater multispectral imagery for coral reef monitoring". In *OCEANS 2007*. p. 1–8. IEEE.

Gonzalez, R. C., R. E. Woods, and S. L. Eddins, 2004. *Digital image processing using MATLAB*.

Grauman, K. and B. Leibe, 2010. *Visual object recognition*. Number 11.

Guo, Z., L. Zhang, and D. Zhang. 2010. "A completed modeling of local binary pattern operator for texture classification". *Image Processing, IEEE Transactions on*, vol. 19, n° 6, p. 1657–1663.

Haralick, R. M., K. Shanmugam, and I. H. Dinstein. 1973. "Textural features for image classification". *Systems, Man and Cybernetics, IEEE Transactions on*, , p. 610–621.

Heikkilä, M., M. Pietikäinen, and C. Schmid. 2006. Description of interest regions with center-symmetric local binary patterns. *Computer Vision, Graphics and Image Processing*, p. 58–69. Springer.

Hoegh-Guldberg, O., P. Mumby, A. Hooten, R. Steneck, P. Greenfield, E. Gomez, C. Harvell, P. Sale, A. Edwards, K. Caldeira, et al. 2007. "Coral reefs under rapid climate change and ocean acidification". *science*, vol. 318, n° 5857, p. 1737–1742.

Johnson-Roberson, M., S. Kumar, O. Pizarro, and S. Willams. 2006. "Stereoscopic imaging for coral segmentation and classification". In *OCEANS 2006*. p. 1–6. IEEE.

Jonker, M., K. Johns, and K. Osborne. 2008a. "AIMS Long-term monitoring Program dataset".

Jonker, M., L. Johns, and K. Osborne. 2008b. *Surveys of benthic reef communities using underwater digital photography and counts of juvenile corals*.

Lazebnik, S., C. Schmid, and J. Ponce. 2005. "A sparse texture representation using local affine regions". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, n° 8, p. 1265–1278.

Lesser, M. and C. Mobley. 2007. "Bathymetry, water optical properties, and benthic classification of coral reefs using hyperspectral remote sensing imagery". *Coral Reefs*, vol. 26, n° 4, p. 819–829.

Liao, S., M. W. Law, and A. Chung. 2009. "Dominant local binary patterns for texture classification". *Image Processing, IEEE Transactions on*, vol. 18, n° 5, p. 1107–1118.

Lim, A., J. D. Hedley, E. LeDrew, P. J. Mumby, and C. Roelfsema. 2009. "The effects of ecologically determined spatial complexity on the classification accuracy of simulated coral reef images". *Remote Sensing of Environment*, vol. 113, n° 5, p. 965–978.

Liu, L., L. Zhao, Y. Long, G. Kuang, and P. Fieguth. 2012. "Extended local binary patterns for texture classification". *Image and Vision Computing*, vol. 30, n° 2, p. 86–99.

Lowe, D. G. 2004. "Distinctive image features from scale-invariant keypoints". *International journal of computer vision*, vol. 60, n° 2, p. 91–110.

Ninio, R., J. Delean, K. Osborne, and H. Sweatman. 2003. "Estimating cover of benthic organisms from underwater video images: variability associated with multiple observers". *Marine Ecology-Progress Series*, vol. 265, p. 107–116.

Ojala, T., M. Pietikäinen, and T. Mäenpää. 2002. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, n° 7, p. 971–987.

Pizarro, O., P. Rigby, M. Johnson-Roberson, S. B. Williams, and J. Colquhoun. 2008. "Towards image-based marine habitat classification". In *OCEANS 2008*. p. 1–7. IEEE.

Prabhakar, C. and P. Kumar. 2012. "An image based technique for enhancement of underwater images". *arXiv preprint arXiv:1212.0291*.

Prévost, I. 2015. "Application de la vision artificielle à l'identification de groupes benthiques dans une optique de suivi environnemental des récifs coralliens". *École de Technologie Supérieure*.

Roelfsema, C., S. Phinn, and W. Dennison. 2002. "Spatial distribution of benthic microalgae on coral reefs determined by remote sensing". *Coral Reefs*, vol. 21, n° 3, p. 264–274.

Sasano, M., M. Imasato, H. Yamano, and H. Oguma. 2013. "Optical Design & Engineering Monitoring the viability of coral reefs".

Schettini, R. and S. Corchs. 2010. "Underwater image processing: state of the art of restoration and image enhancement methods". *EURASIP Journal on Advances in Signal Processing*, vol. 2010, p. 14.

Shihavuddin, A., N. Gracias, R. Garcia, A. C. Gleason, and B. Gintert. 2013. "Image-based coral reef classification and thematic mapping". *Remote Sensing*, vol. 5, n° 4, p. 1809–1841.

Simonyan, K. and A. Zisserman. 2014. "Very deep convolutional networks for large-scale image recognition". *arXiv preprint arXiv:1409.1556*.

Szeliski, R., 2010. *Computer vision: algorithms and applications*.

Todorovic, S. and N. Ahuja. 2009. "Texel-based texture segmentation". In *Computer Vision, 2009 IEEE 12th International Conference on*. p. 841–848. IEEE.

Trefnỳ, J. and J. Matas. 2010. "Extended set of local binary patterns for rapid object detection". In *Proceedings of the Computer Vision Winter Workshop*.

Tusa, E., A. Reynolds, D. M. Lane, N. M. Robertson, H. Villegas, and A. Bosnjak. 2014. "Implementation of a fast coral detector using a supervised machine learning and Gabor Wavelet feature descriptors". In *Sensor Systems for a Changing Ocean (SSCO), 2014 IEEE*. p. 1–6. IEEE.

Uppuluri, A. 2008. "GLCM texture features". *Matlab Central, The Mathworks..(accessed 22.03. 11)*.

Van De Weijer, J. and C. Schmid. 2006. Coloring local feature extraction. *Computer Vision–ECCV 2006*, p. 334–348. Springer.

Varma, M. and A. Zisserman. 2005. "A statistical approach to texture classification from single images". *International Journal of Computer Vision*, vol. 62, n° 1-2, p. 61–81.

Wolpert, D. H. and W. G. Macready. 1997. "No free lunch theorems for optimization". *Evolutionary Computation, IEEE Transactions on*, vol. 1, n° 1, p. 67–82.

Zhang, L., R. Chu, S. Xiang, S. Liao, and S. Z. Li. 2007. Face detection based on multi-block lbp representation. *Advances in biometrics*, p. 11–18. Springer.

Zhang, W., S. Shan, W. Gao, X. Chen, and H. Zhang. 2005. "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition". In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. p. 786–791. IEEE.

Zhao, G. and M. Pietikainen. 2007. "Dynamic texture recognition using local binary patterns with an application to facial expressions". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, n° 6, p. 915–928.