THE UNIVERSITY *of* EDINBURGH

Balance-Guaranteed Optimized Tree with Reject option for live fish recognition

Phoenix X. Huang

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh

2014

Abstract

This thesis investigates the computer vision application of live fish recognition, which is needed in application scenarios where manual annotation is too expensive, when there are too many underwater videos. This system can assist ecological surveillance research, *e.g.* computing fish population statistics in the open sea. Some pre-processing procedures are employed to improve the recognition accuracy, and then 69 types of features are extracted. These features are a combination of colour, shape and texture properties in different parts of the fish such as tail/head/top/bottom, as well as the whole fish. Then, we present a novel Balance-Guaranteed Optimized Tree with Reject option (BGOTR) for live fish recognition. It improves the normal hierarchical method by arranging more accurate classifications at a higher level and keeping the hierarchical tree balanced. BGOTR is automatically constructed based on inter-class similarities. We apply a Gaussian Mixture Model (GMM) and Bayes rule as a reject option after the hierarchical classification to evaluate the *posterior* probability of being a certain species to filter less confident decisions. This novel classification-rejection method cleans up decisions and rejects unknown classes. After constructing the tree architecture, a novel trajectory voting method is used to eliminate accumulated errors during hierarchical classification and, therefore, achieves better performance. The proposed BGOTR-based hierarchical classification method is applied to recognize the 15 major species of 24150 manually labelled fish images and to detect new species in an unrestricted natural environment recorded by underwater cameras in south Taiwan sea. It achieves significant improvements compared to the state-of-the-art techniques. Furthermore, the sequence of feature selection and constructing a multi-class SVM is investigated. We propose that an Individual Feature Selection (IFS) procedure can be directly exploited to the binary One-versus-One SVMs before assembling the full multiclass SVM. The IFS method selects different subsets of features for each One-versus-One SVM inside the multiclass classifier so that each vote is optimized to discriminate the two specific classes. The proposed IFS method is tested on four different datasets comparing the performance and time cost. Experimental results demonstrate significant improvements compared to the normal Multiclass Feature Selection (MFS) method on all datasets.

Acknowledgements

First of all, I would like to thank my supervisor, Bob Fisher, for his support, patience and wise guidance both in a scientific and a personal meaning. I would like also to acknowledge Chris Williams, Victor Lavrenko and Bastiaan Boom for the guidance and encouragement through my thesis work.

I am also thankful to my thesis examiners, Amos Storkey and Mark S. Nixon, for their brilliant and challenging questions, technical comments and suggestions for improving this thesis.

Many thanks to all my colleagues in the Computer Vision Lab, Michael Xiao, Steven McDonagh, Lucia Ballerini, Lily Xiang Li, Çiğdem Beyan and Luis Horna Carranza, for their inspirations and feedbacks. I will miss the time that we were working together.

I would also like to thank the Fish4Knowledge project for providing the financial support for this thesis, and for all brilliant colleagues to their contributions to the development of technology and science.

My everlasting gratitude to my wife Rui Gu for her love and support. I would like to conclude by thanking my parents, who have patiently support me for this work to be finished and for all their support.

Above all, I would like to say thank you to my friends, Zhunchen Luo, He Wang, Jinli Hu, Leimin Tian, Xingxing Zhang, Wei Sun, Xiaofeng Zhao, Zhanxing Zhu, Xin He, Zhe Liu, Jianshen He, Zhengshuai Lin, Guoli Yang, Jie Wei, Yichuan James Zhang, Xi Zhao, Hsiu-Chin Lin, Feng Cheng, Gayathri Nadarajan, Peter Sandilands, Charles Di Leo, Vicente Morell Gimenez, who have been backing me all the way.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified. Part of the material in Chapters 3, 4, 5 and 6 has already been published or submitted in the following papers:

P. X. Huang, B. J. Boom, R. B. Fisher, "Underwater Live Fish Recognition using a Balance-Guaranteed Optimized Tree", ACCV 2012. 422-433.
http://homepages.inf.ed.ac.uk/rbf/PAPERS/accv2012finalpaper.pdf

P. X. Huang, B. J. Boom, R. B. Fisher, "Hierarchical Classification for Live Fish Recognition", BMVC student workshop, September 2012.
http://www.bmva.org/bmvc/2012/WS/paper1.pdf

P. X. Huang, B. J. Boom, R. B. Fisher, "GMM improves the reject option in hierarchical classification for fish recognition", WACV 2014, accepted.
http://homepages.inf.ed.ac.uk/s1064211/thesis/egpaper.pdf

P. X. Huang, B. J. Boom, R. B. Fisher, "Hierarchical classification with reject option for live fish recognition", submitted to Machine Vision and Application, 2014.
http://homepages.inf.ed.ac.uk/s1064211/thesis/fishRecognition.pdf

P. X. Huang, R. B. Fisher, "Individual feature selection in each One-versus-One classifier improves multi-class SVM performance", submitted to PRL, 2014.
http://homepages.inf.ed.ac.uk/s1064211/thesis/icpr14.pdf

B. J. Boom, P. X. Huang, J. He, R. B. Fisher, "Supporting Ground-Truth annotation of image datasets using clustering", 21st Int. Conf. on Pattern Recognition (ICPR), 2012.
http://homepages.inf.ed.ac.uk/rbf/PAPERS/PID2432553.pdf

Dr. Boom coordinated the collecting and labelling of the ground-truth data. My supervisor Prof. Fisher and I had plenty of discussions.

(*Phoenix X. Huang*)

Table of Contents

Chapter 1

Introduction

1.1 Why we want to recognize live fish?

Live fish recognition in the open sea has been investigated to promote commercial and environmental applications like fish farming, meteorologic monitoring and fish quota monitoring. It helps understanding of the marine ecosystem which is vital for studying issues that affect the marine environment, such as factitious pollution and climate change. Computer vision and pattern recognition techniques can help biologists observe marine ecosystems where manual annotation is too expensive, when there are too many underwater videos (from a tera-scale video database). In such environments, fish are swimming with general 3D freedom and a complex background including coral, sand and the open sea. Computer vision techniques can also help detect significant events and filter out most worthless content from mass video databases. An application system, when integrated with marine knowledge, can analyse underwater objects and compose high level interpretations, like fish counting, fish species distribution variation, and fish behaviour patterns. Marine scientists can benefit from the computer-assisted analysis of underwater videos, *e.g.* fish detection and species recognition for long-term observation [Walther et al., 2004], without needing specialist programming skills. Statistics about specific oceanic fish species distributions or aggregate counts of aquatic animals can assist biologists with resolving issues ranging from food availability to predator-prey relationships [Rova et al., 2007, Zion et al., 2000, Heithaus and Dill, 2002].

1.1.1 Introduction to underwater surveillance approaches

Traditionally, marine biologists have employed many tools to examine the appearance and quantities of fish. For example, they cast nets to catch and recognize fish in the ocean. They also dive to observe underwater, using photography [Caley et al., 1996]. Moreover, they combine net casting with acoustic (sonar) [Brehmer et al., 2006]. Nowadays, much more convenient tools are employed, such as hand-held video filming devices. There are two main disadvantages using this equipment. Firstly, these activities disturb fish swimming and habits, and thus giving rise to abnormal situations. This drawback is apparent: the fish are sensitive to their surrounding environment. Secondly, small amounts of acquisition data can not meet the demands for extensive underwater animal analysis, and the recorded data may omit valuable information. To resolve these issues, some researchers have implemented automatic analysis by using a Digital Signal Processor (DSP) chipset with camera onboard [Dunbabin et al., 2006]. It is cost-effective and easy to program (using standard C code). The DSP applies image processing algorithms and records data to its flash memory. A more popular and practical equipment is Remotely Operated Vehicle (ROV) [Blidberg, 2001, Gomes et al., 2003] with standard PC computation [Salam et al., 2004, Torres-Mendez and Dudek, 2005]. This equipment obtains video from mid water and produces high-resolution images of different fish species. The use of an ROV has achieved great success in collecting such data. They generate huge amounts of video containing animals from underwater cameras [Spampinato et al., 2008]. However, these techniques have their own shortcomings. A DSP chip with an embedded program cannot perform a rapid calculation, and an ROV can only stay underwater for a limited time.

In the Fish4Knowledge project, embedded video cameras in Figure 1.1 are used to record underwater animals (including insects, fish, *etc.*) at the Third Taiwanese Power Station as well as three other locations, and observe fish presence and habits at different times [Nadarajan et al., 2009]. The Fish4Knowledge project investigated methods for capture, storage, analysis and query of multiple video. The project goal is to analyse large amounts of data using a combination of computer vision, semantic web, database storage and query and work flow methods. Figure 1.2 is a surveillance system that is deployed at the HouBiHu station.

Figure 1.1: Embedded camera in Fish4Knowledge.

1.1.2 Automatic underwater fish recognition

Nowadays, underwater videos are mostly analysed by biologists [Spampinato et al., 2010], but it can be a tedious procedure. The difficulties are mainly two-fold.

- The huge amount of data
  As a camera produces $2 * 10^{12}$ bytes data ($5 * 10^4$ video clips) in a year, it may take 15 years for a marine biologist to analyse, recognize and label fish in these videos. In the whole project, 11 cameras have been recording for the last six years, which entails about 900 years' manpower to process this huge database. It is sensible to employ some automatic image processing methodologies to help marine biologists analyse them as the task of video processing is monotonous and complex.

- Complex foreground & background objects and low quality of video
  Live fish recognition in open water is fundamentally challenging because fish can move freely and illumination levels change frequently in such environments

Figure 1.2: Surveillance System in Fish4Knowledge project.

[Strachan, 1993a, Toh et al., 2009, Schettini and Corchs, 2010]. Furthermore, many fish images are blurred, and have fish at different distances and orientations or are against coral or ocean floor backgrounds. The Fish4Knowledge project presents a novel system to process massive sets of observations. It provides video analysis that automatically extracts information (*e.g.* fish detection, tracking and species recognition) about the observed marine animals.

As discussed above, the Fish4Knowledge project uses computer vision based automatic methodologies for underwater fish processing. Nadarajan *et al.* in [Nadarajan et al., 2009] proposed an integrated workflow system that aims at helping marine biologists annotate fish in underwater videos. Figure 1.3 shows a typical detection result.

1.2 The primary contributions

The primary contributions of this research project are:

Recognizing fish from underwater environment is challenging due to the difficult condi-

Figure 1.3: Fish detection result with bounding box on detected fish [Boom and Fisher, 2011]

tions: water blur, freely swimming fish, distance colour degradation, variable lighting and caustics. Previous algorithms in the literature were designed for dead fish, seen orthographically and using controlled lighting. Using features extracted from the underwater video stream that contains essentially 2D fish shapes moving freely in 3D, we developed an automatically generated hierarchical classification system with reject option. Based on these developments, we have developed a fish recognition system capable of recognising more species with high accuracy than previously, and tested on a larger database than previously.

The accuracy is based on the proportion of correct recognitions while robustness means recognizing fish in a complex environment (e.g. light distortion, fish occlusions and illumination transformations). To verify this claim, the project expects to recognize fish from different distances and angles from the camera, and to distinguish species from a large video set using the temporal information from tracked trajectories as well as using an individual's appearance similarity. A reject option, which assesses the *posterior* probability of whether the classification result belonged to the predicted species, is also implemented and evaluated.

To support this claim, an investigation into state-of-the-art fish recognition technologies is involved, which is evaluated by comparing accuracy and robustness with other models. Our procedure uses a combination of selected features such as the collective appearance, boundary geometry, specific shape geometry (*e.g.* fins, heads and tails),

colour and texture distributions as well as features within a special species. The automatically generated hierarchical classification also provides a reject option to filter less confident decisions of known classes or to detect and remove fish from untrained classes. A multiple-frame voting method is applied to improve the accuracy of the classification result.

1.3 Proposed solution and considerations

Research on fish recognition involves machine learning and computer vision. We are interested in correlations and differentiations between fish species, which are the bridge linking computer vision and marine biology. We analyse the formalization of fish species division to help design and implement the classifier. To ensure that the fish recognition algorithm was developed sufficiently robustly and precisely, our methodology is a combination of the items below.

- Computer vision and fish ichthyology characteristics
  We apply computer vision techniques for live fish recognition. This is a challenging task due to the low quality of the underwater video stream, which affects the accuracy of fish recognition by adding distortions and noise to the original image. The motion and diffraction effects blur the fish appearance like applying a convolution upon the original image. Furthermore, illumination levels change frequently both locally from caustics arising from the ocean surface waves and globally due to the sun and cloud positions. These factors decrease the video quality and produce classification errors. We introduced several types of descriptors that are effective and invariant to environmental changes. They are designed to integrate domain knowledge with machine vision methods. For example, some species of fish have specific colours, fin shapes, stripes or texture. Computer vision techniques exploit these colour/shape/texture similarities and present similar samples in the same cluster of feature density distribution.

- Hierarchical classification with automatically generated tree
  Live fish recognition is an application of multi-class classification. Marine biologists recognize fish based on taxonomic technology that uses special features like vivid colours, specific spots, *etc.* Abundant valuable knowledge used in constructing the hierarchical biological system can be adopted into the construction

of a fish recognition decision tree, which combines machine learning methods and inter-class visual similarity among fish species. Unlike a flat classifier that uses a feature set based on the average accuracy over all classes, hierarchical analysis pays more attention to grouping similar fish in the beginning and leave them for further processing, where these species can be better separated by specifically selected features. This strategy also helps reduce the imbalance of data. We present our hierarchical classification method called Balance-Guaranteed Optimized Tree (BGOT) for live fish recognition. It improves the normal hierarchical method by arranging more accurate classifications at a higher level and keeping the hierarchical tree balanced.

- Temporal Information
  The low quality of the underwater video frame greatly limits the accuracy of the fish recognition procedure, especially the visual distortion of fish phyletic description. As each fish appears in multiple frames from a video shot, we exploit the trajectory analysis to integrate the performance among these frames. Furthermore, fish may change direction and posture while swimming, which also impacts on the representation of features. Figure 1.4 shows a four-frame sequence of the same fish. It may be difficult to recognize the fish from just one single frame due to low quality. We combine the results from several frames to improve the recognition performance.

Figure 1.4: Multiple views of the same fish based on the tracking result [Boom and Fisher, 2011]

- Reject option for eliminating less confident decisions
  A hierarchical classification method has the problem of error accumulation. Each level of the hierarchical tree has some classification errors. In fish recognition, especially when our database is extremely imbalanced, misclassified samples are passed into deeper layers and reduce the average accuracy of the final recognition performance. Furthermore, the normal multi-class classifier identifies every test sample into one of the training classes. Although our fish recognition ground-

truth dataset covers the most dominant species of fish, there are still many observed fish from unmodeled species. These fish images are classified incorrectly as known species, and the precision is thus decreased. A "reject option" for a multi-class classifier was developed, which allows the classifier to reject less confident classification results, labelling them as recognition errors or unknown classes. The approach assumes that the properties of the expected class can be clustered into a few self-consistent clusters. Misclassifications along the paths in the hierarchy will lead to samples with low likelihood scores. We apply a Gaussian Mixture Model (GMM) at the leaves of the hierarchical tree. It evaluates the *posterior* probability of the testing samples and reduces the false positive rate since some misclassification errors in the BGOT classifier can be overcome at the price of a slightly lower true positive rate due to incorrect rejections.

- Individual feature selection for OvO classifier
  Multiclass One-versus-One (OvO) SVM, which is constructed by assembling a group of binary classifiers, is usually treated as a black-box. The usual Multiclass Feature Selection (MFS) algorithm chooses an identical subset of features for every OvO SVM. We propose that Individual Feature Selection (IFS) can be directly applied to each binary OvO SVM. More specifically, the proposed method selects different subsets of features for each OvO SVM inside the multiclass classifier so that each vote is optimized to discriminate between the two specific classes.

The Balance-Guaranteed Optimized Tree with Reject option (BGOTR) presented in this thesis is believed to be the first application of the hierarchical classification method with reject option for free swimming fish in an unconstrained environment. It is a novel hierarchical classification method suited for greatly unbalanced classes, and a novel classification-rejection method to clear up decisions and reject unknown classes. This system assists ecological surveillance research, *e.g.* fish population statistics in the open sea.

1.4 Organization of the thesis

This thesis is concerned with marine knowledge-based classification from underwater video streams and filtering less confident decisions. It is integrated in an automatically

generated hierarchical framework BGOTR, to provide an effective live fish recognition in an unrestricted environment. The first chapter illustrates the introduction and motivation of this thesis. It focuses on stating the background knowledge for underwater video analysis and its benefits to marine biologists. The second chapter summarizes fish recognition approaches in the literature. We review these research works and outline their advantages and issues. Chapter 3 is the first technical section, and it presents our feature extraction work. We employ several types of feature extraction methods to compute effective descriptors for fish. Some idiosyncratic fish features are also designed to integrate computer vision techniques with marine knowledge. The hierarchical classification method is discussed in chapter 4. We propose a set of heuristics which are helpful to construct the BGOT hierarchical tree. The proposed method is evaluated on a live fish dataset. In order to filter false detections in the fish detection results and eliminate false positives after the hierarchical classification, we propose a GMM-based reject option and evaluate its performance in real videos. This result refinement research is discussed in chapter 5. In chapter 6, we propose that an individual feature selection procedure can be directly used for each binary one-versus-one SVM before assembling the full multiclass SVM. The last chapter summarizes the whole system presented in this thesis.

Chapter 2

State of the art of fish species recognition

2.1 Introduction

This chapter presents a comprehensive literature review for research on fish recognition in both the marine biology and machine vision areas. This introduction is followed by a further analysis of the limitations of traditional onsite analysis due to difficulties in acquiring samples, and then we discuss the recent computer vision applications to fish species recognition. At the end, this chapter gives a summary of the state-of-the-art of fish recognition approaches and discusses previous systems to recognize free swimming fish in complex background environment. Chapter 3 will review commonly used features for fish species recognition.

Fish recognition is a challenging and worthy task considering that it is widely demanded for commercial and agricultural purposes [Heithaus and Dill, 2002]. In this chapter, we surveyed the pertinent literature on the study of fish recognition and summarize these approaches to demonstrate the evolution of fish recognition methods. In section 2.2, we briefly review some traditional research in the field. In the past decades, especially as machine learning methods were introduced to computer vision applications, computer-assisted recognition systems became popular since they are efficient and effective. Section 2.3 presents some recent computer vision applications used for object recognition and discusses their applicability to our problem. We give an introduction to the Fish4Knowledge project in Section 2.4. We discuss some important issues related

to literature on machine vision for fish recognition in the Section 2.5. We investigate novel techniques to perform effective live fish recognition in an unrestricted natural environment where the prior research is mainly restricted to constrained environments.

2.2 Traditional fish recognition methods

Traditionally, marine biologists identify fish from their ichthyological characteristics such as meristics and morphometrics, scale morphology, parasites, cytogenetics, protein electrophoresis (isoelectric focusing), immunogenetics *etc.* ([Begg and Waldman, 1999]). The ichthyology ontology is an academic question which aims to construct a scientific methodology to systematize animals into their hierarchical categories. A fish species taxonomy tree is shown in Figure 2.1. Taxon, as the leaf node of the whole tree, is

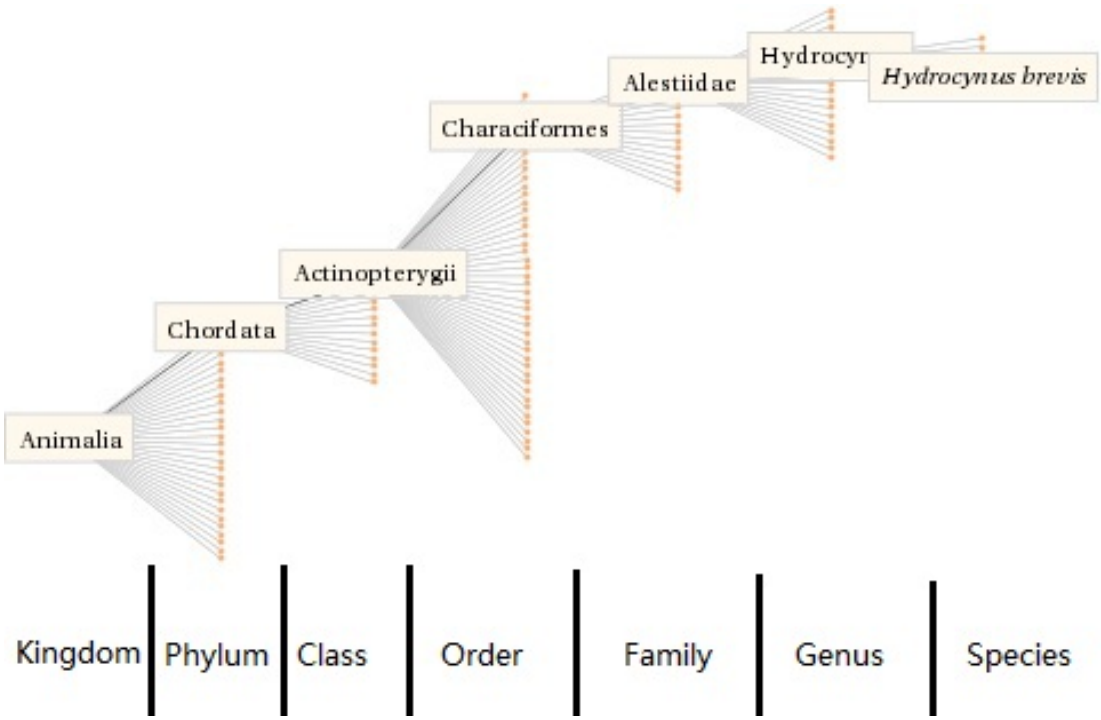


Figure 2.1: Fish Taxonomy Tree (from Tree of Life website)

the basement of taxonomy knowledge. For each taxon in the taxonomic tree, there is a top-to-bottom description to identify its hierarchical information. Taxonomy information is based on the synapomorphies characteristic from the extent to which the taxon is monophyletic, and it makes the explicit distinction between species, *e.g.* the presence or absence of components, specific numbers, particular shape, *etc.* Figure

2.2 shows examples of the tail and dorsal fin shapes and construction, which are utilized to identify fish from the ichthyological categories. In order to capture this in-

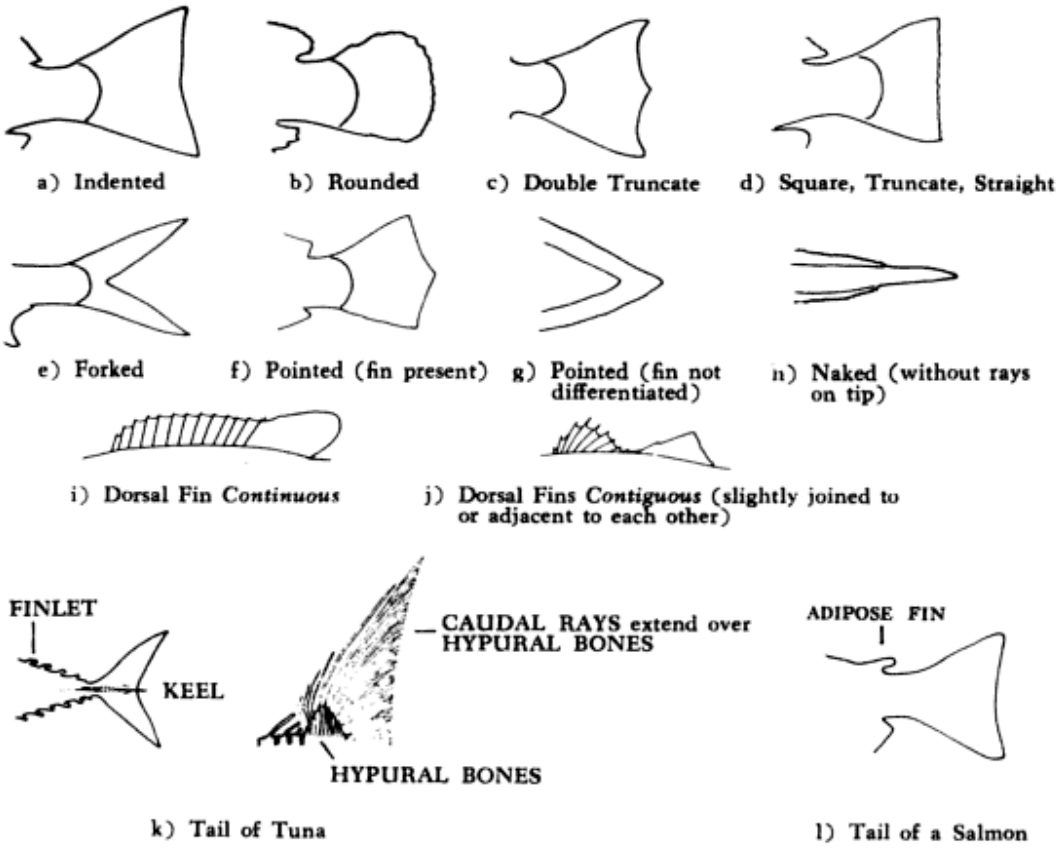


Figure 2.2: 12 examples of the tail and dorsal fin shapes and construction [Miller and Lea, 1976].

formation, marine biologists have to use many tools to examine the appearance and quantities of fish. For example, they cast nets to catch and recognize fish in the ocean. [Zompola et al., 2008] collected *Anguilla anguilla* (L., 1758) by using fyke nets for two years. These observed fish reveal the inland ecosystem characteristics of the glass eel short-term freshwater migration along the Atlantic coast of southwestern Europe. The authors investigated the correlations between environmental factors and the decline in European eel recruitment. Marine biologists also dive to observe underwater environment, using photography as introduced by [Caley et al., 1996]. They record underwater images with caution, so as to not interfere with fish activities. They investigate the long-term dynamics of marine stocks which are used for commercial fish stock estimation. Alternatively, marine scientists combined net casting with acoustic (sonar) [Brehmer et al., 2006] for monitoring *Amphidromous* fish school migration.

They used sonar to detect the swimming characteristics and estimate the abundances of fish school. The cast net is also employed for direct sampling which confirms the presence of the *Dicentrarchus labrax* schools. [Katselis et al., 2007] utilize six lagoons and the fish traps located at the lagoons to observe the relationships of fish migratory behaviours and various climatic variables. Their work includes four numerically dominant euryhaline fish species in the Messolonghi-Etoliko Lagoons.

2.3 Machine learning and computer vision applications in fish recognition

Traditional marine analysis methods require onsite observation or even anatomical dissection to locate the ichthyology characteristics. Nowadays, much more convenient tools are employed, such as hand-held video filming devices. Embedded video cameras are also used to record underwater animals (including insects, fish, *etc.*), and observe fish presence and habits at different times [Nadarajan et al., 2011]. Video recording has produced large amounts of data, and it requires informatics technology like computer vision and pattern recognition to analyse and query the videos. Statistics about specific oceanic fish species distribution, besides an aggregate count of aquatic animals, can assist biologists resolving issues ranging from food availability to predator-prey relationships [Rova et al., 2007]. This section introduces some useful methodologies in this area and their applications. Figure 2.3 indicates three stages of the common procedures for the parametric approaches. There are also some non-parametric classifiers, *e.g.* K nearest neighbour algorithm. These approaches employ some state-of-the-art computer vision techniques to the fish processing area, including detection, tracking and recognition.

In the computer vision literature concerning fish recognition, there are roughly three groups of theories regarding the underlying input data: dead fish, live fish in constrained environments and live fish in open water. Section 2.3.1 provides a brief review of recognition systems for dead fish. They are either still or acquired on a conveyor. Section 2.3.2 discusses some well-known fish recognition applications in constrained environments. Finally, Section 2.3.3 compares some systems related to freely swimming fish.

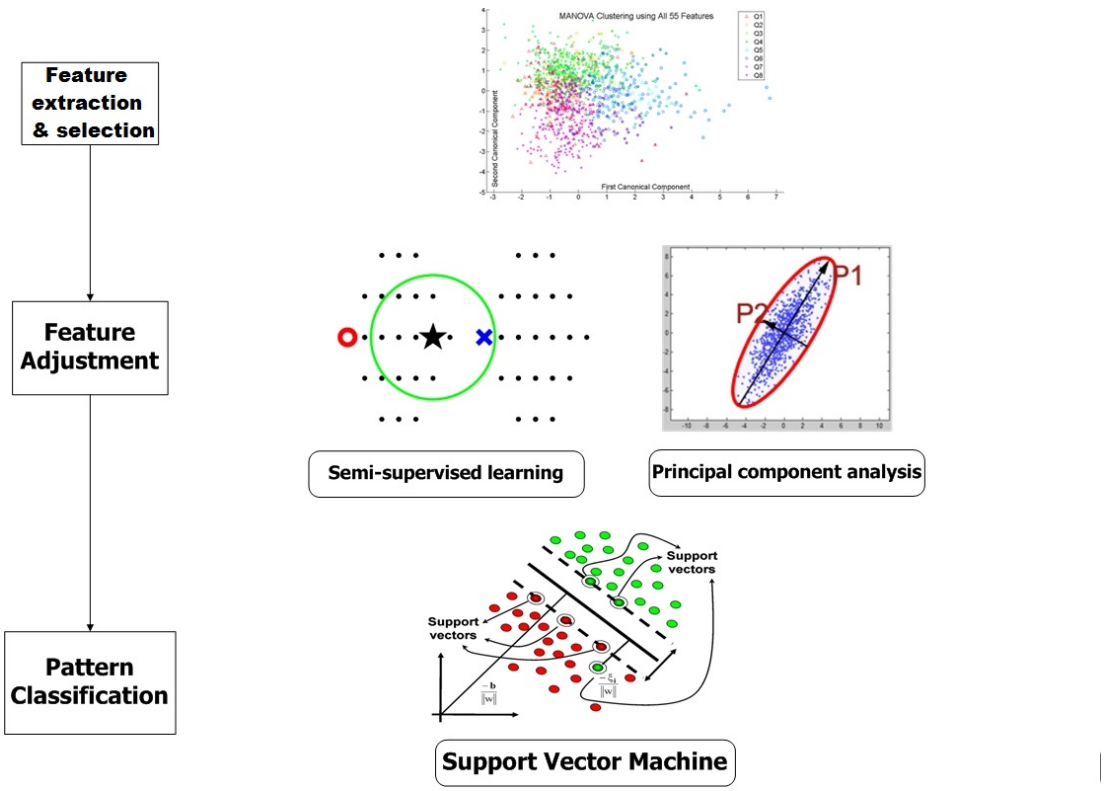


Figure 2.3: Some common components used in fish recognition algorithms for feature extraction & selection, feature adjustment and pattern classification.

2.3.1 Fish recognition applications for dead fish

The recognition applications for dead fish images, which are located in an expected observation area with fixed distance and pose and direction, are obtained by processing the still fish objects in a clean background. For example, [Zion et al., 1999] proposed a moment-invariant based method for fish species recognition. The method focuses on three species: common carp (*Cyprinus carpio*), St. Peter fish (*Oreochromis* sp.) and grey mullet (*Mugil cephalus*), all of which, in normal cases, live together. In the fish detection step, the fish images are grabbed through a transparent tunnel. The authors use background thresholds to determine fish, which extract background samples from first and last rows and columns. The experiment used the equipment shown in Figure 2.4. To avoid light distortion, they only use the green band of the image and calculate two boundaries' moment invariants. The prior probability distribution model of each species is considered and then a decision tree based on this information is built. As fish species always have an obvious bias distributions, which take top 80% quantities for top 20 species, prior information promises effective performance. In 124 sample

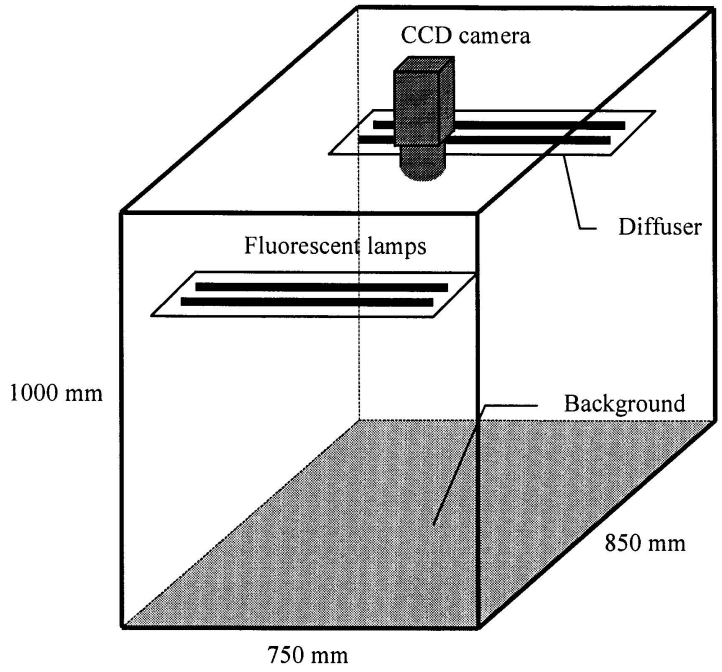
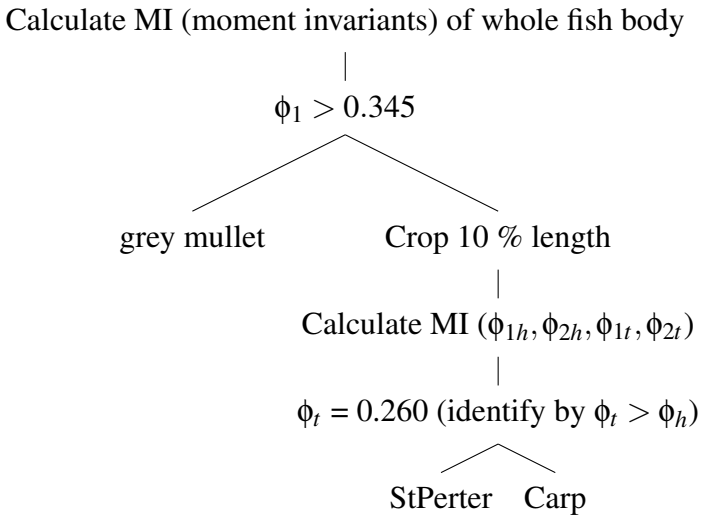


Figure 2.4: Equipment of fish image capture [Zion et al., 1999]

images, a 4-fold cross validation method is used, and it achieves a recognition rate of 100%, 89% and 92%, respectively for the three species. In the result analysis, correlation coefficients show a high relative connection between these species that are 0.954, 0.986 and 0.986, respectively.

Decision tree:



The decision tree shown above is based on the prior probability distribution of fish knowledge. Common fish recognition algorithms are based on a general feature clas-

sification method, which ignores special prior knowledge of fish *e.g.* what are good features for fish recognition and how to classify the features into different species. Actually, the point is how to collect and evaluate the prior information and to stabilize the classification procedure while the number of species is increasing. This paper only considers 3 species and finds a good classification threshold, 0.345 of threshold ϕ_1 for instance. According to this consideration, it may be a reasonable solution to use prior information to build a coarse decision tree whose responsibility is to determine common and uncommon fish because common fish have prior proportion statistics. Furthermore, the proposed solution in the paper is based on dead fish. This factor simplifies the fish recognition problem, and makes fish segmentation easier than in real underwater environments.

Inherited from the PDM model, two important approaches, called the Active Shape Models (ASM) [Matthews and Baker, 2004] and the Active Appearance Models (AAM) [Cootes et al., 2001], are widely used. ASM constructs a local texture model to optimize shape matching while AAM makes use of a global texture model which is insensitive to the illumination changes. These models have a common problem. They rely on the accuracy of landmark points. The quality of annotation affects the final performance. Furthermore, their computational complexities cannot be ignored. [Larsen et al., 2009] presented a shape and texture based fish classification method. 108 dead fish images of three species are mentioned in the experiment: Cod, haddock and whiting. An active appearance model is used to generate shape and texture features shown in Figure 2.5 on a set of training data. Marine scientists annotate the training fish images, including contours of the eye and backbone areas. Using this prior information, an optimal Minimum Description Length (MDL) curve model is built. This model contains curve appearances and connections between them. After analysing a collection of shapes and texture, an invariant model is derived by combining various shapes and texture models. In the classification procedure, principal component scores and linear discriminate analysis are introduced. Finally, based on the two best combined modes of variation, this paper achieved a recognition rate of 76% using linear discriminate analysis. Although this paper proposed a traditional classification system based on the shape and texture descriptions, it mainly concerned the deformable object problem. When fish swim across the camera, the distance and angle between the object and lens both varied. This phenomenon also affects fish descriptions and triggers geometrical deformation. The main progress of this paper's solution is a flexible feature

model.

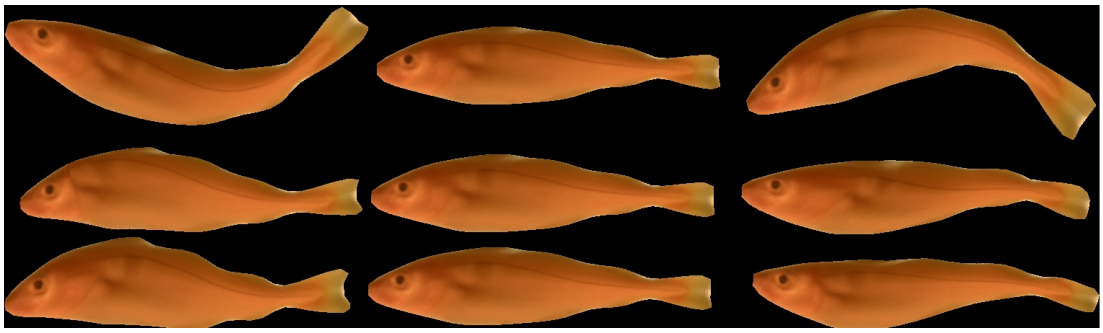


Figure 2.5: Active Appearance models (AAM & Landmark). [Larsen et al., 2009]

Principal Component Analysis (PCA) involves finding a transformation to convert the observed variables into orthogonal spaces. The technique is widely explored in the machine learning field especially for face recognition [Zhao et al., 2003]. [Rodrigues et al., 2010] employed the PCA method on the colour components of the YUV data as well as SIFT features for parameterizing shape, appearance and motion of fish species. They invested observed variables' correlations based on principal components. Each of these principal components is defined iteratively by extracting the highest correlative direction from the observed data. A K-nearest neighbour algorithm is then implemented as the classification method. In the experiment, each fish from nine different species is classified into a category of the smallest Euclidean distance of the PCA features. The first 10 higher variance principal components from images are preserved in the PCA process. These components are evaluated for their impacts in the overall accuracy. The authors have applied two algorithms, Artificial Immune Network (aiNet) and Adaptive Radius Immune Algorithm (ARIA), to cluster individuals from the same species. The experiment was carried on a database with 162 images of 6 species and achieves accuracy of 92%.

[Mokhtarian et al., 1997] used Curvature Scale Space (CSS), which is firstly presented by [Mokhtarian and Mackworth, 1992] that describes the index of the curves by using maxima or the concavity of the curve, to present the shape of marine animal images. Specifically, they authors treated this task as a image retrieval problem. Given an input image of a marine animal (including fish), the system finds the maxima of the computed CSS image, and evaluates the similarity to all images in the database by the aspect ratio, eccentricity and circularity. The system is tested on a database of 450 images. These images are randomly selected for a subset of 50 images. Some volunteers evaluated the shapes manually. The subjective results indicate that the results of the

proposed method are similar to some of the manually labelled images. An example of the CSS figure and their corresponding maxima points is shown in Figure 2.6. The first column shows three fish boundary images where the last two images have the similar shape. The normalized maxima of third column present a successful matching of these two shapes after a rotation of the data.

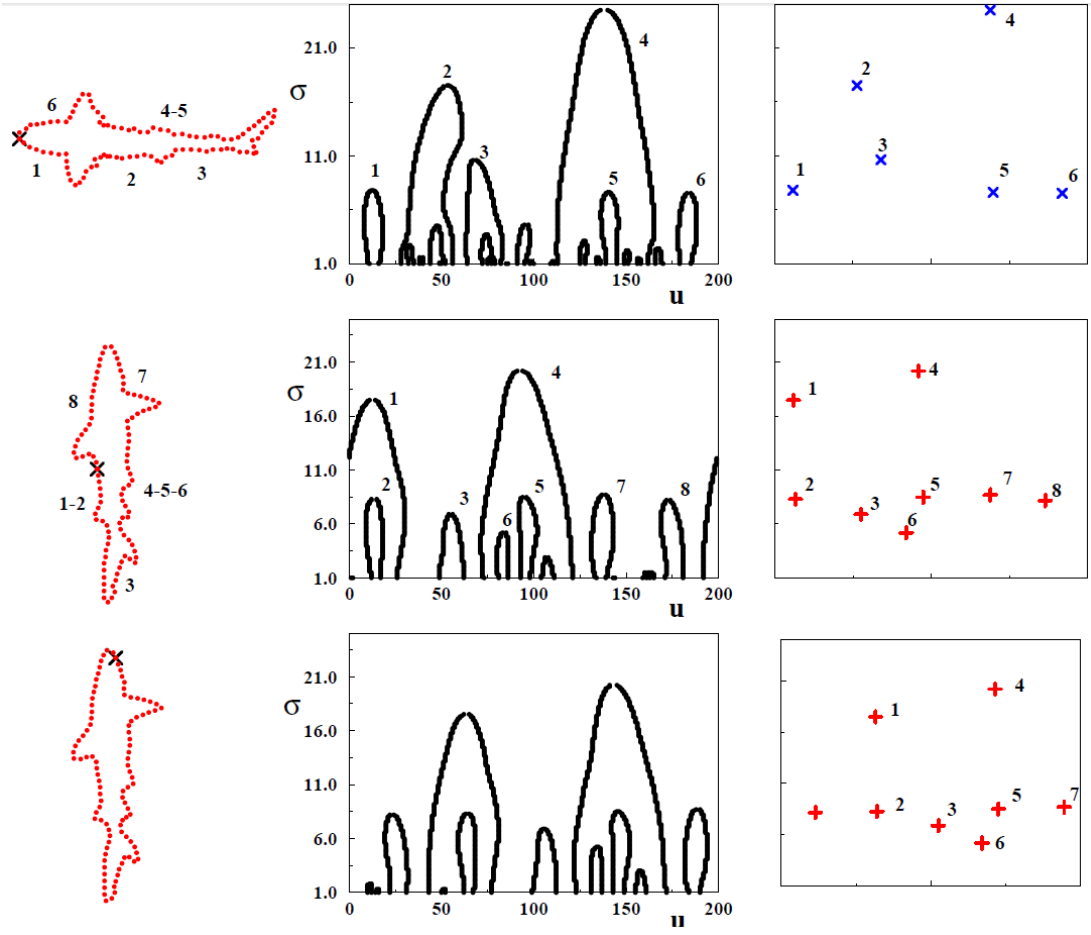


Figure 2.6: The fish images, their CSS figure and maxima points. first column: fish boundary with the marked starting point, middle: CSS figure, right: normalized maxima of CSS figures.

2.3.2 Fish recognition applications for constrained environment

In the view of fish detection and tracking theories, a constrained environment limits the searching area and provides prior knowledge, such as fish number and shape information. For example, [Benson et al., 2009] uses underwater images of a variety of fish species from Birch Aquarium, within a constrained fish tank. The authors propose

an automated computer vision fish species recognition system. This system aims at counting and classifying fish images using a classification method known as the Haar classification. Fish images are divided into positive (manually cropped) and negative set. These images are converted into grey scale and cut into 20x20 blocks, which are prepared for Haar-like feature calculation. Although Haar description and classification are widely applied in face recognition and content based image retrieval area, this paper employs these methods for the underwater fish recognition problem. The experiment used a 16 stage Haar classifier of 83 features in 1077 positive and 2470 negative images, respectively, and achieves 92 successful classifications in 100 test images. The proposed method mainly focuses on FPGA equipment for underwater fish recognition. The proposed system has evaluated their method on only very limited fish species, more specifically only the Scythe Butterfly fish. The adjustment of parameters for only one species of fish is simple, but there is no evidence showing their algorithm's suitability for identifying other species.

[McFarlane and Tillett, 1997] fit a 3D Point Distribution Model (PDM) of fish to stereo images. A shape template is trained from fish shape points while the principal modes of variation are also generated to fit the strength and proximity of local edges iteratively. The PDM model aims to resolve shape flexibility issues by creating the eigenvectors to represent the distortion from mean data. The parameters of the linearly combined eigenvectors are used to compute the corresponding score between a new shape and the model. Data always has noise. In the computer vision area, noise causes low-contrast, distortion and uncertain affine transformations. It is more complicated to tackle shape matching tasks with noisy data. Moreover, the shape of an object may vary considering different angles and distance. It demands a more flexible matching algorithm and requires a statistical model learned from the example shape set. The training data provides the information with which to build a mean geometric shape while the flexibility of the PDM model encodes the geometric variation. These models are based on landmark points which refer to annotated points on the contour. These annotating points are processed using the generalized procrustes analysis. After training, a new shape is divided into two parts: mean shape across all training images and scaling values for each principal component. The scaling values are generated by the PCA method from analysis of the covariance matrix across all training shapes. The proposed method successfully fits 19 out of 26 (73%) test images.

[Toh et al., 2009] used a direct way to count fish automatically. Firstly, the image

is converted to a binary image and then the fish positions are found by marking the blobs. After background estimation, this system subtracts the background from the adjusted image. Secondly, low-level image information is collected from the blobs, such as area and position, *etc.* These blobs are filtered by counting their pixels. Those blob of too small or too large size are eliminated. The average number of fish over all frames is then recorded. Blobs that have an area lower than 140% of the median area are classified as having only one fish. Experimental results show that the correct number of fish can be obtained for a school of 5, 10, 15, and 50 fish. For the 5 and 10-fish videos, all frames register the correct number. When 15 fish are used, the accuracy is reduced. The accuracy is dropped to 80% when 50 fish are exploited, using median value of area. As the fish counting procedure plays a significant role in the automatic fish processing system, the performance of this component would affect the fish recognition result. The biggest advantage of the proposed solution is its efficiency. According to the Fish4knowledge project, the estimated number of fish per frame is not more than five. The proposed technique could be employed to determine the quantity of fish. Although it achieves a good result, this solution uses only one threshold to generate binary images which would lack robustness to environmental changes (light, blurring, *etc.*).

[Morais et al., 2005] proposed an algorithm framework based on "BraMBLe" (a robust Bayesian multiple-blob tracker). The "BraMBLe" system tracks and counts fish by comparing different appearances of each potential target, and these corresponding blobs are tracked with a Bayesian probability distribution function. Their job is to monitor objects entering, leaving and moving. In this paper, a fish is modelled as an ellipse and its state is described by 8 parameters: the coordinates (x_c, y_c) of the ellipse's central point, the half lengths (a, b) of the major and minor axes of the ellipse, the angle θ that measures the ellipse's rotation with respect to x-axis, label r and the velocity components (v_x, v_y) of the fish. These parameters are calculated from the pixel/blob attributes. There are no colour/texture features involved. In the process, the system trains an appearance model off-line and describes each species of fish with the 8 parameters. Then it divides each image into W locations spaces with intervals. Each location block is computed by a 4-component Gaussian mixture model. It also uses a multi-blob likelihood function $P(I||S)$ to estimate the distribution function. In the tracking procedure, the tracker searches for potential targets by computing the likelihood between blobs and pre-learned foreground models while the pixels outside the blobs are

similar to background according to a background model. An example is shown in Figure 2.7. More specifically, it aims to evaluate the formula $P(S_t||I_{1...t})$ by using a particle filter to approximate and consider N random hypotheses as predicted samples. These hypothetical samples are assigned weights with log-likelihood results. Based on the likelihood distribution, the predicted centre is estimated at the peak point on the distribution map. The fish counting component is simple because of the reliability of tracking component. It counts fish within counting regions. The approach is robust under significant environmental changes and occlusions. Unlike existing fish-counting methods, this paper proposes a model based on relevant information about characteristics of different fish species. This information includes swimming ability, time of migration and peak flow rates [Morais et al., 2005]. The "particle filter" is used to estimate the projection configuration. Pixel distribution models between the fish object and background are discriminative. The distance between fish centroid and a random sample centre determines the centre of particle filter mass. The method in this paper was designed for bodies of water with different varieties of fish so they include some form of fish species recognition algorithm to count the number of each species. Thus, their suggested methods are computationally intensive and will increase costs and slow down the counting speed. For farmed fish counting, the problem is simplified because it only involves one species of fish which looks similar in shape and size to each other. As discussed above, the total number of fish may vary because fish may swim in and out of the video frame, in the fish tracking component. The core algorithm tracks the 8 parameter ellipses that stand for a fish configuration. That allows tracking multiple fish. In addition, fish tracking algorithms are also used in the counting of fish. This procedure focuses on a multi-target likelihood function and a randomly chosen set with N particles (in the experiment N=2000, specifically). The tracker shows a remarkable performance in a constrained environment (fish tank), while the classifier achieves an 81% accuracy which is a remarkable score.

In order to reduce the weighty task for marine biologists to identify fish from raw data, [Lee et al., 2003] propose an automated system to classify fish species and monitor migration. The proposed FIRM system is designed at the fish passageways so that fish are guided through a narrow passage and images can be taken at a close range. This paper aims to be invariant to normal distortions (3-dimension rotation, size, and shape deformation, *etc.*). The authors calibrate their system with an hourly updated reference image of empty water which is subtracted from subsequent images. Despite the success

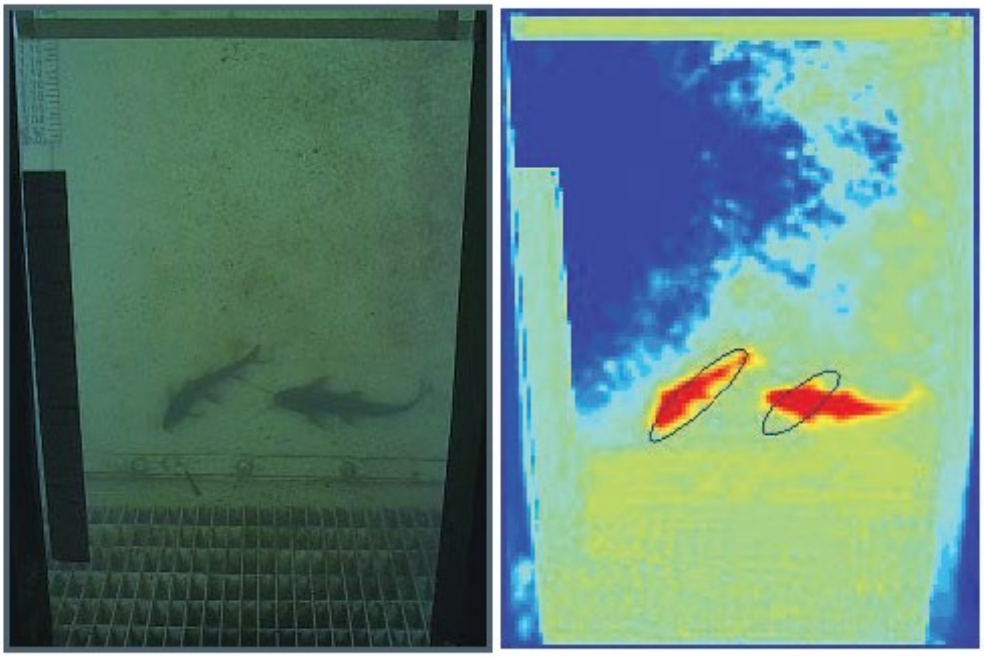


Figure 2.7: Log-likelihood ratios of foreground and background models. [Morais et al., 2005]

of these methods, the measurement has to be conducted in a controlled environment with known and fairly stationary backgrounds. Fish are detected by using differential motion analysis and closed shape contours are extracted from different images. To achieve 3-dimensional invariance, the authors focus on finding significant landmark points of the fish. The main problem of this step is the inaccurate position of the landmark points. Redundant data points were removed by a shape analysis algorithm. Finally, landmark points are determined by a curvature function analysis. This system records 22 images, from 9 different species. This system used an average frame as the background with a fast eight-neighbour contour trace algorithm to detect fish in images. For each detected fish, it extracted the fish contour of each image by using a closed boundary curve as well as fish shape characteristics, including Adipose, Anal, Caudal, head and body shape, length/depth ratio, and developed a new shape analysis framework for edge noise removal and detecting redundant data points (short straight lines) as in Figure 2.8. The authors located critical landmark points using a curvature function analysis, and these landmark points are used for fish contour segmentation parts. Then, the classification component employs the segment results for recognition of fish species. The classification component has three functions: Decision tree, Curve

Segments of interest and Feature vector distance evaluation. The result shows that this solution achieves high accuracy (all 7 test sample are correctly classified).

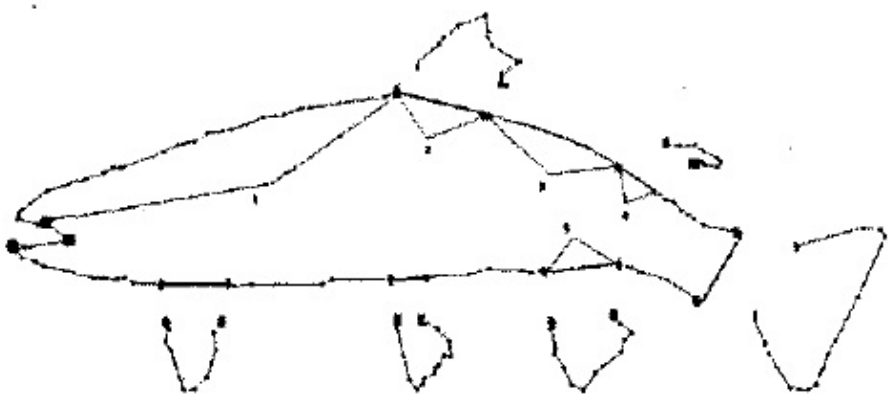


Figure 2.8: Fish contour divided into different parts. [Lee et al., 2003]

2.3.3 Fish recognition applications for open water environment

This section reviews some well-known approaches and common ideas found in machine vision modelling of live fish recognition from open water observations. In this environment, fish are freely swimming, and the background may also change. The recognition system has to deal with affine transformations and distortions such as scale, rotation, illumination changes, and blurring.

[Edgington et al., 2006] addresses the problem of computational complexity in the saliency map generation, and then detects and tracks and classifies animals in underwater videos. Firstly, this paper uses a selective attention algorithm that is initialized by pre-selecting salient targets. Two algorithms, the luminance normalization algorithm and the subtraction of the average background method, are combined. Frames are separated into the foreground and background by using average frames and graph cuts. During the detection procedure, the saliency based method is employed together with filter normalization. A linear Kalman filter is used for tracking visual events. Some special strategies are presented to reduce the complexity of multi-target tracking. More specifically, if an object is observed in several frames, the tracking component creates a visual event and passes this event to the classification component to determine the class of this event. The proposed system could only process one frame in every five due to the computational complexity. The authors have to choose the trade-off between

performance and efficiency. The classification component consists of two technologies for three training classes: a feature vector based on Schmid invariants and a Gaussian mixture model. By using the labelled training data from the MBARI's annotators, the salience detection result achieves accuracy of 90% on a data set of 200 detections. Their recognition result achieves a recall of 90% on 210 *Rathbunaster californicus* fish.

Support Vector Machine (SVM), as firstly presented by [Cortes and Vapnik, 1995], seeks the data samples that exist at the classification boundaries and could maximize the classifier distance. [Rova et al., 2007] apply the SVM algorithm to the fish recognition, and construct a texture based mechanism to distinguish two species of fish (the *Striped Trumpeter* and the *Western Butterfish*). The proposed algorithm is based on 2D textural appearance called the deformable template object recognition method ([Belongie et al., 2002]). In such deformable template matching research, the Canny edge detector is used, then shape contexts are combined with a Minimum Spanning Tree (MST), a large-scale spatial structure preservation) algorithm. By choosing general graph matching methods, each pixel is matched with the lowest global cost. In order to achieve better robustness, two procedures were introduced respectively for images exhibiting sparse edges: distance transforms [Felzenszwalb and Huttenlocher, 2005] on dynamic programming and four iterations of image warping. Each image is convolved with a 3-pixel-tall vertical central difference kernel, and the system uses filter responses in the input feature vector. The two templates, one for each type of fish, are built separately, and each query image is warped to both templates as shown in Figure 2.9. The proposed method has a 90% accuracy on a data set of 320 images. It focuses on a tree-structured spatial constraint of Canny edge points optimized using distance transformations while deformable template matching is also employed to align template images in addition to shape context matching. The proposed methods are effective distance transform matching algorithms. Finally, iterative warping is applied to the query images to improve the performance of the texture-based classifier. In the experimental results, both the linear and polynomial SVM kernels warp the images into alignment with a template prior to classification. This improved the classification accuracy by up to 6% (90% versus 84%). The merit of this paper is the combination of tree-structured and shape context matching, which aims at creating affine invariant features in the fish species recognition procedure. [Ma et al., 2000] discusses the robustness and image application of the minimum spanning tree. The tree-structured spatial

constraints perform well in experiments. There are still some issues in this paper - the details of using iterative warping and distance transform are not well described. After checking the distance transforms proposed in [Felzenszwalb and Huttenlocher, 2005], the proposed algorithm uses Mahalanobis distance and dynamic programming to seek an optimum result. Furthermore, the authors match the shape context from vertices in model images to each point location in the query image. Thus, the computational complexity becomes huge with the growth of the image resolution.

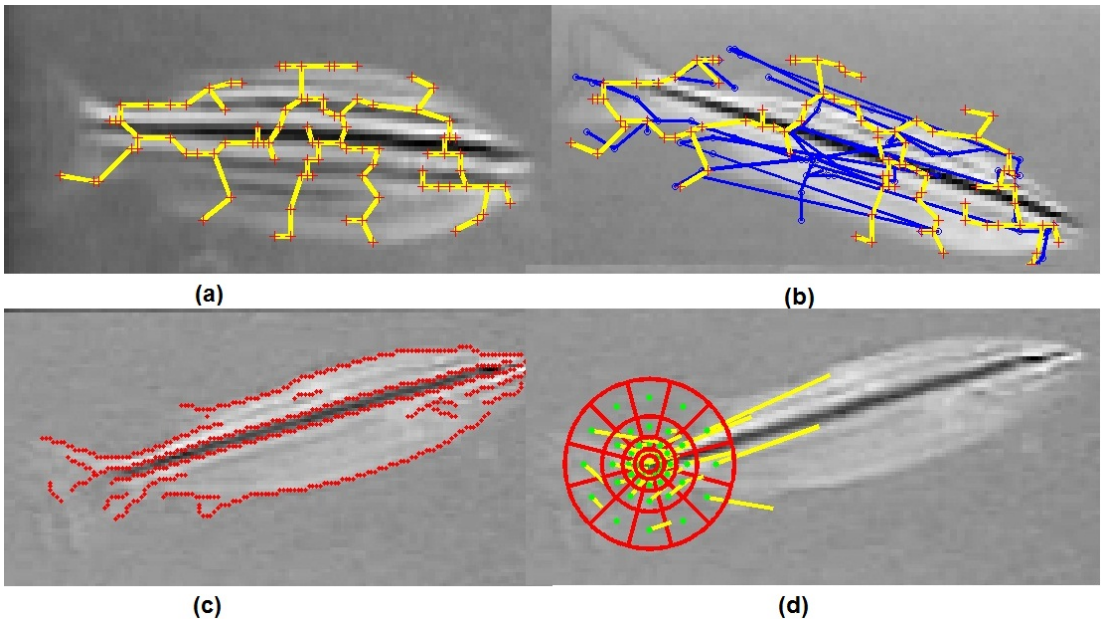


Figure 2.9: (a) original MST (b) estimated correspondences (c) Detected edges (d) A shape context. [Rova et al., 2007]

[Spampinato et al., 2010] proposed an automatic system to help marine biologists understand fish behaviour by classifying fish species. Automatic fish recognition systems are beneficial to underwater fish research. Firstly, the authors used a moving average algorithm and Adaptive Gaussian Mixture Models with Adaptive Mean Shift for tracking. Secondly, they combined two types of features for fish classification: Texture Features and Shape Features (Curvature Scale Space, as shown in Figure 2.10). To improve robustness, an affine transformation is applied and this technology presents fish in multiple views. After feature extraction, this paper employed PCA to reduce the dimension from 120 to 24. The system is tested on a database containing 360 images of ten different species. The database contains 14 streaming images and 18 affine transformation images for each species. The result achieves accuracy of about 92%. The results are achieved using cross validation classification and are analysed in order

to research for the connection between behaviours and species. There are two steps in the fish trajectory analysis system. The goal of the first one, using preprocessing, is to generate a trajectory description and clustering. In order to sub sample input vectors, the authors use the Douglas-Peucher algorithm. After that, the I-kMeans algorithm is employed to cluster trajectories.

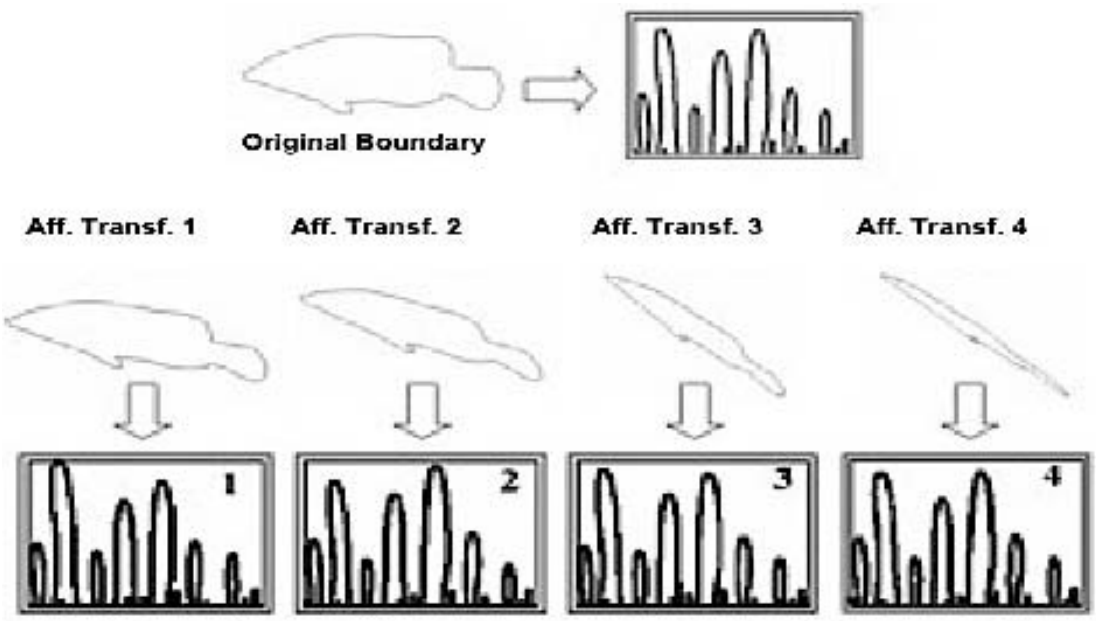
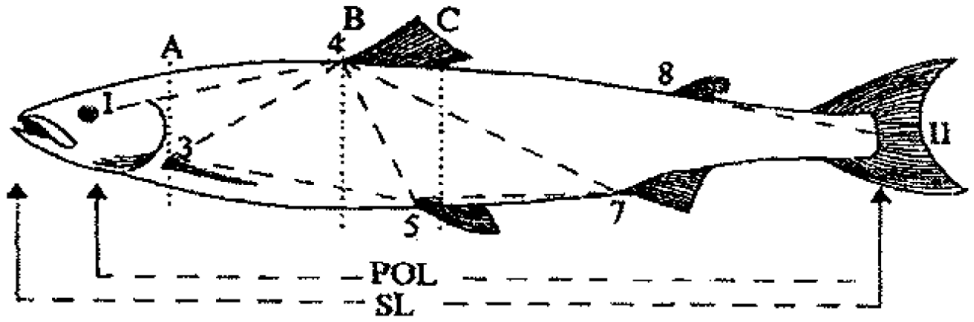
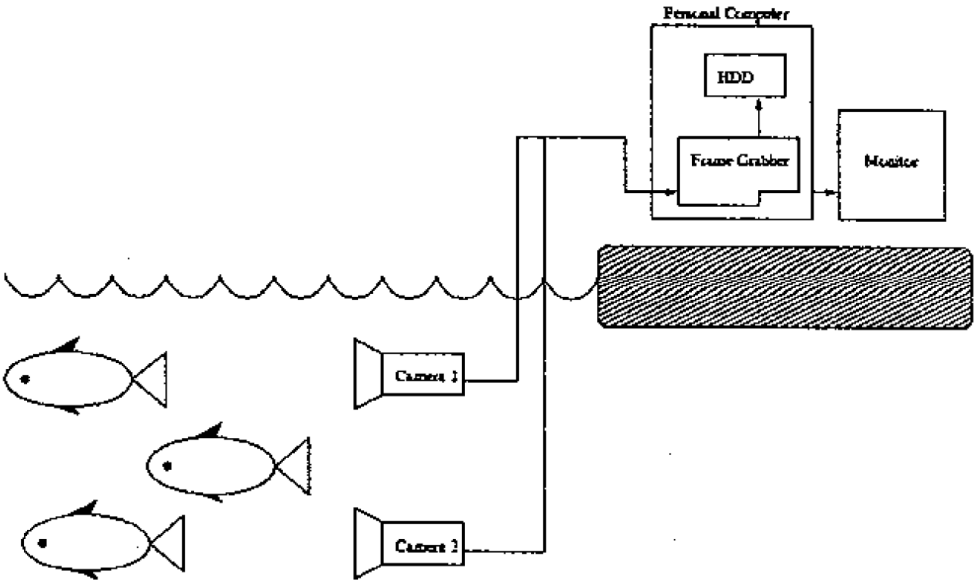


Figure 2.10: CSS images for the contour of Pseudochilinus Hextataenia species for 4 different affine transformations. [Spampinato et al., 2010]

[Chan et al., 1999] employed a 3D PDM approach to monitor fish in an underwater environment for feeding strategies in fish farms. It assists the salmon farmers to decide on feeding, grading and harvesting. The PDM method has some shortcomings, such as the need for annotation which makes it time consuming. An n-tuple classifier is implemented to overcome these limitations and give the initial estimate of fish, from its unique characteristic of flexibility, simplicity and efficiency. In the experimental part, the authors have trained the proposed system with five fish head images of resolution 96x38 pixels. WISARD, a trainable binary pattern classifier, is introduced as a binary classifier. The WISARD algorithm builds a Look Up Table (LUT), which holds information about the classification pattern. A score of the test image is computed after performing a pixel-tuple mapping into n-tuples. The test dataset contains 16 underwater image sequences. The authors also showed the potential of the WISARD algorithm in the whole estimation process, and and the best accuracy (54%) is achieved when a decision boundary is drawn at WISARD score equal to 0.27.



(a)



(b)

Figure 2.11: Underwater stereo system (a) and the Salmon truss network (b) [Chan et al., 1999].

2.3.4 Other marine species recognition methods

[de Zeeuw et al., 2010] constructed a computer-assisted system, capable of automatically matching pink spot photographs against a database of earlier encounters, with the purpose of identifying individual and migrating leatherback (sea turtles). Based on the Scale Invariant Feature Transform (SIFT), this system provides a new direction for marine species recognition. The Bag of Features (BOF) method is wide spread recently, and many experiments have shown its robustness and accuracy in content based image retrieval area, especially in affine transformed target retrieval [Mikolajczyk and Schmid, 2005]. This paper applied the BOF method (on SIFT features) for turtle recognition. It used opponent based colour contrast as a feature to improve recognition performance and remove false negative key points. Image cropping also improves the performance but remains a huge task for the marine biologists. An automatic saliency detection system would be better than cropping due to the saliency detection primarily based on the quality of the image. The authors test their methods to binary classification using a 76-image database. The separation between result scores for matching and non-matching pairs looks promising with one false negative and 4 false positives. Experiments on another database (151 images) gets a perfect classification.

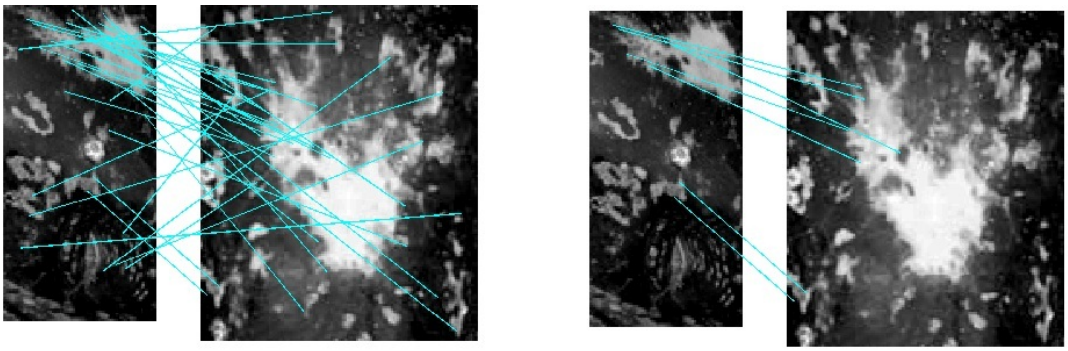


Figure 2.12: Matching the pineal spots. Left: Using Lowes matching algorithm. Right: Largest affine consistent constellations. [de Zeeuw et al., 2010]

[Cline and Edgington, 2010] focuses on automatic underwater image-processing, including detection, tracking and classification procedures. It uses Remotely Operated Vehicles (ROVs). Firstly, in the fish detection component, a saliency map is built from the independent salient patches. In this scheme, an image is decomposed into seven channels and kept for peak points. By using a winner-take-all neural-network,

the system implements an inhibition-of return mechanism. After detection, the authors introduce graph cuts and nearest neighbour tracking for fish segmentation. The threshold, which determines the performance of fish segmentation, is generated by the OSTU method [Otsu, 1979]. The authors propose a concept called an "interesting" event, and use it to stand for a salient object. An event is tracked over several continuous images. A neurotrophic model is introduced to analyse this kind of event and achieves significant improvement. After this procedure, a Bayesian classifier using a Gaussian mixture model determines the probability distribution. It achieves a recognition result of 100% accuracy for a type of deep-sea benthic animal called *Rathbunaster californicus* and 95% accuracy for the *Parastichopus leukothele*. This system utilizes some interesting methods, such as graph cut, saliency map and the Otsu method. The authors combine many different algorithms and achieve an excellent result on a specific species. But the result of still images does not perform well. For "Echinocrepis rostrata" and "Beathocodon", this system only achieves 24% and 26%, respectively. Due to the complexity and incoherence between each component, this system attaches too much importance to some species while fails to account for others. Moreover, this system involves many parameters and is hard to choose appropriate settings.

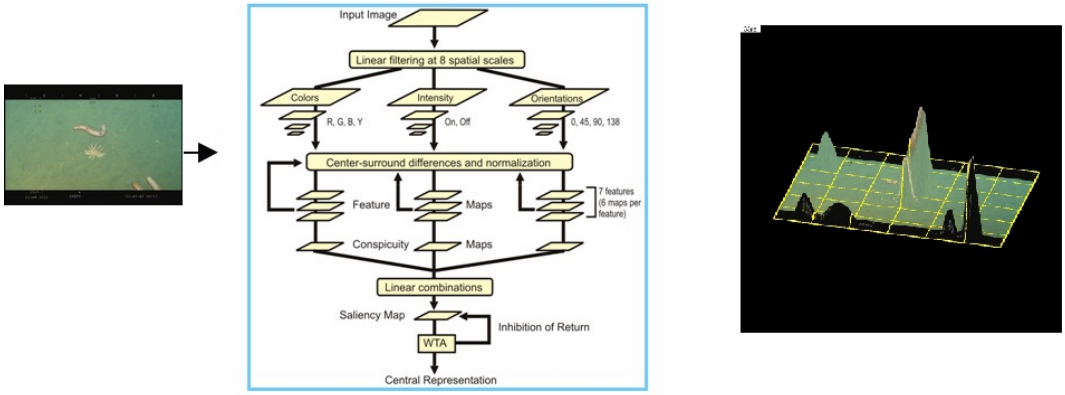


Figure 2.13: Saliency map from a single video frame. [Cline and Edgington, 2010]

2.4 Introduction to the Fish4Knowledge project

My research project is part of the Fish4Knowledge Project that is funded by the European 751 Union 7th Framework Programme [FP7/2007-2013] and by EPSRC [EP/P504902/1]. The Fish4Knowledge aims to increase the ability of researchers to analyse massive sets of underwater data (from 10E+15 pixels to, approximately, 10E+9 detected fish). It

designs an automatical system to analyse recorded media of long-term observation and extract necessary information of the marine animals. These processed data are stored in distributed storages, and a specialized user query interfaces provides high-level interpretations, such as abnormal climate events or ocean pollution. Our fish recognition system provides the basic evidences of marine animal analysis, as the foundation of higher level investigation. Fish detection and tracking results are the prerequisites of fish recognition, and the recognition system itself is the basis of fish behaviour analysis and counting. We selected 24150 fish images that belong the top 15 species. All fish are manually labelled by following instructions from the marine biologists, presented by [Boom et al., 2012]. Figure 2.14 presents a sample interface for annotators. Note, these fish images are low quality because of specific environmental and application context: blurred, occlusion by other fish or background objects, which include coral, the sea flower and open sea. It is also constrained by the underwater capturing devices due to technical difficulties, with 320x240 resolution and 5 frames per second. The averaged size of fish bounding box is 96x104, while the mean and median value of fish body size is 1447 and 1093 pixels, respectively. Also the designed system has to deal with various environmental factors, including light distortions, murky water, frequently illumination level changes arising from the ocean surface. Our methods have to balance the tradeoff between robustness and accuracy.

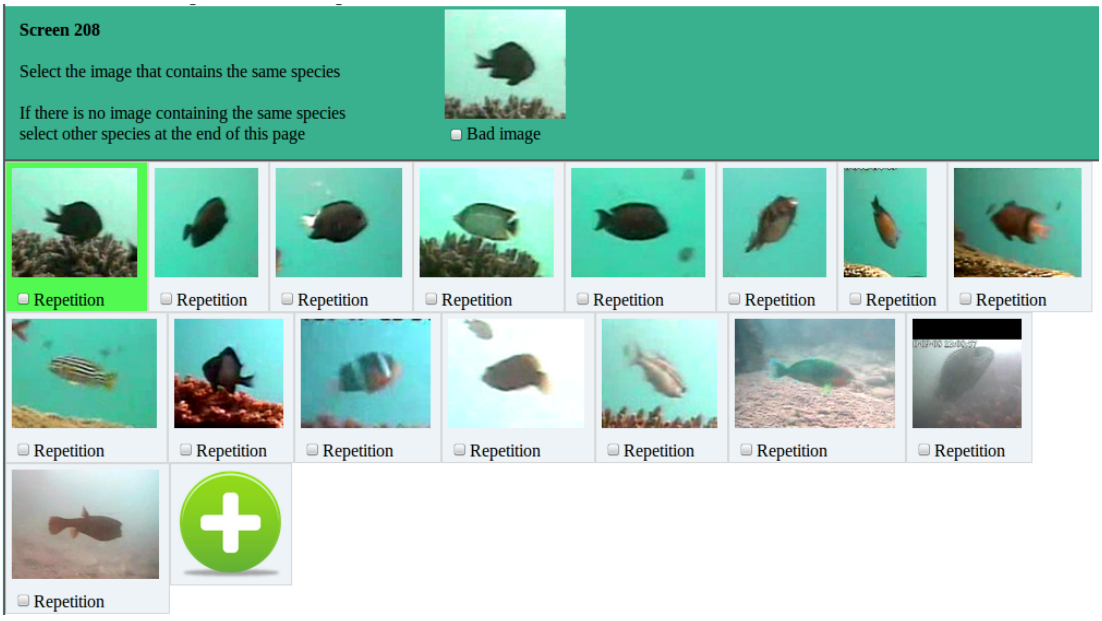


Figure 2.14: A sample interface for annotators. Each time, we label the fish images of a whole cluster.

2.5 Literature summary

Current fish recognition approaches mainly focus on dead fish [Larsen et al., 2009], fish in tank [Lee et al., 2004] or on a conveyor system [Ruff et al., 1995]. Some of the underwater solutions only classify a few species [Benson et al., 2009, Edgington et al., 2006]. These systems mainly employ global appearance shape descriptors. Not many fish species classification approaches have been investigated in the natural environment [Spampinato et al., 2010].

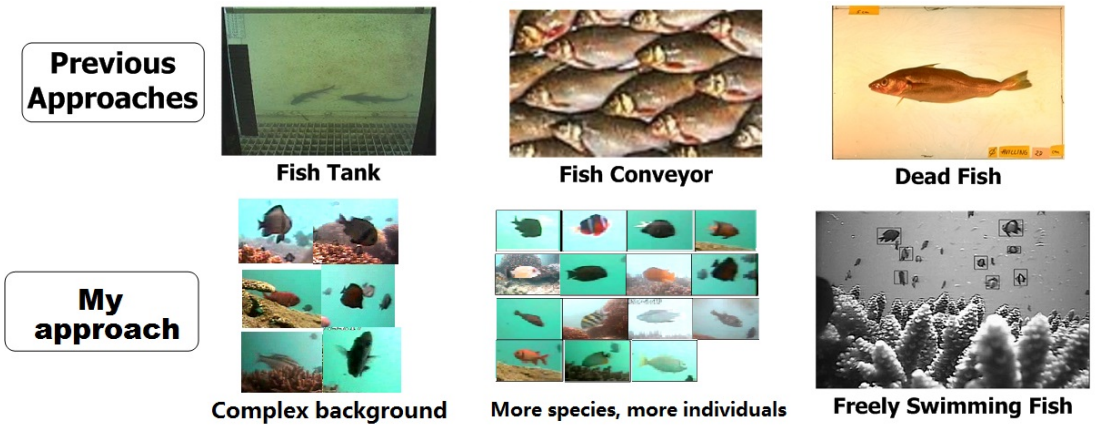


Figure 2.15: Problem Comparison between previous approaches and my project

Unlike the simple experimental environment found in the majority of previous work, my project has to deal with a more complex situation. Some vital factors differentiate my project from previous approaches. Table 2.1 summarizes the literature approaches. Figure 2.15 gives an indication of the differences between the work surveyed and that which we have carried out. These differences influenced our choice of features and also influenced the choices of machine learning methods. The first consideration is that the proposed method should be robust to noise and distortions, which are almost inevitable in the underwater videos. The second factor is the computational complexity. The huge amount of data (as described in Chapter 1) demands an efficient algorithm framework. Last but not the least, the appearances among diverse species of fish have different distributions. There are about 20+ species of common fish in our database, and the

abundance of fish images from all species are greatly unbalanced. So the precision of fish recognition is challenging and therefore becomes the most crucial factor in our research considering the low quality of recorded videos. Table 2.2 summarizes more details about the differences.

With these considerations in mind, we summarize the state of the art of fish recognition approaches and discuss their advantages, as well as their disadvantage, below:

- Current popular systems conduct fish recognition by using only the general and global features. But some descriptive fish features, especially the ichthyic descriptions, correspond to a particular species and can be used as important factors to recognize fish. We explore these descriptors to improve the classification

Table 2.1: Summarization of the literature approaches.

Literature	data type	application	# sp.	# instance	accuracy
[Zion et al., 1999]	dead fish	recognition	3	124	94%
[Larsen et al., 2009]	dead fish	recognition	3	108	76%
[Rodrigues et al., 2010]	dead fish	recognition	6	162	92%
[Mokhtarian et al., 1997]	drawing shape	retrieval	-	450	-
[Benson et al., 2009]	live fish (fish tank)	recognition	1	100	92%
[McFarlane and Tillett, 1997]	live fish (fish tank)	recognition	1	26	73%
[Toh et al., 2009]	live fish (fish tank)	counting	1	50	80%
[Morais et al., 2005]	live fish (fish tank)	counting	1	-	81%
[Lee et al., 2003]	live fish (pipeline)	recognition	9	22	100%
[Edgington et al., 2006]	live fish (open sea)	recognition	3	210	90%
[Rova et al., 2007]	live fish (open sea)	recognition	2	320	90%
[Spampinato et al., 2010]	live fish (open sea)	recognition	10	360	92%

Table 2.2: Comparisons between the my project and former approaches

Previous approaches	**My project**
Constrained area	Natural environment
Dead fish	Freely swimming
Fixed distance	Various Distances
Small number of species	20+ species
Equal size of species abundance	Greatly imbalanced dataset

performance.

- Common multi-class classifier could be considered as a flat classifier because it classifies all classes at the same time and omits the inter-class correlations. A shortcoming of the flat classifier is that it uses the same features to classify all classes without considering that some classes have certain similarities and can be better separated by some customized features at a later stage.

- Almost all the existing approaches identify fish species using features from single fish in one image. However, the extracted features from an individual frame are not always reliable because the features are easily affected by external environmental factors like motion blur or light reflection. Furthermore, fish may change direction and posture while swimming, which also impacts the representation of features. Therefore, we leverage the temporal property of our data set, deriving new features and exploiting information from multiple consecutive frames to improve performance.

- Around one billion of fish images are recorded in the Fish4Knowledge project. Previous approaches do not possess that sum of information due to computation and memory requirements. Furthermore, these images contain fish from new classes and false detections, *e.g.* blurred images, occlusion by other fish or background objects, non-fish objects (coral, sea flowers, *etc.*). Normal multi-class classifier identifies every test sample into one of the training classes. Although our fish recognition dataset covers the most dominant species of fish, there are still many observed fish from unmodeled species. These new species of fish images, as well as the false detections, are classified as known species and precision is thus decreased. Manual annotation work for these minority species is expensive because of the small proportion of these images, when compared to the major species. Thus, the reject option helps the fish recognition application in finding new species and eliminating false detections.

To sum up, current fish recognition techniques still have problems and require improvements in the above proposed issues. Live fish recognition in the open sea is fundamentally challenging because it is a complex situation where the illumination changes frequently. As a result, this task remains an open research problem. Prior research is mainly restricted to constrained environments. In contrast, our work investigates novel techniques to perform effective live fish recognition in an unrestricted

natural environment and presents an application of hierarchical classification with rejection method for 20+ species of freely swimming fish, with an accuracy of *c.* 97% on the top 15 species.

Chapter 3

Idiosyncratic feature extraction for fish recognition

This chapter describes the feature extraction methods that are implemented for fish recognition in unconstrained circumstances since the quality of underwater video streams affect the recognition accuracy by adding distortions and noise to the original image. The following section summarizes the traditional ichthyology characteristics such as meristics and morphometrics that are examined by marine biologists to identify fish individuals. It briefly discusses some popular features used for fish species identification. It also introduces some new idiosyncratic fish features. The pre-processing procedures are undertaken to improve the quality of features, including a Grabcut method for better segmentation of the fish inside the bounding box, a novel fish rotation algorithm to align the fish into the same direction. Afterwards, we give the technical details about our feature extraction algorithms and idiosyncratic fish descriptors. A combination of colour, shape and texture properties in different parts of the fish such as tail, head, top and bottom are extracted.

We observe fish images from underwater telerecording streams. These fish images record either the illumination values (RGB) or human colour perceptions (CIE) of pixels over the observing range. However, instead of using pixel values directly, computer vision techniques assemble the information of pixels into features, which are more independent to their circumstance and more reliable for further analysis. These types of feature are carefully designed, thus they are expected to present the domain knowledge as relevant information in order to provide comparable measurements, instead of the

original pixels. This situation could be described as the input data is too redundant to be evaluated, and it is processed by the feature extraction which transform the input data into a new representation set of reduced numerical matrix, so-called feature vector. For example, recognizing clown fish from black fish is a natural perceptional ability for human-beings. However, computers can only distinguish the fish from digital numeral data of extracted features. The feature input, which models the underlying characteristics of the given class, represents the hypothetical distribution of density probabilities where samples belong to the same group are neighbours in the feature space. For example, in fish recognition, some species of fish have specific colours, fin shapes, stripes or texture. The computer vision techniques exploit these colour/shape/texture similarities and present the similar samples in the same feature density distribution.

3.1 Related work

Traditionally, fish recognition is processed using ordinary fish features such as weight, length and width, *etc.* as presented by [Strachan, 1993a]. However, these features are only applicable for onsite measurement. It is difficult to measure these attributes from an underwater fish video because the size of marine animals vary according to distance from the camera and the body posture. Several physical factors such as absorption and scattering, reduced amount of light and poor visibility due to exponential light attenuation also affect the quality of the recorded video when light propagates, where the image restoration and enhancement techniques are preferable ([Schettini and Corchs, 2010]). In such complicated circumstance, the computer-vision-based features, which summarize the characteristics of an image itself, are introduced to analyse the semantic objects of the video stream. These features include image edges, statistical attributes of textures, local descriptors, shape context and curvature scale space features *etc.* (as presented by [Strachan et al., 1990, Toh et al., 2009, Walther et al., 2004] and shown in Figure 3.1). A table of useful fish descriptors is given in Table 3.1.

These features aim to be invariant to affine transformations and distortions such as scale, rotation, illumination changes, and blurring. This chapter discusses several types of frequently-used descriptors in the literature for analysing underwater videos. We

then propose our feature set: 69 types of features (2626 dimensions) are introduced as a mixture of colour/shape/texture descriptors to cover fish characteristics, with a pre-processing procedure for fish orientation which aligns the fish images to the same direction before further processing.

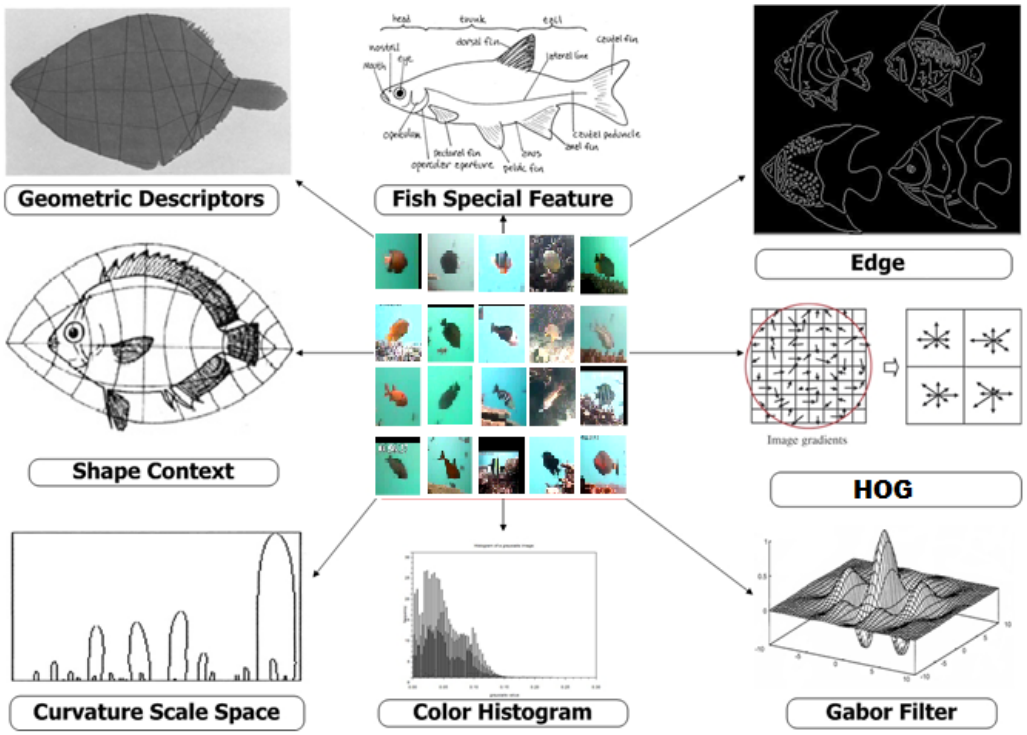


Figure 3.1: Features from different types are usually combined and used for marine applications due to complex environmental factors.

Table 3.1: Fish description table

Colour Section 3.1.1	Contour Section 3.1.2	Texture Section 3.1.3	Fish special Section 3.1.4
RGB	CSS	Gabor	Geometric Shape Descriptors
Norm RGB	Curvature Points	SIFT	head/tail
HSV	Point Distribution	PCA-SIFT	Translucency
HSL	Shock Graph	Covariance Matrix	eye/mouth/fin/rim
L_{AB}	ASM/AAM/MDL	Canny detector	Spots/Stripes

3.1.1 Colour-based features

Colour-based features describe the spatial (or temporal) intensity of the original image. [Zion et al., 1999] use RGB colour and shape features from 8-bit colour resolution images to deal with the shape-based retrieval problem. Their technique performs scale and rotation invariant retrieving of *Cyprinus carpio*, *Oreochromis sp.* and *Mugil cephalus*, by placing the fish on a conveyor belt. [Nery et al., 2005] investigate the effectiveness of features for the fish classification task. The colour features, *i.e.* YUV and HSI colour signatures of the dorsum and the ventral region of the fish, are included since they provide luminance and chrominance information in separate bands. Features are ranked and evaluated by their discrimination and uncorrelatedness in a Bayesian classifier, with six species of fish from the Rio Grande river in Minas Gerais, Brazil.

Although the colour-based features are intuitive from observation, we note the divergence of the perception model in the underwater environment. [Schettini and Corchs, 2010] discusses how the water medium absorbs and scatters light when it propagates. Figure 3.2 shows images recorded by the same equipment at two different scenes: water surface and *c.* 10m depth. The light model is changed in three aspects: diverse distortion of colours (image tends to be blue) as the water absorbs most red/yellow/orange energy at the depth of 10m; limited visibility distance due to attenuation of light by water or suspended solids; and low contrast of image and haziness. Even the colour distortion model itself changes at different depths because different wavelengths of light attenuate at different distance. These distortion factors remind us that the colour is diminished and needs to be restored/enhanced before feature extraction.

(a) . (b) .

Figure 3.2: Example of how light distorts from the surface to underwater environment.

Due to the distortion of light in an underwater environment, native colour values of image pixels are not suitable. Realistic applications (not placed on the conveyor or observed in the fish tank) should use colour features that are independent, or at least less affected, by the circumstantial factors. [Chambah et al., 2003] use an Automatic Colour Equalization (ACE) method to enhance the colour features for automatic live fish recognition in aquariums. The authors utilize underwater lightness/colour constancy and apply the global information from colour equalization to produce a stable perception which is invariant to the changes of mean luminance/colour intensity. In their experiment, hue, grey levels, colour histograms and chrominance values are used as colour features and improved by the ACE algorithm.

3.1.2 Shape features

Shape feature describes the boundary of the fish body and fins. This kind of feature is popular for fish recognition since it is less sensitive to lighting variations. More specifically, the shape features represent fish edge information describing the trend of changes along the fish outline. This trend is the relative value of edge pixels from their neighbours, thus, it is robust to the effects from the ambient lighting; even though the shape feature relies upon the size of the fish and its orientation, *e.g.* a unique fish presents various shapes when it swims in different directions, especially when it heads towards the recording camera. Figure 3.3 shows a set of detections from a whole trajectory of *Dascyllus reticulatus*. These recorded images illustrate that the shape descriptors may have large variation if the fish is swimming in an open area without constraints.

Figure 3.3: An example of fish detections from a whole trajectory of *Dascyllus reticulatus*. This fish changes its position and posture while recording. The shape of the fish is stable; its projection is not. The red contour shows the fish detection result and is used as the shape feature for further processing.

Therefore, in previous research, most applications require a fully controlled field of view (*e.g.* on a conveyor or in a fish tank or by using stereo equipment) or choosing a method insensitive to those variations.

Geometric Shape Descriptors, as introduced in [Strachan, 1993b], are fish shape grid descriptors relative to position reference. This method divides a fish body into a certain number of grids along the fish axis. These grids are invariant to fish size and direction. They describe the relative distance between certain parts of a fish, and demand high image quality and at least a complete fish contour. In an underwater video, a number of silhouettes are incomplete and it is difficult to localize the geometric position of features. [Strachan, 1993b] employ geometric shape descriptors with colour features. These features are used to classify fish into eighteen species. Some small fish easily bend and deform. The authors adapt a vertical grid along with the shape gradient. This kind of feature is considered to be a hybrid type because it divides a fish body into small blocks by the meaning of shape attributes and describes these blocks by the average R, G and B values.

[Hu, 1962] introduces the method of using Moment Invariants(MI) to identify visual patterns. According to [Hu, 1962], every image can be reconstructed by an infinite set of moments, and the magnitudes of invariant moments describe the statistical attributes of image shape. MI is considered to be a useful feature and widely used in computer vision applications [Flusser et al., 2009]. [Strachan et al., 1990] compares six invariant moments with two other methods (optimization of the mismatch and shape descriptors) for six fish species recognition: megrim, whiting, saithe, haddock, gurnard and herring. The authors conclude that MIs are better than the optimization method and worse than the shape descriptors. [Zion et al., 2000] employs MI features to build a decision tree and classifies three species of fish. In the paper, the authors discuss that MIs are preferable if a fish shape is significantly different from others (*e.g.* distinguish grey mullet from St. Peter fish and carp). Both fish recognition experiments in [Strachan et al., 1990, Zion et al., 2000] use part of the MI set, in which case the high order MIs do not have appropriate physical interpretation. There are other applications that integrate MIs with colour features. MIs could summarize the statistical attributes of the spatial average of image intensity, including low-level image features such as grey-level histogram features. These features are also invariant to colour distortion, image rotation and scale changes [Flusser et al., 2009]. [Spampinato et al., 2008] analyses underwater video texture and determines the frame environment (*e.g.* uniformity, entropy, brightness and smoothness) based on the features that include the statistical moments of the grey-level histogram.

Shape context describes the statistical radial histograms of edge points. These meth-

ods concern point correspondences and provide a way to evaluate shape similarity. [Belongie and Malik, 2000] developed the shape context algorithm in 2000. In the previous work presented by [Mikolajczyk and Schmid, 2005], the authors compare shape context algorithm with other common local descriptors, *i.e.* PCA-SIFT, SIFT, differential invariants, *etc.*, where the shape context shows high performance in most tests (only behind GLOH and SIFT), excluding those containing textured scenes and weak edge situation. [Rova et al., 2007] combines this kind of feature and constructs a minimum spanning tree to recognize two categories of fish: butterfish and trumpeter.

Curvature Scale Space (CSS), which is presented by [Mokhtarian and Mackworth, 1992], was proposed as a shape description for planar curves. This algorithm describes the index of the curves by using maxima or the concavity of the curve. In later research, as shown in [Mokhtarian and Suomela, 1998], this algorithm is robust to image rotation, translation, deformations and affine transformations. [Spampinato et al., 2010] develops the CSS images for contours of fish and extracts the first 20 local maxima of the CSS image as a feature vector. After combination with other features, the authors reduce feature dimension and use the reduced data to classify fish species. [Torres et al., 2004] compares the performance of CSS descriptors with fractal dimension, Fourier descriptors, moment invariants and Beam Angle Statistics in their invariance to fish shape characteristics and their ability to separate objects of distinct classes. The CSS method is robust and effective according to the experimental results.

3.1.3 Texture features

Unlike colour features which are sensitive to environmental factors, texture features illustrate the spatial arrangement of pixel values within the area of foreground object. These changes are either described by the local variations (edges or local descriptors) or statistical attributes (Co-occurrence Matrices). Many computer vision researches have comprised rapid development of the texture features since they are more robust than the colour features and carry more description than the shape features. [Mikolajczyk and Schmid, 2005] compares nine types of typical local descriptors in terms of their distinctive ability as viewing conditions change. The comparison is conducted under six distortions, including affine transformations, scale changes, rotation, blur, JPEG compression and illumination changes. The experiment shows that Gradient Location and Orientation Histogram(GLOH) achieves the best recall result, closely

followed by SIFT [Lowe, 2004]. This summarises local descriptors' performance and their recall ranks in object matching.

The Canny edge detector, which was introduced by John Canny in 1989 [Canny, 1986], is commonly used for fish recognition applications. This optical edge detection algorithm is a combination of several steps, and a psychological model explains how people observe the edge from an image. The algorithm consists of four steps: noise reduction, calculating the intensity gradient, non-maximum suppression and tracing edges using hysteresis thresholds. [Lee et al., 2004] extracts Canny edges and removes insignificant feature points. The remaining points are compared with contours in the database by using Fourier descriptions. The authors employ this method in fish recognition and migration monitoring system. [Rova et al., 2007] combine Canny edges with a minimum spanning tree. The authors use the strengths of shape context descriptors with distance transform for the deformable template matching to recognize two species of fish in an underwater video: Striped Trumpeter and Western Butterfish.

Unlike traditional image retrieval techniques which conduct global shape matching, local descriptors are concerned with partial features. [Benson et al., 2009] implement an automated fish species recognition system by introducing Haar descriptors. The experiment applies a Haar classifier using 83 features in 2547 images and achieves an accuracy of 89%. [Spampinato et al., 2010] classifies 360 images of 10 different species using a spatial Gabor filter as texture features. The result achieves an average accuracy of about 92%. [de Zeeuw et al., 2010] construct a computer-assisted system based on the SIFT descriptor. The experiment is constructed on a 76-image database collected at Juno Beach, Florida (USA). The result has one false negative and four false positives. In the same paper, a 100% correct result is achieved on a database of 151 images from Matura, Trinidad. The perfect result mainly occurs because of the prominent spot on the animals.

3.1.4 Fish special features

Traditionally, marine biologists have employed many tools to examine the appearance and quantities of fish. For example, they cast nets to catch and recognize fish in the ocean. They also dive to observe underwater, using photography in [Caley et al., 1996]. Moreover, they combine net casting with acoustic (sonar) ([Brehmer et al., 2006]). In such cases, the weight and size attributes of fish body or parts (tail, fins, head, *etc.*)

are suitable because these characteristics are easily required and associated with fish species. On the other hand, underwater surveillance techniques demand computer vision expertise integrated with marine knowledge of living organisms so that these features represent the species characteristics other than a general illustration of the image attributes.

[Walther et al., 2004] aim at resolving detection and tracking problems in underwater environments. In order to detect foreground regions, the average background method is used. Constant background features are calculated for each frame and the result subtracted from the current frame. Firstly, they use saliency methods in a selective attention algorithm. This step decomposes input frames into seven channels and computes 42 feature maps from six spatial scales per channel. It also introduces an iterative spatial competition to select robust locations. A psychology model called "inhibition-of-return" is implemented. Secondly, across-orientation normalization is used to exclude marine snow noise. Thirdly, the algorithm extracts the target's outline and centroid for tracking, which is processed with a linear Kalman filter. Finally, detected objects are marked in the video frame. The fish descriptions include major and minor axes, aspect ratio, total area size and maximum and average luminance. Their work concerns saliency maps that are used to minimize multi-agent tracking. They describe a system for tracking marine animals from a remotely operated underwater vehicle. This system is proposed for low-contrast images where targets are translucent in the underwater video. To tackle this issue, several features maps are employed to detect the potential animals. After that, a single saliency map is combined from these maps. Considering fish images in our database, the proposed algorithm needs an adaption due to the large size of our objects and the slow movement of the objects in the image. This paper also proposes an interesting concept: in underwater fish video, the ten most common animals correspond to 60%, and the 25 most common animals to 80% of all observed objects. Instead of a universal recognition system, a biased fish species recognition system would be more effective and accurate.

[Ros-Sánchez et al., 2010] introduced a new architecture to define fish and environmental features. The definition describes the accurate location and an easy-to-read representation based on star charts of fish (shown in Figure 3.4). The chart can be compared to charts of different fish. The proposed method involves two techniques: one is an improved adaptive background model ([Kaewtrakulpong and Bowden, 2001]), and the other is a median operator based locator. By employing Gaussian Mixed Models

and an EM-online algorithm, this method removes environmental noise (vibrations, shadows, reflections, *etc.*). Based on these algorithms, this paper uses a specialized tracking solution. The solution tracks each fish from the videos and provides their position at every second. It allows the quantification of fish activity and proposes a uniform standard for further processing. Based on their proposed method, these quantified data are exported for further analysis and graphical visualization. Although this proposal is clear and robust, the main limitation is that the fish images and attributes are observed in a fish tank. Unlike in a natural environment, in which one would anticipate light reflections, colour changing, *etc.*, the fish's state and activity in the tank is bounded, and can be well described using limited parameters. The authors apply this method to recognize sea animals (zebra fish and gilt-head sea bream), and the result shows this framework is robust both in day mode and night mode. The success rate of day and night mode is 96% and 89.8%, respectively.

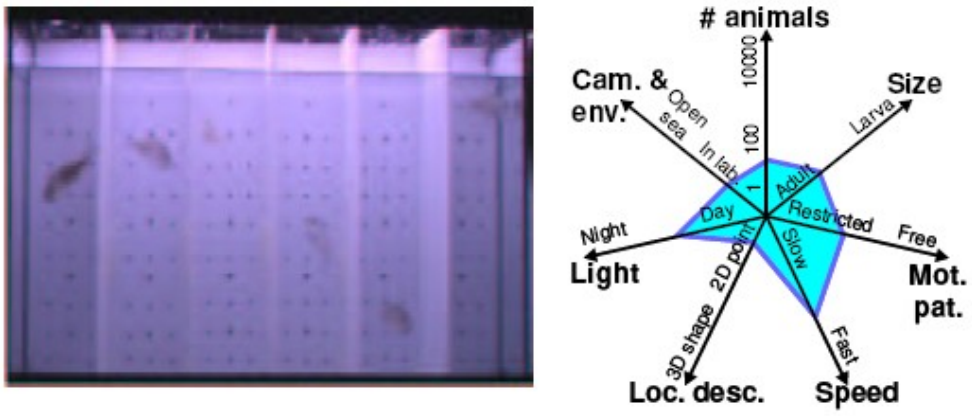


Figure 3.4: Sample frames and associated star charts. 2D location of gilt-head sea bream (GSB) [Ros-Sánchez et al., 2010].

[Spampinato et al., 2008] introduced some special fish features, including Adipose fin, anal fin, caudal fin, head and body shape size and length/depth ratio. These special features describes the distinctive attributes of fish and they need prior biological knowledge to locate, extract and describe. These features play a crucial rule in fish recognition. [Zion et al., 1999] also employs prior fish distribution knowledge and builds a decision-tree. This approach uses two thresholds to determine fish species, while other methods such as template-based approaches use a training set and build the templates. Although fish special features provide a clear and accurate way to recognize fish, this approach has some vital disadvantages. For example, these features need special prior knowledge, which may be difficult to obtain. Furthermore, the selection of fish special

features affects the recognition results. The selection aims at choosing the most distinguishing features between species. Some useful fish special shapes are listed in table 3.2.

Table 3.2: Useful fish special shapes

Component	fin	stripes	spots
Property	rim	fringe	translucency
Fish part	top/bottom	head/tail [Frouzova et al., 2005]	eye/mouth

3.2 Methodology

We acquired underwater videos from the Fish4Knowledge project, which has deployed 9 embedded video cameras to record halobiotic activities (including fish, snakes, insects, *etc.*) at three sites in the Taiwan Sea and stored the videos in a cluster server for three years as the project proceeded. These videos contain discriminative information for analysis. However, the quality of underwater video streaming affects the accuracy of fish recognition by adding distortions and noise to the original images. The motion and diffraction effects smear the fish shape model like applying a convolution to the original image. These factors decrease the video quality and produce classification errors.

Figure 3.5 gives a snapshot of how these factors affect the video quality. More specifically, there are at least four factors that play a primary role. The first factor is the image anamorphosis, which distorts the geometric optics and results in optical aberrations such as shape deformation. This distortion model is irregular and complex because the scattering of light changes the position of halobiotic objects. The effects due to the influences of surface illumination variations also aggravate this problem. The waves experience light fluctuation at the water surface and add an undulant illumination phenomenon onto the video frames. The second factor is colour distortion since the cameras are placed at a depth of five to ten metres. The three primary colours (referring to red, blue, and green) have their own transmission rates in water. This issue makes colour descriptors, one of the most significant features in computer vision, become less reliable. The third factor comes from the motion blur, especially when fish are swimming rapidly across the camera sight. Typically, our cameras record five

Figure 3.5: Original underwater video frame with two fish and coral (acquired from Fish4Knowledge project website).

frames per second. The motion of fish in this 200 ms period triggers blurring. As to the fourth influence, the water impurity also affects the quality of recorded images because it limits the range of observation and changes the physical attributes of the light.

To conclude, the underwater environment affects the video quality via two factors. The first is that transient phenomena produce non-constant degradation. The other one is the distortion of light when it is transmitted in water media. Our approach is to improve the quality of fish detection results and strengthen the feature extraction processing so that it is robust to the four issues discussed above. The whole procedure of feature extraction is illustrated in Figure 3.6. It includes image pre-processing, feature extractions, and feature normalization. These steps are described in the following sections.

3.2.1 Image pre-processing

The pre-processing procedures are undertaken to improve the quality of features. Firstly, the detection and tracking software described in [Nadarajan et al., 2011] is used to obtain the fish and mask images. Then the Grabcut algorithm [Rother et al., 2004] is employed to improve segmentation of the fish objects. Given prior information such

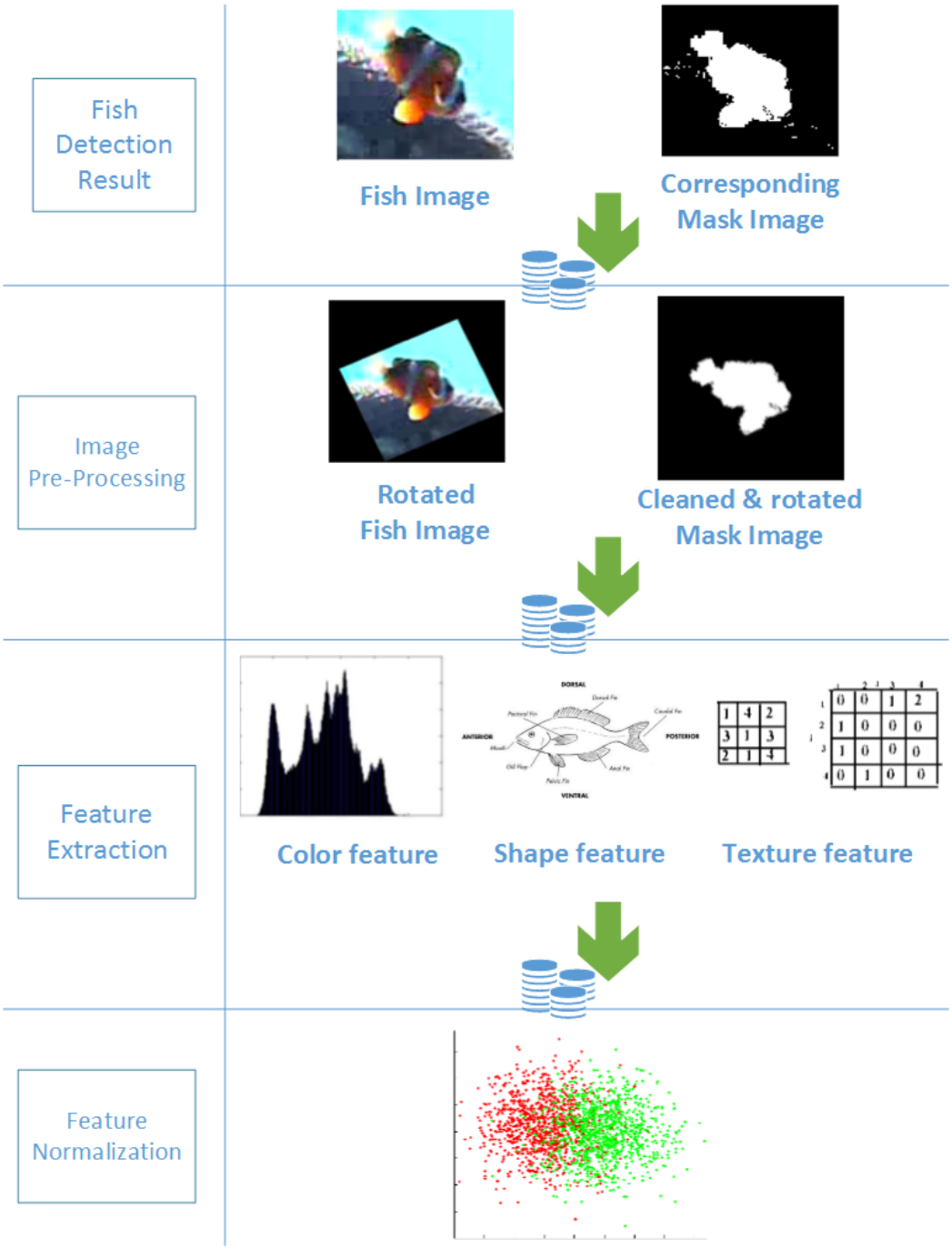


Figure 3.6: Feature extraction workflow. Fish features are extracted using the same sequence of steps as described above.

as the reference frame or pre-labelling of the foreground area, the graph cut solution gives each pixel a weight between the foreground (source) and background (sink) and solves the segmentation problem with a minimum cost cut method to divide the source from the sink. The Grabcut method improves location of the fish in bounding box. We then add padding around the detected fish to ensure that the whole fish is included. The padding may extend outside the input frame if the fish is close to the edge of the frame. An example of a detected fish is provided in Figure 3.7, where most parts of the key feature (white tail) are preserved by the segmentation algorithm.

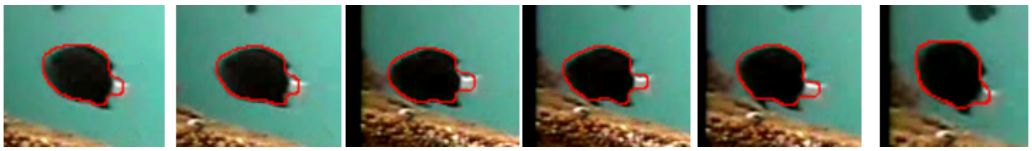


Figure 3.7: An example of fish detections from a whole trajectory of *Chromis margaritifer*. This species of fish has a noteworthy white tail. This feature is essential for discriminating it from other species of fish, especially *Dascyllus reticulatus*. These images have successfully maintained most parts of the white tails.

However, an issue that can be observed here as well is that the detected contour (marked by a red line along the body of the fish) is not exactly coordinated with the actual outline. A piece of the dorsal fin, and a portion of the anal fin, are missing while some background such as water is falsely included as part of the fish. These inaccurate segmentations fabricate a distorted image of the fish outline and lose detailed descriptions when generating the shape features. Thus, we append texture and colour features besides the shape features to produce a more comprehensive and robust set of descriptors. Grabcut has a boundary smoothness energy term that avoids sharp changes of cuts off small features of bridge over small gaps.

After acquiring the fish bounding boxes, we align the fish images in the same direction before further processing. We rotate their bodies by an estimated angle so that fish from the same species are facing the same directions. Thereafter, we can divide the fish into several parts and extracts specific features (*e.g.* focus on the white tail part for *Chromis margaritifer*). The rotation angle is estimated by using a heuristic method inspired by the streamline hypothesis. It assumes that a fish's head is smoother than its tail and fins because it needs a more frictional tail (caudal fin) to swim and keep its body balanced. As a result, the centre position of the curvature value (Formula 3.1) along the fish contour is stable given fish images of the same species. We justify this streamline

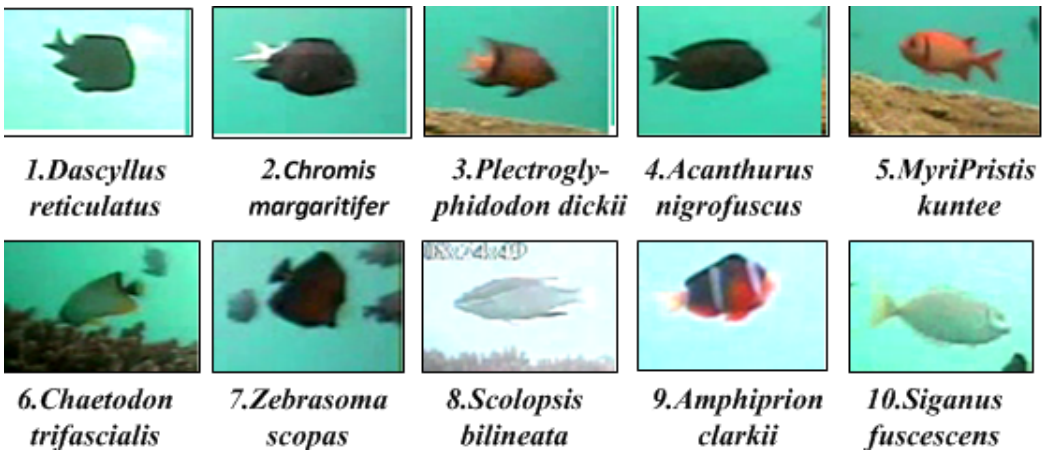


Figure 3.8: Top 10 species of fish in the dataset. This figure describes the various densities of curved points within different fish parts. Normally the fish head is smoother than its rear part since this kind of shape helps reduce resistance from water while swimming. Conversely, fish need a more frictional caudal fin to swim and to keep their bodies balanced.

hypothesis by showing the top 10 species in Figure 3.8. The curvature accumulation, which is the accumulation of the curvature value along the contour pixels, of the top part of the fish (including the dorsal and adipose fins) is neutralized by evaluating the curvature of the bottom (pelvic and anal fins). That being the case, the caudal fin (fish tail) determines the direction of the weighted curvature centre. Some species of fish have smoother tail parts because their tails are hard to detect. For example, the species *Chaetodon trifasciatus* has a more transparent tail which is difficult to be completely detected. In this case, fish from this species tend to be orientated by the algorithm proposed below in the opposite direction since their head shape dominates. Also, we have to flip some fish from left to right so as to keep their back upward.

The following discussion presents the technical details of implementing and evaluating this hypothesis. To do so, we firstly smooth the fish boundary with a Gaussian smoothing to eliminate small noise, and then calculate the curvature value of each boundary pixel as following [Mokhtarian and Suomela, 1998, He and Yung, 2004]:

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2))^{\frac{3}{2}}} \tag{3.1}$$

where $X_u(u,\sigma)/X_{uu}(u,\sigma)$ and $Y_u(u,\sigma)/Y_{uu}(u,\sigma)$ are the first and the second derivative of $X(u,\sigma)$ and $Y(u,\sigma)$, respectively; $X(u,\sigma)$ and $Y(u,\sigma)$ are the convolution result of 1-D Gaussian kernel function $g(u,\sigma)$ with fish boundary coordinates $x(u)$ and $y(u)$. We

fix σ so that κ depends only on u. As the pixel curvature is sensitive to local corners, we normalize it using the logarithm function:

$$\kappa_{normalize} = \begin{cases} log(\kappa) & \text{if } \kappa \geq 1 \\ -log(2-\kappa) & \text{if } \kappa < 1 \end{cases} \tag{3.2}$$

Afterward, we calculate the fish orientation by weighting each contour pixel with its local curvature value. The centre of curvature value is averaged by the coordinates and weighted by the curvature value of these pixels along the fish contour, as following:

$$< x_c, y_c >= \frac{\sum_i \kappa_i * < x_i, y_i >}{\sum_i \kappa_i} \tag{3.3}$$

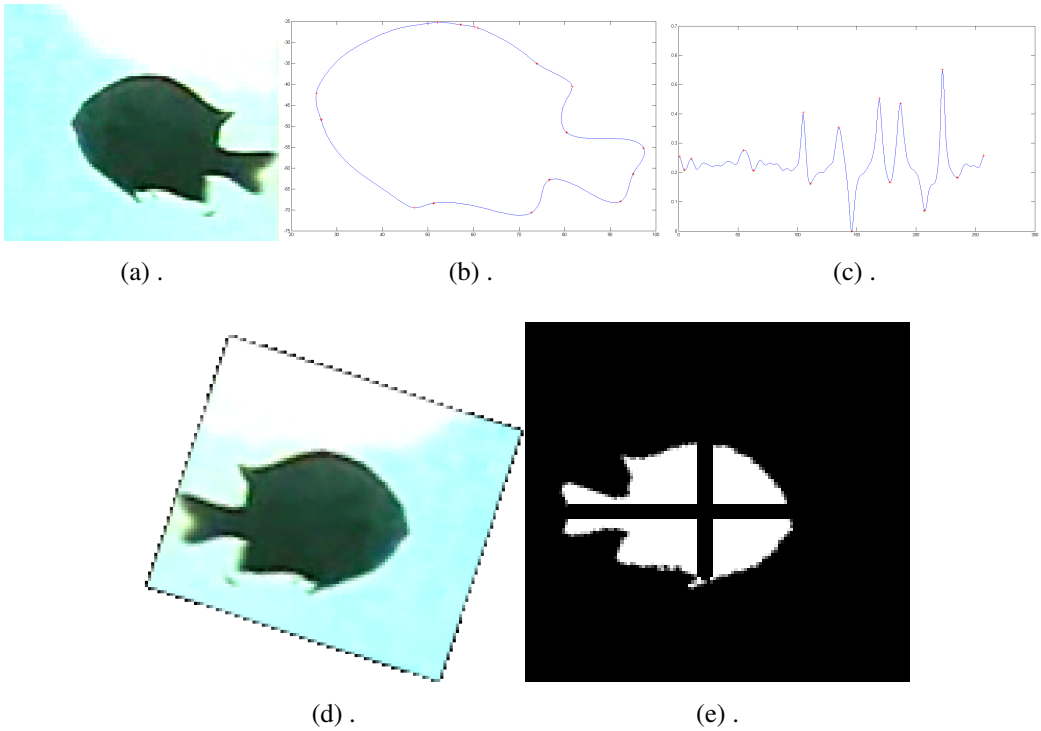
where i is the index of boundary pixel, $< x_i, y_i >$ is its corresponding coordinates and κ_i is the corresponding curvature value. the The calculated centre of curvature value $< x_c, y_c >$ points to an anchored direction from the centre of the fish body. The tail direction is considered to be the roughest part of the fish shape.

A typical fish orientation procedure is illustrated in Figure 3.9. Considering the first image (Figure 3.9a) as input, we first smooth the contour image with a Gaussian filter to eliminate the spines, which generate pulses in curvature and should be excluded since we only care about substantial components (Figure 3.9b). The degrees of curvature of fish contour are illustrated in Figure 3.9c, where the x-axis is the index of pixels of contour starting from the top part of the fish and passing anti-clockwise and the y-axis stands for the curvature degree. The curvature degree fluctuates more severely on the right side than on the left since the curvature is concentrated at the rear half of the fish. In order to refine the estimation of tail direction, we fit the fish boundary into an ellipse shape, and then use the deflective angle for minor trimming. Figure 3.9d shows the final result, where the *Dascyllus reticulatus* is rotated horizontally and faces right. The fish orientation method achieves 95% correct fish orientation $\pm 15^{\circ}$ using 1000 manually labelled fish images.

After orientation, fish images are divided into four parts (head/tail/top/bottom) according to the positions relative to the fish centre for feature extraction.

3.2.2 Feature extraction

The procedure of feature extraction is often considered as a black box in object recognition applications. However, the quality of features is critical in the following clas-



(a) . (b) . (c) .

(d) . (e) .

Figure 3.9: Fish orientation demonstration: (a) input image of *Dascyllus reticulatus* fish; (b) fish boundary after Gaussian smoothing, with small spines eliminated since we are only interested in substantial fluctuations; (c) curvature levels along fish boundary, where the x-axis is the index of pixels of the contour starting from the top part of the fish and counting anti-clockwise, and the y-axis shows the degree of curvature; (d) oriented fish image for further processing. (e) the fish mask image is segmented into four parts: head, tail, top, and bottom. This method helps to divide fish in a constant way and extracts specific features (*e.g.* the white tail of *Chromis margaritifer*).

sification step. In practice, feature engineering work aims at obtaining discriminative characteristics of input data, which are illustrated for causal inference. In this section, we propose a set of feature-engineering methods to extract effective computer vision descriptors for fish. We treat this as an incremental process, where new features are designed and complemented based on the accuracy acquired by existing features. More specifically, we put all existing features into a pool for selection, and the selection algorithm chooses the candidate features which maximize the averaged classification accuracy over all species. We also introduce a set of new features which help distinguish fish species that tend to be misclassified. We categorize all features into four types, as described in the following sections.

3.2.2.1 Normalized colour descriptors and the REHIST method

Colour, as an intuitive kind of feature, is adapted in our fish recognition and regularized so that it is robust to environmental factors. As discussed previously, colour features in underwater video frames suffer from several physical factors such as absorption and scattering, reduced amounts of light, and poor visibility. Instead of employing the raw values of pixels, our methods for colour feature extractions involve colour normalization, using illumination-invariant components and recalculated histograms in order to capture the constant colour idiosyncrasies of fish.

Colour normalization compensates for the illumination variations from camera equipment, scenes, and weather conditions. This method assumes that each pixel value is a fraction of the illumination level (grey scale) applied to the red, green, and blue colour channels. Therefore, value of a pixel that is independent to illumination can be calculated by dividing each channel by the sum of all three channels as described by the following formula:

$$(R', G', B') = \left(\frac{R}{\sum R+G+B}, \frac{G}{\sum R+G+B}, \frac{B}{\sum R+G+B}\right) \tag{3.4}$$

where R,G, and B are the values of three channels. The normalized colour histograms of five detections are illustrated in Figure 3.10. The first column shows detected *Dascyllus reticulatus* fish. Their original colour histograms are shown in the second column, with substantial variations. The normalized colour histograms are more stable because normalization reduces the effect of light variation.

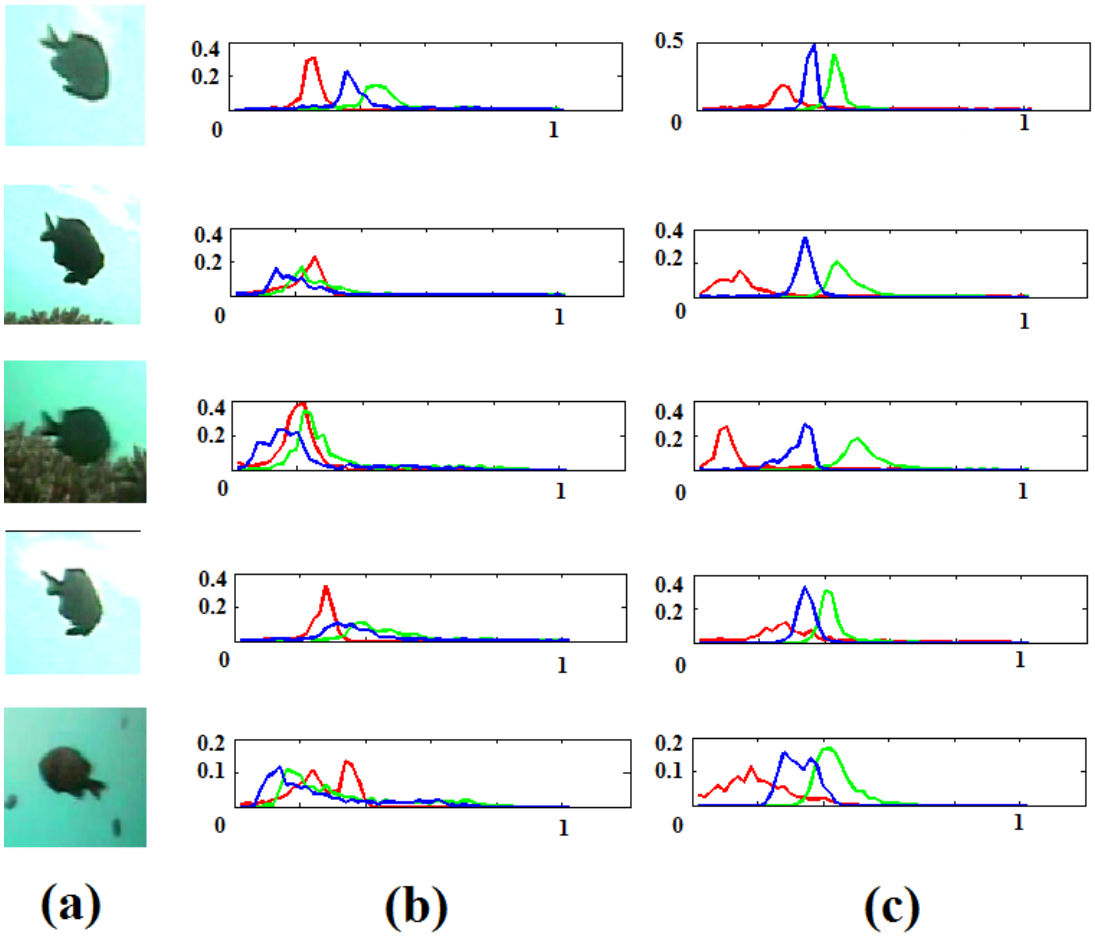


Figure 3.10: Colour histograms and their corresponding normalized colour histograms from five different detections of *Dascyllus reticulatus* fish. (a) Fish detections are in the first column. (b) Various colour distributions in the histogram since the illumination changes due to absorption, scattering, and reduced amounts of light. The histogram only covers the pixels of the detected fish. The background pixels are ignored. (c) Colour distributions after normalization. These histograms are more stable because normalization reduces the effect of light variation.

Another colour property, hue, is also introduced and calculated as a histogram. We do not use other correlates of colour appearance, such as colourfulness, chroma, and saturation, since they are sensitive to circumstantial factors. A global offset is applied in order to adjust the average colour value to the centre of the histogram, since the value of the hue component wraps around. The hue value is calculated from the RGB value by the following formula:

$$Hue = \arctan \frac{\sqrt{(G-B)}}{2*R-G-B} + \theta_{offset} \tag{3.5}$$

where the θ_{offset} is pre-calculated so that the Hue value is mainly located in the middle range. We calculate the histograms of normalized colour and hue components of five fish parts: head, tail, top, bottom, and whole body. Every histogram consists of 51 bins (RGB value from 1 to 255 with the bin-width set to 5). Figure 3.11 shows examples of colour histograms that compare the mean value and standard deviation of two species.

By combining these histograms, we capture the perceptional properties and present their diversities through the variations of density. However, as seen in Figure 3.10, the distributions of histogram bins are not uniform. Their values are concentrated around the major colour elements while some other bins tend to be empty. In order to equalize the colour histogram and create a more uniform distribution for the whole dataset to maximize contrast, we calculate the average distribution of the whole dataset and use it as the global probability function for histogram equalization. More specifically, a global $\overline{B}$ is calculated by averaging the colour histograms of all species.

$$\overline{B_j} = \frac{1}{N} \sum_{n=1}^{N} B_{n,j} \tag{3.6}$$

where $B_{n,j}$ ($j \in \{1,...,51\}$) is the jth original colour histogram bin of sample n, N is the number of samples. Then, we recompute the range of all histogram bins according to the global probability in $\overline{B}$ and map them into an 11-bin histogram to take full advantage of all ranges. Since we only use 11 bins after REHIST, this method also helps reduce the risk introduced by quantization. The whole method is described below:

$$\widetilde{B}_i = \sum_{j=a_i}^{a_{i+1}} B_j \qquad s.t. \qquad a_i = min\{X \in \mathbb{N}^+ \mid \Sigma_{j=1}^{X} \overline{B}_j \geq \frac{i}{11}\} \tag{3.7}$$

where B_j ($j \in \{1,...,51\}$) is the original colour histogram bin, $\overline{B}_j$ ($j \in \{1,...,51\}$) is the averaged histogram over all samples and $\widetilde{B}_i$ ($i \in \{1,...,11\}$) is the recomputed bin.

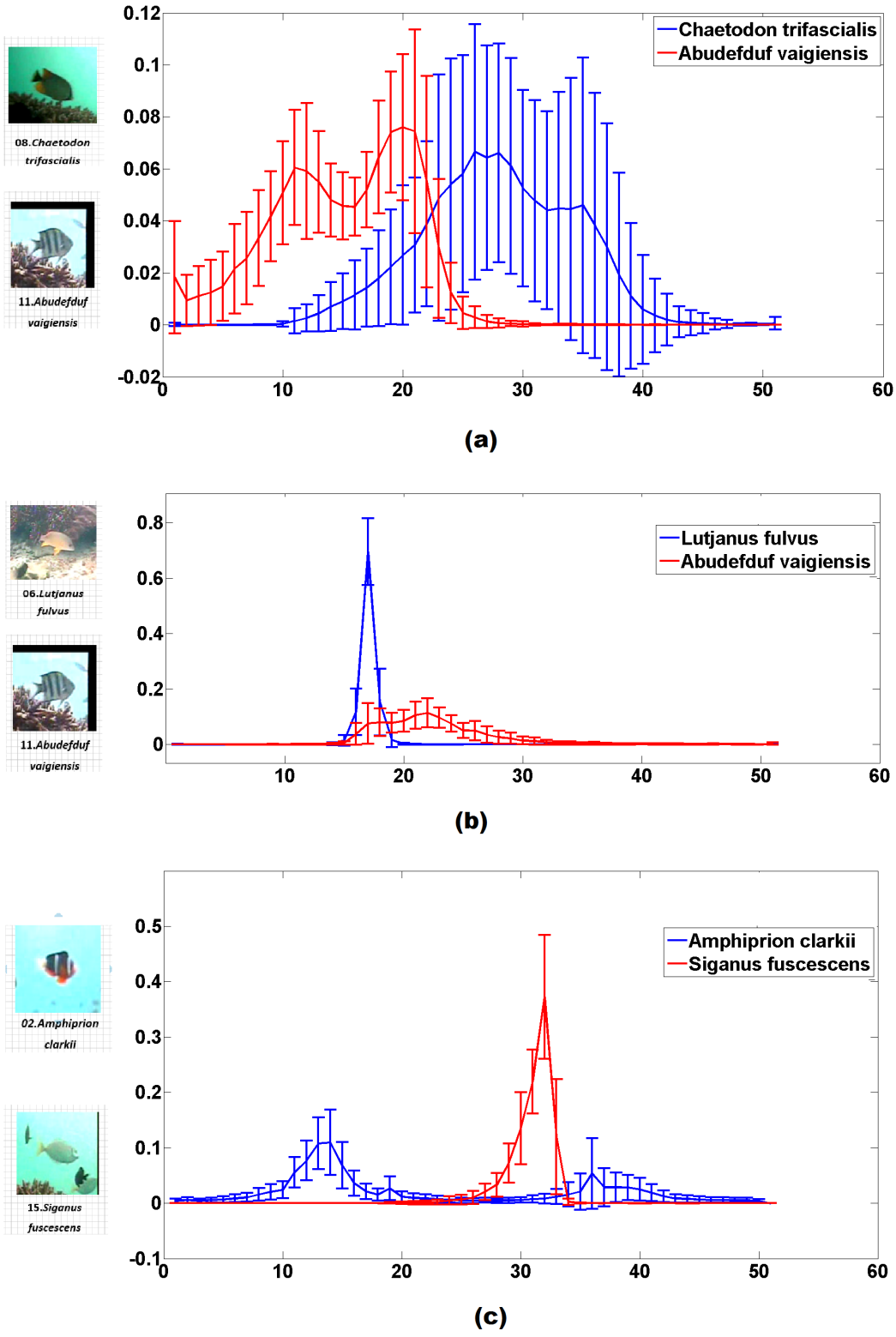


Figure 3.11: Example histograms of colour values and the comparisons of the mean value and standard deviation from two species. (a) Normalized Red colour histogram. (b) Normalized Green colour histogram. (c) H histogram in HSV colour model.

Once the $\overline{B}_j$ are generated, we recompute the histograms of each fish individually. Figure 3.12 shows examples of REHIST colour histograms that compare the mean value and standard deviation of two species.

3.2.2.2 Combined shape descriptors from fish contours

With the same notation in Section 3.1, shape features are provided by the segmented contour of the detected fish. The curvature value varies along the contour and this trend is captured and presented as shape features. These features will describe the deformations of patch bounder and are useful for geometric matching. This requires a reliable segmentation from the background, and the boundary descriptors can be used to describe the species variations. Unlike the simple experimental environments found in the majority of previous works, we have to deal with a more complex situation with visual noise and distortions. Thus, we use statistical attributes such as Moment Invariants (MIs) and Fourier Descriptors (FDs) that are abstracted from fish contours as the shape features.

The first type of shape feature is represented by the FDs [Zahn and Roskies, 1972] since it is a representation of 2D points that is independent of variations in location, rotation, and scaling. Considering the coordinates of the t-th points x_t and y_t, the FD method utilizes all complex pairs $x_t + i * y_t$ to generate the coefficients by a discrete Fourier transformation:

$$FD(k) = DFT(x_t, y_t) = \frac{1}{N} \sum_{m=0}^{N-1} (x_t + i * y_t) * e^{-i2\pi tk/N} \tag{3.8}$$

where N is the number of input pixels, $k \in 0, ..., N-1$ is the index of coefficients in the frequency domain. We first compute the local maxima of a low scale curvature for each contour and then eliminate the rounded corners and false corners, as introduced by [He and Yung, 2004]. All remaining corner points are input into a Fourier transformation, and we utilize the first 15 coefficients of the frequency domain as the feature group. Then these coefficients are normalized by dividing the sum of histogram.

MIs are designed to represent the region's properties regardless of its translation, rotation, or scale. We use the related independent basis set of MIs presented by [Flusser et al., 2009]. These MIs are combined and rescaled results of the complex central moments $C_{u,v}$, as

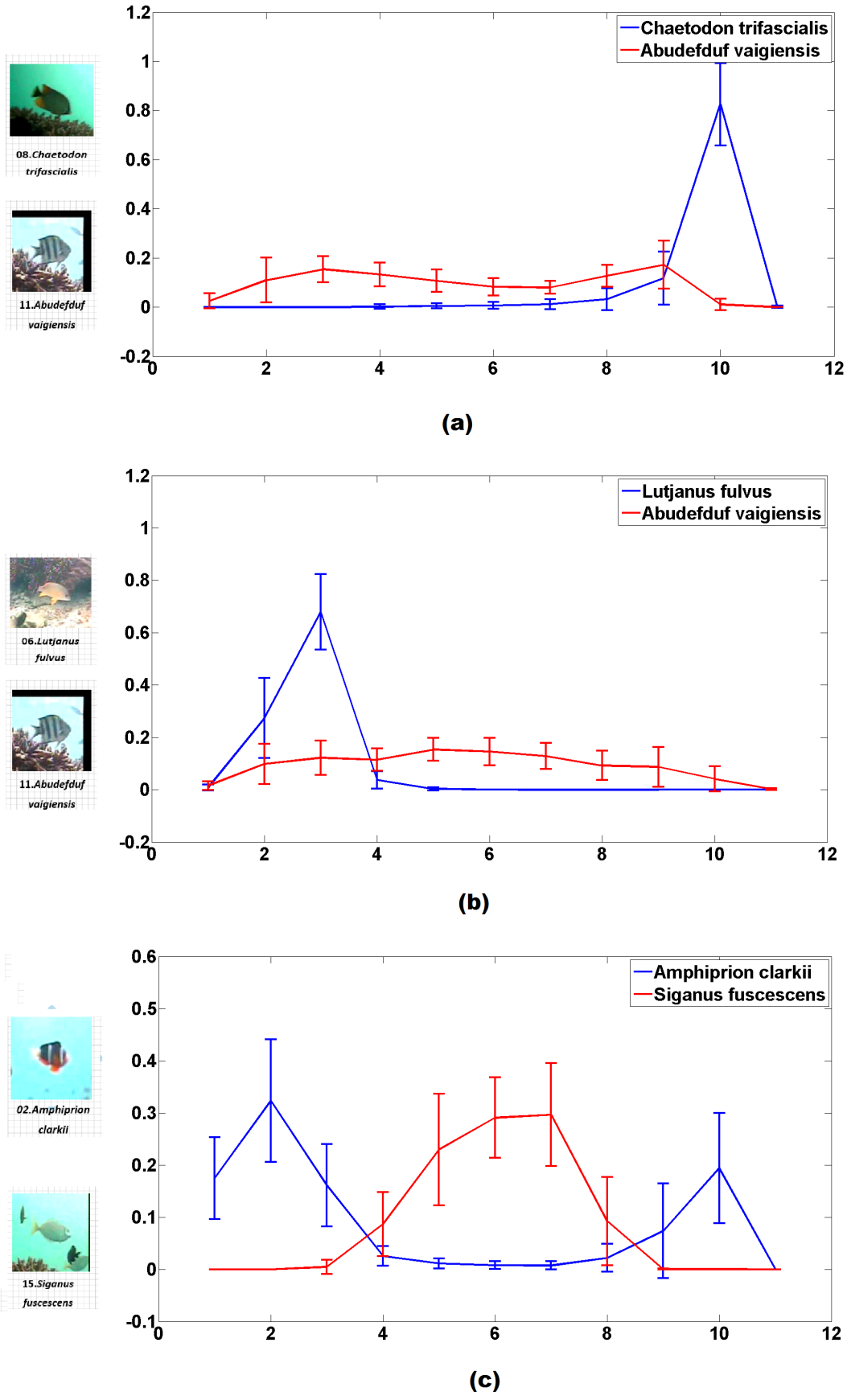


Figure 3.12: Example histograms of REHIST colour values and the comparisons of the mean value and standard deviation from two species. (a) Normalized Red colour histogram. (b) Normalized Green colour histogram. (c) H histogram in HSV colour model.

Moment Invariants	definition
ϕ_1	$C_{1,1}$
ϕ_2	$C_{2,1} * C_{1,2}$
ϕ_3	$Re(C_{2,0} * C_{1,2}^2)$
ϕ_4	$Im(C_{2,0} * C_{1,2}^2)$
ϕ_5	$Re(C_{3,0} * C_{1,2}^3)$
ϕ_6	$Im(C_{3,0} * C_{1,2}^3)$

Table 3.3: The definition of the first 6 Moment Invariants (MI).

defined below:

$$C_{u,v} = \sum_r \sum_c ((r-\tilde{r}) + i*(c-\tilde{c}))^u * ((r-\tilde{r}) - i*(c-\tilde{c}))^v * f_{rc} \tag{3.9}$$

where ($\tilde{r}$, $\tilde{c}$) is the centre of a connected region, f_{rc} is the image pixels with the foreground = 1. Six MIs, which are generated from the complex area moments and have a rank of up to three, are computed on each patch of fish and grouped together. These MIs are defined in Table 3.3.

The MI features encode the 2D shape transformations, and they are useful only when the fish keeps their body pose parallel to the camera surface. However, the recorded video frame is the result of projecting the 3D fish body onto the camera surface, and it is affected by affine distortions. Thus, we use Affine Moment Invariants (AMIs) [Flusser et al., 2009]. Firstly, affine transformation can be expressed as:

$$\begin{aligned} u &= a_0 + a_1 * x + a_2 * y \\ v &= b_0 + b_1 * x + b_2 * y \end{aligned} \tag{3.10}$$

where (x, y) and (u, v) are coordinates in the image before and after the transformation. The AMIs are designed so that they are independent of the affine transformations. More specifically, this method investigates the ratios $u_{pq}/u_{00}^{(p+q+2)/2}$ that are invariant to translation and scaling, where p and q are the orders, u_{pq} are the central moments as defined below:

$$u_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-x_t)^p * (y-y_t)^q f(x,y) dx dy \tag{3.11}$$

where $p,q = 0,1,2,...,x_t,y_t$ are coordinates of the centroid, x,y are over all pixels in the shape, $f(x,y)$ is the image pixel value. These AMIs are chosen in a graphical method that integrates the product of the triangular areas over the object by first representing the invariants as a graph and then removing the reducible invariants as presented in [Flusser et al., 2009]. We use nine AMIs in each part of a fish with a weight of up to 12. Since the fish mouth and tail contain more unique information, we have also applied the AMIs to the first half part of the fish head and tail shape image.

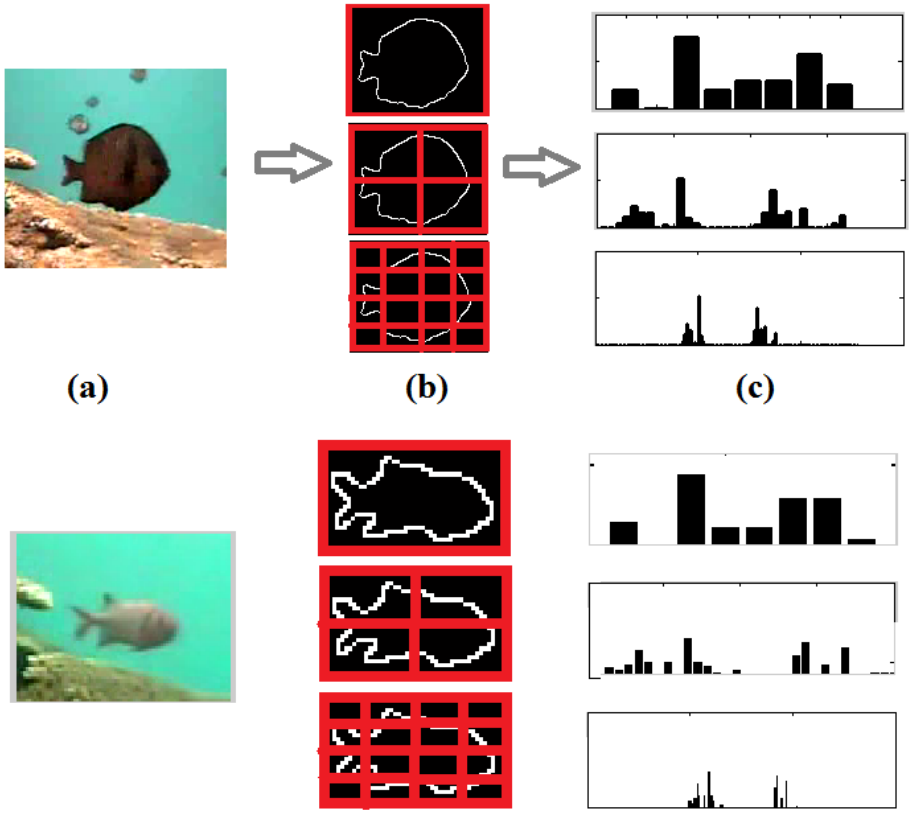


Figure 3.13: Example of Pyramid Histogram of Oriented Gradients. This method counts the gradient directions of fish contours and arranges them into the bins to which they belong. The pyramid descriptors illustrate more details at deeper levels while they are more robust to noise in the top layer.

As well as calculating machine vision attributes that illustrate the geometric properties of fish contours directly, we exploit the Pyramid of Histograms of Orientation Gradients (PHOG) [Bosch et al., 2007] to divide the fish contours into several levels with a pyramid spatial structure and obtain the statistical results of the gradient orientations of each area. An example of a PHOG feature vector is demonstrated in Figure 3.13. It denotes the spatial histogram representation of gradient directions in a pyramid struc-

ture of layers, and the final PHOG vector is a weighted combination of all levels. In our work, we use a four-layer pyramid (Figure 3.13 shows a three-layer example), and concatenate each normalized histogram as a group of features for selection for further processing.

Figure 3.14 shows examples of boundary feature that comparing the mean value and standard deviation of two species.

3.2.2.3 Fish texture analysis and feature extraction

Texture features create the summaries of the intensity arrangements (*e.g.* energy measures of primitive pixels, statistical attributes) within the image. The representation of fish texture is obtained by subtracting the background scene and extend the border in every direction of the fish bounding box to make sure the whole fish is covered. The texture features are more reliable than both colour and shape features as they are the features most commonly selected by the feature selection procedure in our experiment.

We calculate the co-occurrence matrix of the fish intensity values and compute the properties of a normalized grey-level co-occurrence matrix (GLCM) [Haralick et al., 1973]. The GLCM describes the co-occurrence frequency of two grey scale pixels at a given distance d:

$$C_{\Delta u,\Delta v}(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases} 1, & \text{if } I(p,q)=i \text{ and } \\ & I(p+\Delta u, q+\Delta v)=j \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

The frequency is calculated for four angles ϕ: 0°, 45°, 90°, and 135°. The offset distance ranges from 1 to 10. We applied Formula 3.12 to the RGB image as a generalization of the GLCM to the multi-spectral image and produced inter-plane combinations of the co-occurrence matrix where six combinations (RR, RG, RB, GG, GB, and BB) are concatenated. We compute 12 features of each normalized GLCM introduced by [Soh and Tsatsoulis, 1999, Haralick et al., 1973] contrast, correlation, energy, entropy, homogeneity, variance, inverse difference moment, cluster shade, cluster prominence, maximum probability, auto-correlation, and dissimilarity, as summarized in Table 3.4.

We use Gabor filters to extract another texture representation and discrimination of various orientations and scales, as described in [Fogel and Sagi, 1989]. In the 2D spatial domain, the Gabor filter is Gaussian-modulated by a complex sinusoid with the

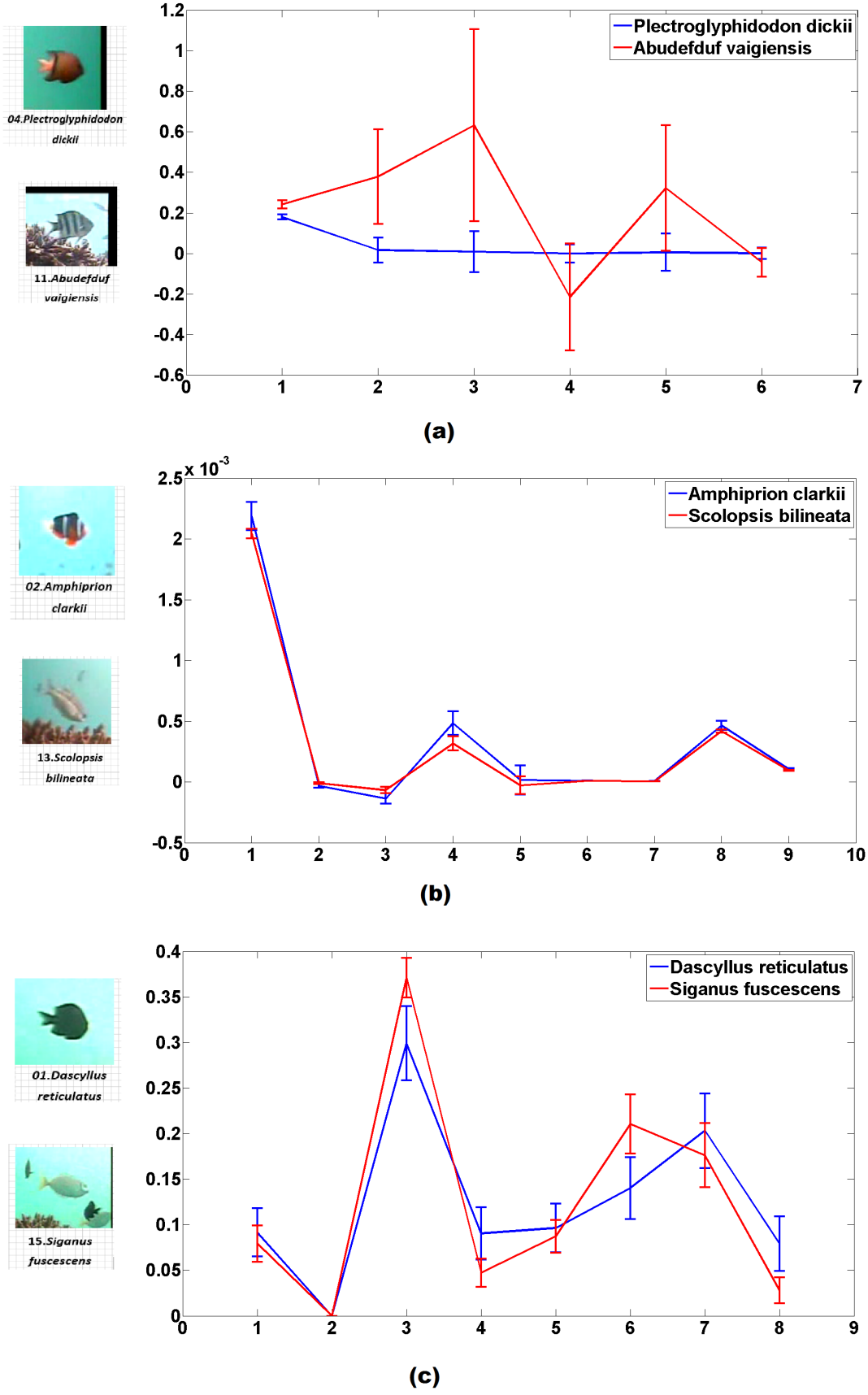


Figure 3.14: Example of two species, and the mean value and standard deviation of the boundary feature. (a) Moment Invariants (MI) of whole body. (b) Affine Moment Invariants (AMI) of whole body. (c) Histogram of oriented gradients, level 0.

feature	Formula
Contrast	$\sum_{i,j} P_{i,j}(i-j)^2$
Correlation	$\sum_{i,j} \frac{(i-u_i)(j-u_j))*P_{i,j}}{\sigma^2}$
Energy	$\sum_{i,j} P_{i,j}^2$
Entropy	$\sum_{i,j} -P_{i,j} \ln P_{i,j}$
Homogeneity	$\sum_{i,j} \frac{P_{i,j}}{(i-j)^2}$
Variance	$\sum_{i,j} P_{i,j}(i-u_i)^2$
Inverse Difference Moment	$\sum_{i,j} \frac{P_{i,j}}{1+(i-j)^2}$
Cluster Shade	$\sum_{i,j} P_{i,j}((i-u_i)+(j-u_j))^3$
Cluster Prominence	$\sum_{i,j} P_{i,j}((i-u_i)+(j-u_j))^4$
Max Probability	$\max P_{i,j}$
Auto correlation	$\sum_{i,j} P_{i,j}(ij)$
Dissimilarity	$\sum_{i,j} P_{i,j}$i-j

Table 3.4: GLCM features. $P_{i,j}$ are the normalized GLCM values (so as to make the whole matrix sums up to 1), u and σ are the mean and standard deviation of the marginal propensity obtained by summing the rows of GLCM.

following equation:

$$
\begin{aligned}
G(x,y,\theta,f) &= exp(-\frac{1}{2}(\frac{x'}{sx'})^2+(\frac{y'}{sy'})^2)*cos(2\pi*f*x') \\
x' &= x*cos(\theta)+y*sin(\theta) \\
y' &= y*cos(\theta)-x*sin(\theta)
\end{aligned}
\tag{3.13}
$$

where Sx and Sy are the variances along x and y-axes, f is the frequency of the sinusoidal function and θ is the orientation of Gabor filter. We use four scales (2, 4, 6, 8), and four orientations (0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$) to produce the output filtered output image as activation values. Similar to the GLCM method, we discard the original Gabor values since they are sensitive to noise and environmental changes, and only the statistical attributes are preserved. The distribution of activations is then counted as a 1D histogram, and the mean value and standard deviation of the histogram are maintained, which evaluate the magnitude of Gabor activations at different directions and scales. More specifically, given the Gabor filter results $F(\theta,f) = I(x,y) \bigotimes G(x,y,\theta,f)$, we compute the histogram $H(\theta,f)$ (range from 0 to 255) from $F(\theta,f)$. Then we calculate the mean value $mean(\theta,f)$ and standard deviation $std(\theta,f)$ of $H(\theta,f)$ for each θ and f. Finally, all of the *mean* and *std* values are combined together as the Gabor features.

Figure 3.15 shows examples of texture features that compare the mean value and standard deviation of two species.

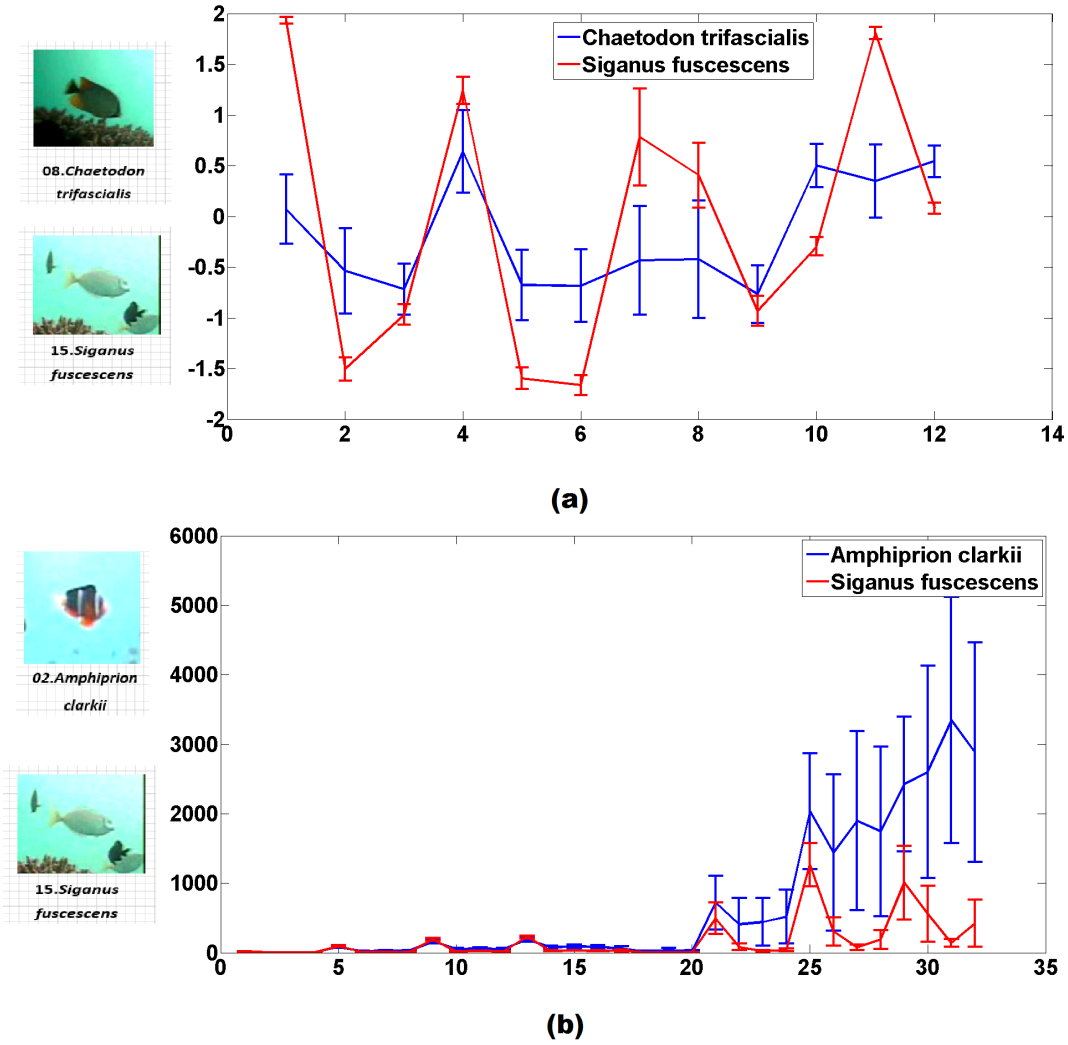


Figure 3.15: Example of two species, and the mean value and standard deviation of the texture features. (a) Attributes of the GLCM, Distance = 9. The histogram attributes are Contrast, Correlation, Energy, Entropy, Homogeneity, Inverse Different Moment, Cluster Shade, Cluster Prominence, Max Probability, Autocorrelation, Dissimilarity, Variance. (b) The attribute histogram of Gabor features of the whole fish body.

3.2.2.4 Idiosyncratic fish features

As well as the generic machine vision descriptors introduced above, some specific features like projected colour density, tail/head and tail/body area ratios, and so on are

included. These features are designed to integrate computer vision techniques with marine knowledge. Marine biologists investigate idiosyncratic features of fish and organize this taxonomic information hierarchically into categories known as the Kingdom, Phylum, Class, Order, Family, Genus, and Species, so those fish that have the same ancestors share similar synapomorphic characteristics. They indicate the distinction between species, for example, the presence or absence of components, specific number, and so on. Some of these synapomorphic characteristics can be obtained from the video frame, mostly from the shape of the fish contour. Firstly, we exploit the projected colour density, which describes the colour variations of fish body changes in both horizontal and vertical directions and generates a density histogram by calculating the mean value of colour along the axis. This feature is useful for describing the significant surface marks such as the colourful tail, stripes, and spots of fish. The mean and standard deviation of the projected density are stored as idiosyncratic fish features.

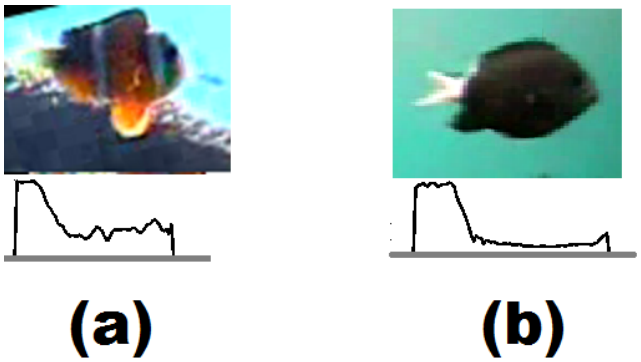


Figure 3.16: An example of the vertical projection of a grey-scale density comparison is presented between the *Amphiprion Clarkii* fish (a) and another species *Chromis margaritifer* (b). The first row is the fish image, and the vertical projected grey-scale density is shown in the row below. In (a), two stripes on the clown fish are captured and reflected in the variants of density. The wide peaks in both (a) and (b) present their white tails.

For example, some species (*e.g. Amphiprion Clarkii*) have stripes so that we can apply the vertical projection of grey-scale metrics to extract their characteristics. This method is illustrated in Figure 3.16, where an example of the vertical projection of the colour density comparison between the *Amphiprion Clarkii* fish and another species *Chromis margaritifer*, which has a white tail, is presented. We could observe from the figure that the projected density describes the vertical changes of fish colour. Two stripes on the clown fish are captured and reflected in the variants of density while both

of the white tails are shown as a wide peak. Similarly, the area ratio of the fish head and tail to the whole body is calculated to represent the geometric structure of the fish and to distinguish them by their relative part sizes. Variations of the fish body are also reflected by the shape of the fish tail. We calculate the mean curvature ratio of the fish tail to its body and use this as the last type of idiosyncratic feature, defined in Equation 3.14.

$$Ratio = \frac{\sum_{u_t \in C_t} \kappa(u_t)}{\sum_{u_w \in C_w} \kappa(u_w)} \tag{3.14}$$

where $\kappa(u)$ is curvature value, as defined in Equation 3.1, C_t is the pixel set of the fish tail (half of the rear part), C_w is the pixel set of whole fish. Given the rotated images that point the fish to the right, we set the left 1/4 part of fish mask image as the fish tail, which is estimated from the manually labelled ground-truth images. We use this coarse estimation because the tail contour from the fish detection is not always stable.

In general, fish from the *Pomacentridae* family have a wider and flatter tail whereas the tail of fish from the *Acanthuridae* family is sharper and more triangular. The averaged curvature is calculated and divided by the average curvature of the fish contour to eliminate the global drift factors. However, the feature quality is constrained by the accuracy of the segmentation algorithm. It is a good strategy to integrate the shape features with other types of features which produce a more reliable combination of features.

Figure 3.17 shows example of idiosyncratic fish features from two species where the feature values are drawn in 2D space.

3.2.2.5 Conclusion

In this chapter, we have discussed the application of feature techniques from computer vision for the recognition of live fish. This is a challenging task due to the low quality of images, light distortions, blurriness, varying range/orientation and diverse backgrounds. We introduced several types of descriptors that are found to be effective and invariant to environmental changes. They are designed to integrate domain knowledge with machine vision methods and considered together as a pool for feature selection in the classification step. This pool is incrementally constructed so that additional features are designed and introduced after analysing the experimental results. As discussed in the beginning, we propose 69 groups of features (2626 dimensions), shown in Table 3.5, to recognize fish. These features are a combination of the colour, shape,

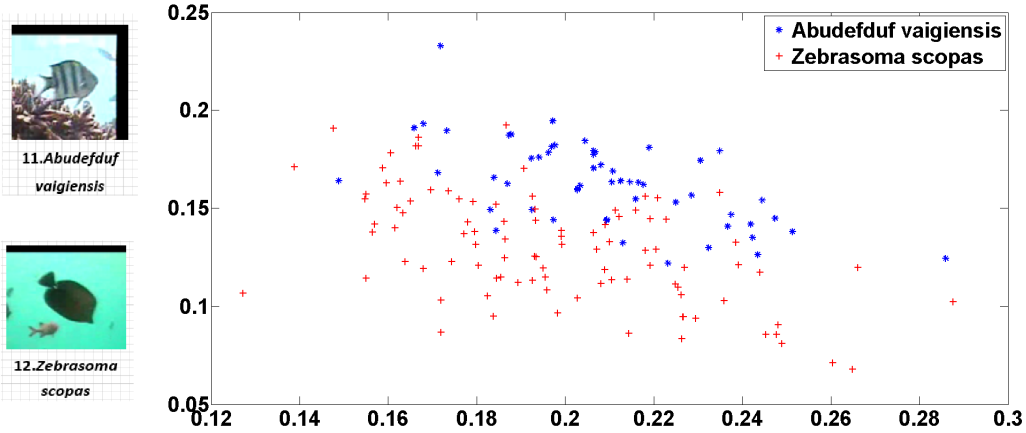


Figure 3.17: Example of idiosyncratic fish features from two species that the feature values are drawn in 2D. X axis is the area ratio from half of the fish head to the whole body., Y axis is the area ratio from half of the fish tail to the whole body.

and texture properties of different parts of the fish such as the tail/head/top/bottom as well as the whole fish. All features are normalized by subtracting the mean and dividing by the standard deviation (z-score normalized after 5% outlier removal).

index	size	Name	Property	Section
1	51	Norm. Red hist	Head	3.2.2.1
2	51	Norm. Red hist	Tail	
3	51	Norm. Red hist	Top	
4	51	Norm. Red hist	Bottom	
5	51	Norm. Red hist	Whole	
6	51	Norm. Green hist	Head	3.2.2.1
7	51	Norm. Green hist	Tail	
8	51	Norm. Green hist	Top	
9	51	Norm. Green hist	Bottom	
10	51	Norm. Green hist	Whole	
11	51	H hist in HSV	Head	3.2.2.1
12	51	H hist in HSV	Tail	
13	51	H hist in HSV	Top	
14	51	H hist in HSV	Bottom	
15	51	H hist in HSV	Whole	
16	11	Norm. Red hist (REHIST)	Head	3.2.2.1
17	11	Norm. Red hist (REHIST)	Tail	
18	11	Norm. Red hist (REHIST)	Top	
19	11	Norm. Red hist (REHIST)	Bottom	
20	11	Norm. Red hist (REHIST)	Whole	
21	11	Norm. Green hist (REHIST)	Head	3.2.2.1
22	11	Norm. Green hist (REHIST)	Tail	
23	11	Norm. Green hist (REHIST)	Top	
24	11	Norm. Green hist (REHIST)	Bottom	
25	11	Norm. Green hist (REHIST)	Whole	
26	11	H hist in HSV (REHIST)	Head	3.2.2.1
27	11	H hist in HSV (REHIST)	Tail	
28	11	H hist in HSV (REHIST)	Top	
29	11	H hist in HSV (REHIST)	Bottom	
30	11	H hist in HSV (REHIST)	Whole	
31	15	Fourier Descriptor		3.2.2.2
32	6	Moment Invariants	Head	3.2.2.2
33	6	Moment Invariants	Tail	
34	6	Moment Invariants	Top	
35	6	Moment Invariants	Bottom	
36	6	Moment Invariants	Half head	
37	6	Moment Invariants	Half tail	
38	6	Moment Invariants	Whole	

index	size	Name	Property	Section
39	9	Affine Moment Invariants	Whole	3.2.2.2
40	9	Affine Moment Invariants	Tail	
41	9	Affine Moment Invariants	Top	
42	9	Affine Moment Invariants	Bottom	
43	9	Affine Moment Invariants	Head	
44	9	Affine Moment Invariants	Half head	
45	9	Affine Moment Invariants	Half tail	
46	8	Histogram of oriented gradients	Level 0	3.2.2.2
47	32	Histogram of oriented gradients	Level 1	
48	128	Histogram of oriented gradients	Level 2	
49	512	Histogram of oriented gradients	Level 3	
50	72	Co-occurrence matrix	D=1	3.2.2.3
51	72	Co-occurrence matrix	D=2	
52	72	Co-occurrence matrix	D=3	
53	72	Co-occurrence matrix	D=4	
54	72	Co-occurrence matrix	D=5	
55	72	Co-occurrence matrix	D=6	
56	72	Co-occurrence matrix	D=7	
57	72	Co-occurrence matrix	D=8	
58	72	Co-occurrence matrix	D=9	
59	72	Co-occurrence matrix	D=10	
60	32	Gabor Filter	Head	3.2.2.3
61	32	Gabor Filter	Tail	
62	32	Gabor Filter	Top	
63	32	Gabor Filter	Bottom	
64	32	Gabor Filter	Whole	
65	1	Half head area ratio		3.2.2.4
66	1	Half tail area ratio		
67	1	Curvature degree ratio		
68	1	Area ratio of fish tail		
69	12	Row/Col Density	RGB	

Table 3.5: Table of the 69 families, includes the index, the number of values in each family, and link to the section describes that family.

Chapter 4

Balance guaranteed optimized tree for live fish recognition

The Balance Guaranteed Optimized Tree (BGOT) is based on the inter-class similarity among fish species, and it groups similar classes at the upper levels of the tree to distinguish them at a later stage. BGOT is a recursive hierarchical structure using a multiclass decision (here using SVM) at each tree node. The feature selection method chooses particular subsets of features to maximize the accuracy over all subsets at each node. Discussion of multiclass classifiers is presented in this chapter, which compares the normal flat classifier approach to the hierarchical classification method. The latter method uses a divide and conquer tactic, and organizes candidate classes into multiple levels. In a greatly imbalanced dataset, the minority classes are grouped with other classes and this strategy helps ease the imbalance of data. The hierarchical classification method also exploits the correlations between classes and finds similar groupings. Unlike biological hierarchical classification methods like the taxonomy tree, which aims to systematize animals into their pre-defined hierarchical categories, the BGOT method chooses an optimal binary split of the given classes at every node. It improves the normal hierarchical method by arranging more accurate classifications at a higher level and keeping the hierarchical tree balanced.

Following the introduction and discussion of the proposed BGOT fish recognition system, this chapter presents a detailed technical description of the BGOT method, including two heuristics for how to organize a single classifier and construct a hierarchical

tree with higher accuracy, and a schematic of the program flow for constructing the hierarchical tree. The foundation of the proposed BGOT algorithm is a multiclass classifier with an optimized feature subset chosen by a forward sequential feature selection. A hybrid set of selected features is essential because a single type of feature is not adequate to describe all fish images, while the whole set of features are neither efficient nor effective. The procedure of choosing the best split was to exhaustively search for all of the possible combinations which is time-consuming and not affordable when the class number grows to be large. We investigated how to deploy the training process on a distributed cluster for heavy computing tasks. We assigned each combination of class set splits to a distributed parallel task. Each pair of class splits is then evaluated to obtain an accuracy score in parallel. Because the binary splits and node classifiers are formed dynamically depending upon groupings, the composing classes of a group are considered as synapomorphies. This means that the constructed hierarchical tree abstracts a connected architecture from a topological graph where the nodes are the class set and the edges reflect the similarities of each pair of two classes. As a result, an integrated hierarchical tree with optimized multiclass classifier is implemented.

We investigate the recognition task of more fish species in a more complex and fundamentally challenging natural environment where the fish are freely swimming. We use underwater cameras to record and recognize fish, where the fish can move freely and the illumination levels change frequently both locally from caustics arisen from the ocean surface waves and globally due to the sun and cloud positions [Toh et al., 2009]. In general, fish recognition is an application of multi-class classification. A common multi-class classifier could be considered as a flat classifier because it classifies all classes at the same time [Carlos and Alex, 2010]. A critical drawback is that it does not consider certain similarities among classes. These classes can be better separated by specifically selected features. One solution is to integrate domain knowledge and construct a tree to organize the classes hierarchically [Deng et al., 2010], called hierarchical classification. This method has significant advantages by grouping similar classes into certain subsets and selecting specific subsets of features to distinguish them at a later stage [Gordon, 1987]. In this chapter, we propose a novel hierarchical classification method suited for greatly unbalanced classes, called the Balance-Guaranteed Optimized Tree (BGOT). To our knowledge, this is the first application of the hierarchical classification method to free swimming fish. It is introduced to process the fish samples from an imbalanced dataset of low quality videos. This system assists

ecological surveillance research, *e.g.* fish population statistics in the open sea. Unlike the biological hierarchical classification method like taxonomy tree, which aims to systematize animals into their pre-defined hierarchical categories by identifying its synapomorphies properties, the BGOT method chooses an optimal binary split of the given classes at every node. It is automatically constructed based on inter-class similarities. It improves the normal hierarchical method by arranging more accurate classifications at a higher level and keeping the hierarchical tree balanced.

The rest of the chapter is organized as follows: Section 4.1 discusses some related work on hierarchical multiclass classification. In Section 4.2, we describe the proposed live fish recognition system and present the algorithm for constructing the hierarchical classification tree. As each fish appears in multiple frames from a video shot, we apply trajectory analysis (Section 4.2.4) to exploit the benefit of multiple views. In the experimental section (Section 4.3), we evaluate the BGOT method on a dataset of live fish images, where all images are manually labelled based on instructions from marine biologists. The experimental results, comparison to the flat SVM and other hierarchical classifiers, and some analysis are presented. The conclusions are drawn in Section 4.4.

4.1 Hierarchical classification method

The task of fish recognition is an application of multi-class classification, which has become an important and interesting research area since the influence of machine learning theory. Over the last decade, SVM [Chih-Chung and Chih-Jen, 2011] has shown impressive accuracy on the multi-class classification task because of its maximum-margin advantages.

Assume training set $\mathcal{D}$ from p classes, which is a set of n sample points of the form:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{1, ..., p\}\}_{i=1}^{n} \tag{4.1}$$

where y_i indicates the class label of m-dimensional vector $\mathbf{x}_i$. Considering the two-class task (p = 2), the Support Vector Machine (SVM) [Cortes and Vapnik, 1995] is optimized to find a hyperplane, called maximum-margin hyperplane, which maximizes the margin between the two classes. A soft margin method is developed to resolve the problem when there is no hyperplane that could perfectly separate the training

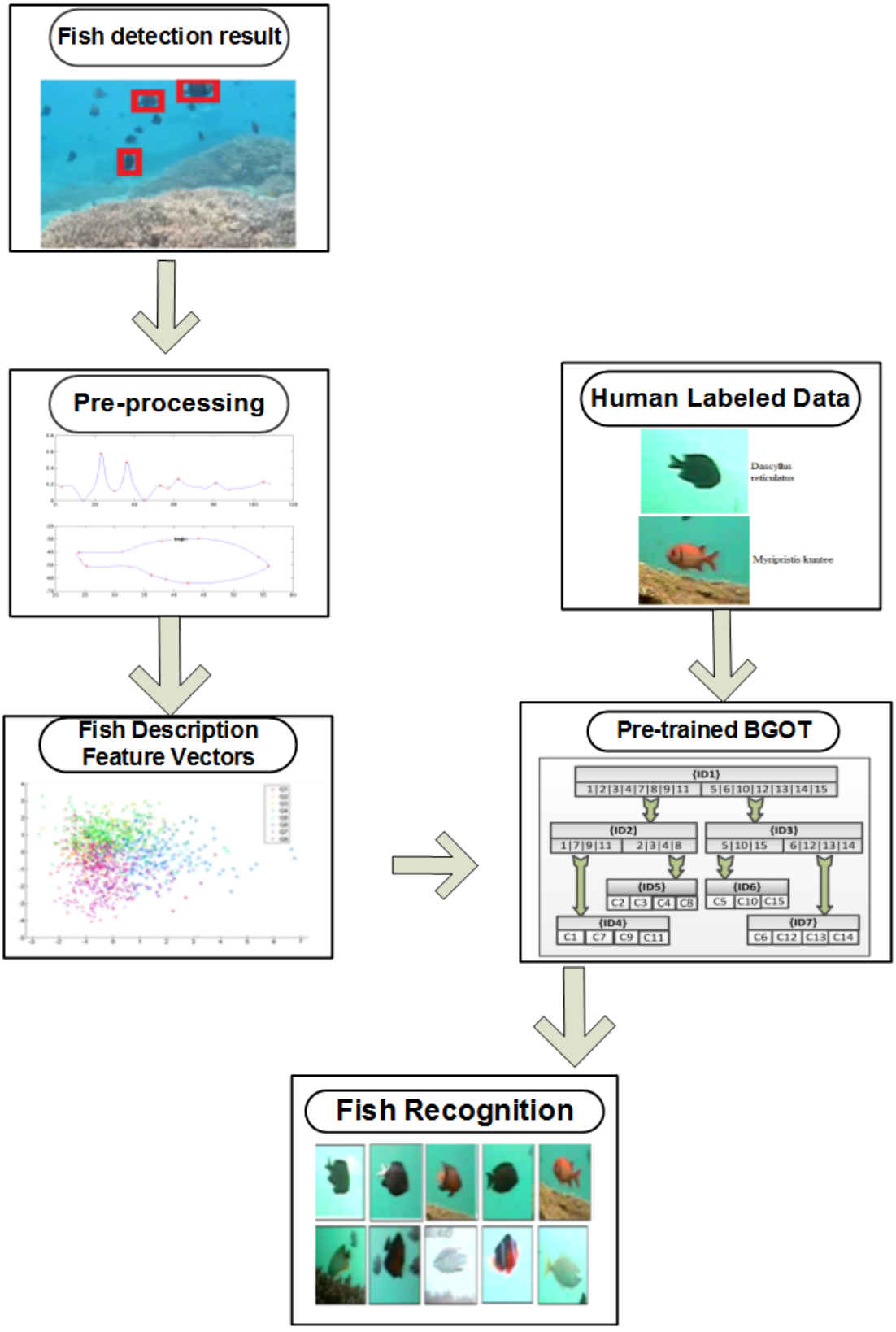


Figure 4.1: The framework of our BGOT-based hierarchical classification system. The workflow shows the training and the recognition procedure. The pre-processing and feature extraction methods are presented in the previous chapter. Section 4.2 describes the proposed live fish recognition system. Section 4.3 shows experimental results in an underwater observational system.

samples of two classes. This method tolerates some misclassifications of the data by introducing slack variables. It aims at maximizing the margin distance from the almost completely separated examples while minimizing the slack variables. Typically, a binary SVM minimizes

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad y_i(\mathbf{x}_i \cdot \mathbf{w}^T + b) \geq 1 - \xi_i \quad and \quad \xi_i \geq 0 \quad \forall y_i \in \{-1, 1\} \quad (4.2)$$

where $\mathbf{w}$ is the normal vector to the hyperplane, b is the bias. This equation can be transformed into a convex quadratic programming optimization problem, and $\mathbf{w}$,b can be calculated by a Quadratic Programming solver.

SVMs were initially designed to be a binary classifier. They can be adapted to form a multiclass classifier by converting the single multiclass problem into multiple binary classification problems [Duan and Keerthi, 2005]. The first strategy is called one-versus-all, which separates one of the classes from the others. Given a new sample, the classification result is predicted by the highest output (winner-takes-all) among all of the binary classifiers. The second strategy uses each pair of the classes and trains a SVM classifier for each of the pairs. This is named the one-versus-one strategy. A voting mechanism is introduced to accept each result of a binary classifier as a vote to the assigned class. Finally all votes are summed up, and the class with the most votes determines the classification result.

To help choose a good classifier for each level of the hierarchy, we tried the Random Forests method [Breiman, 2001] as an exploration on a small dataset of 7200 fish images of 15 fish species, when the full dataset of 241500 images was still in progress. A Random Forest is made of a number of decision trees with binary splits for classification. It predicts responses for new data with the ensemble learned model. In our experiment on 15 species of fish, the Random Forests method was implemented with 50 decision trees. Each tree was constructed using 500 randomly selected features. This Random Forests method and another popular method, Ada-Boost [Liang et al., 2010], were implemented to compare with the multiclass SVM method, as an exploration to choose the appropriate classifier. The experimental results demonstrated that the performance of the multiclass SVM method was better than the Random Forests and Ada-Boost methods.

This kind of multi-class SVM classifier could be considered as a flat classifier because

Method	AR (%)	AP (%)	AC (%)
Random Decision Forests [Ho, 1995]	0.772	0.662	0.914
Random Forests [Breiman, 2001]	0.625	0.782	0.903
Ada-Boost [Liang et al., 2010]	0.753	0.769	0.923
SVM [Cortes and Vapnik, 1995]	0.863	0.858	0.934

Table 4.1: Fish recognition exploration for choosing the most effective classifier. Average Recall (AR), Average Precision(AP), Accuracy by Count (AC) are introduced in the experimental section.

it classifies all classes at the same time [Carlos and Alex, 2010] and omits the inter-class correlations. A shortcoming of the flat classifier is that it uses the same features to classify all classes without considering that some classes have certain similarities and can be better separated by some customized features in a later stage. To overcome the problem of flat classifier, one possible solution is to integrate a domain knowledge database with the flat classifier and construct a tree to organize all classes hierarchically [Deng et al., 2010]. This strategy is called hierarchical classification which inherits from the divide and conquer tactic. Essentially, it uses a hierarchical classification procedure where a customized classifier is trained with specific features at each level [Gordon, 1987].

A taxonomy tree is a typical biological hierarchical classification method. The taxonomy ontology aims to systematize animals into their hierarchical categories. Taxon, as the leaf node of the whole tree, is the foundation of taxonomy knowledge. For each taxon in the taxonomic tree, there is a top-to-bottom description to identify its hierarchical information which contains several concepts, known as Kingdom, Phylum, Class, Order, Family, Genus, Species. The taxonomy methodology is based on the synapomorphies characteristic from the extent to which the taxon is monophyletic, and it indicates the distinction between species, *e.g.* the presence or absence of components (anal-fin, nasal, infraorbitals), particular number (six dorsal-fin spines, two spiny dorsal-fins), particular shape (second dorsal-fin spine long, thick caniniform teeth), *etc.* The abundant valuable knowledge that the taxonomy technique uses in constructing the hierarchical biological system can be adopted into the construction of a fish recognition decision tree, which leads to a combination of machine learning and marine taxonomic knowledge. Based on the assumption that the top 20 species occupy 80% of all observations, hierarchical analysis will help pay more attention to distinguish those popular

fish in the beginning and leave the rare ones for further processing. We used biological taxonomy knowledge to help construct a hierarchical classification tree of the 15 most common fish species as a baseline hierarchical classification method. This tree splits all classes into nine groups at the first level according to their family synapomorphies characteristic and leaves a few similar species to a deeper layer where a customized classifier is used. In Table 4.2, we summarize the most dominant fish from our collection and organized them by their order, family, genus and species. All of them belong to the Actinopterygii class. We analyse the formalization of fish species division to help design and implement the classifier.

Hierarchical classification has several noticeable advantages. Firstly, it divides all classes into certain subsets and leaves similar classes for a later stage. This strategy also helps balance the number of species. Secondly, unlike the flat classifier choosing a feature set based on the average accuracy over all classes, the hierarchical method applies a customized set of features to classify specific classes. As a result, it achieves better performance on similar classes. Thirdly, the hierarchical solution exploits the correlations between classes and finds similar groupings. This is especially useful with a large number of categories [Deng et al., 2010]. Hierarchical structures are popular in document and image categorization. Mathis [Mathis and Breuel, 2002] organizes documents hierarchically by making use of the correlations between topical subjects. Deng *et.al.* [Deng et al., 2009] introduced a new dataset called ImageNet where a large scale hierarchical ontology of images are constructed based on the WordNet knowledge. However, these approaches use pre-defined hierarchical structures without considering how to construct a more accurate tree based on given classes and their properties.

4.2 Algorithm for constructing the hierarchical classification tree

In this section, we present our novel hierarchical classification method called Balance-Guaranteed Optimized Tree (BGOT). It improves the normal hierarchical method by arranging more accurate classifications at a higher level and keeping the hierarchical tree balanced. The whole system consists of two stages: feature extraction and hierarchical classification (illustrated in Figure 4.1).

Order	Family	Genus	Species
Beryciformes	Holocentridae	Myripristis Cuvier	Myripristis kuntee
Perciformes	Acanthuridae	Acanthurus Forsskål	Acanthurus nigrofuscus
Perciformes	Acanthuridae	Zebrasoma Swainson	Zebrasoma scopas
Perciformes	Chaetodontidae	Chaetodon Linnaeus	Chaetodon auriga
Perciformes	Chaetodontidae	Chaetodon Linnaeus	Chaetodon trifascialis
Perciformes	Haemulidae	Plectorhinchus Lacepède	Plectorhinchus vittatus
Perciformes	Labridae Cuvier	Hemigymnus Günther	Hemigymnus fasciatus
Perciformes	Nemipteridae	Scolopsis Cuvier	Scolopsis bilineata
Perciformes	Pomacentridae	Amphiprion Bloch	Amphiprion clarkii
Perciformes	Pomacentridae	Dascyllus Cuvier	Dascyllus reticulatus
Perciformes	Pomacentridae	Pomacentrinae	Pomacentrus moluccensis
Perciformes	Pomacentridae	Pomacentrinae	Plectroglyphidodon dickii
Perciformes	Pomacentridae	Pomacentrinae	Chromis margaritifer
Perciformes	Siganidae	Siganus Forsskål	Siganus fuscescens
Tetraodontiformes	Balistidae	Balistapus Tilesius	Balistapus undulatus
Tetraodontiformes	Ostraciidae	Lactophrys Swainson	Lactophrys bicaudalis
Perciformes	Scaridae Rafinesque	Scarus Forsskål	Scarus rivulatus
Perciformes	Labridae Cuvier	Anampses Quoy	Anampses meleagrides
Tetraodontiformes	Tetraodontidae	Arothron Müller	Arothron hispidus
Perciformes	Chaetodontidae	Chaetodon Linnaeus	Chaetodon speculum
Perciformes	Acanthuridae	Ctenochaetus Gill	Ctenochaetus striatus
Perciformes	Labridae Cuvier	Hemigymnus Günther	Hemigymnus melapterus
Tetraodontiformes	Tetraodontidae	Canthigaster Swainson	Canthigaster valentini
Perciformes	Kyphosidae	Kyphosus Lacepède	Kyphosus cinerascens
Perciformes	Scaridae Rafinesque	Calotomus Gilbert	Calotomus zonarchus
Perciformes	Labridae	Labroides Bleeker	Labroides dimidiatus

Table 4.2: 26 most dominant fish from our collected data and these species are organized by their order, family, genus and species.

Given a set of samples $\{\mathbf{x}_i\}_{i=1}^n$, the feature vector $\mathbf{f}_i = \{f_{i,1}, ..., f_{i,m}\}$ denotes the m feature values for sample $\mathbf{x}_i$. Let $\{y_i\}_{i=1}^n$ indicate the class label of $\mathbf{x}_i$, and $y_i \in \{1, ..., C\}$ where C is the number of classes. Our aim is to construct a classifier h which uses the feature $\mathbf{f}_i$ as input to predict the class label $\tilde{y}_i = h(\mathbf{f}_i)$ that maximizes the classification accuracy.

4.2.1 Constructing the hierarchical classification tree

A hierarchical classifier h_{hier} is designed as a structured node set. Fundamentally, a node is defined as a triple: $\text{Node}_t = \{\text{ID}_t, \tilde{\text{F}}_t, \hat{\text{C}}_t\}$, where ID_t is a unique node number, $\tilde{\text{F}}_t \subset \{\mathbf{f}_1, ..., \mathbf{f}_m\}$ is a feature subset chosen by a feature selection procedure that is found to be effective for classifying $\hat{\text{C}}_t$, which is a subset of classes and their groups. We only consider binary splits (until the final layer), so each node has at most two groups. All samples that are classified as the same group will be transmitted into the same child node for later processing. An example with 15 classes is shown in Figure 4.2, where the ID_t is illustrated in each node and $\hat{\text{C}}_t$ are the local groups. The binary splitting process stops when each group has at most 4 classes (*e.g.* Node ID 4,5,6,7) in order to limit the maximum depth of the tree and avoid overtraining. All the leaf nodes are multiclass SVMs using the One-versus-One strategy.

This hierarchical classification method is presented as an assembly of individual multiclass classifiers. These classifiers are treated as tree nodes. At each node, there are at least two groups of classes. We use the term "group" to indicate a super-class, which includes several classes as a single item. In the following paragraph, we will introduce our strategy to organize training classes into groups. Every child node corresponds to a choice of group. During classification, every sample starts from the root node at the top, and goes through the hierarchical architecture. At a non-leaf node, the classification decision determines which group the test sample belongs to. The sample is then passed to the corresponding child node for further classification. The procedure continues until the test sample reaches a leaf node whose classification result is a single class, instead of a group of classes.

To construct the hierarchical tree, we first aim at finding an optimal split of the given classes at the current node by minimizing the mean misclassification rate between the two child nodes. We search for all possible splits of the classes into two nearly equal sets of classes. We also select the feature subset that achieves the best accuracy for

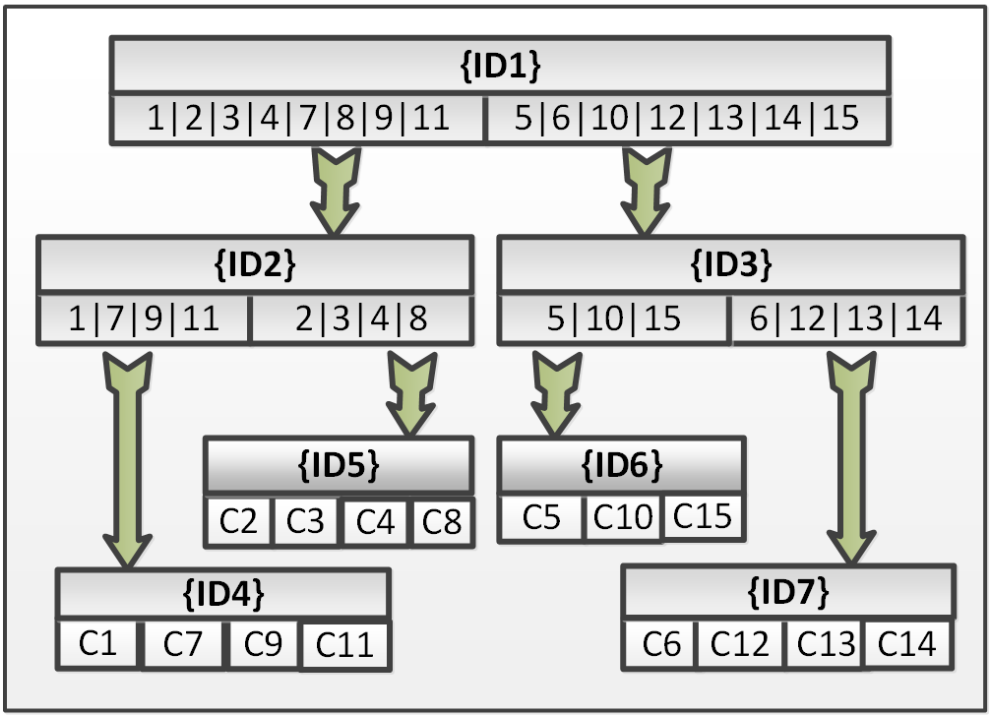


Figure 4.2: Automatically generated tree, the hierarchical example tree of 15 classes ($C1, ..., C_{15}$).

the given split, using forward sequential feature selection based on grouped subset of features. Section 4.2.2 describes the feature selection algorithm. This process is repeated for each child node. A well-designed hierarchical tree can help improve the accuracy of some confusable classes while suppressing the error accumulation. In this section, we propose two heuristics for how to organize a single classifier and construct a hierarchical tree with higher accuracy.

1. Arrange more accurate classifications at a higher level and leave similar classes to deeper layers.

2. Keep the hierarchical tree balanced to minimize the max-depth and control error accumulation. Here we split the tree by equal number of classes, but one could also use other splits, such as by equal *a priori* fish appearance probabilities, or non-equal numbers of classes to minimizing error.

When constructing the hierarchical tree, we focus on balanced trees for computational reasons, and because a balanced tree structure produces reduced tree depth, which reduces error accumulation. More formally, our tree generation algorithm can be described as follows:

A schematic of the program flow is illustrated in Figure 4.3. Firstly, the algorithm splits the current set of classes c into all $\binom{c}{|c/2|}$ combinations of pairs of disjoint subsets with size $\mid \frac{c}{2} \mid$ and then sends each combination to the performance evaluation stage. After evaluating all of the possible splits, the best subset pair, in terms of classification accuracy, is chosen and this split is used to construct two new child tree nodes. This procedure is iterated for both child branches until the stopping criteria are satisfied. Performance evaluation of each subset at a given tree level is independent of every other split. We assign each combination of class set splits to a distributed parallel task. Each pair of subsets is then evaluated to obtain an accuracy score in parallel (the accuracy score for each distributed task is found by taking the mean classification accuracy of the two subsets assigned to the task). After all distributed tasks in a superstep have concluded, we collect all of the mean accuracy scores and select the class split with the highest score (our superstep conclusion).

An example of an automatically generated tree is shown in Fig 4.2, where 15 classes are arranged into 3 layers. The first layer splits all classes into two groups: C1, C2, C3, C4, C7, C8, C9, C11 and C5, C6, C10, C12, C13, C14, C15. Then it chooses the feature subset to maximize the average accuracy of these groups. This procedure keeps on until all groups have at most 4 classes.

4.2.2 Forward sequential feature selection based on grouped subset of features

We use forward sequential feature selection based on grouped subsets of features. The designed features integrate domain knowledge with machine vision methods and considered altogether in the pool for feature selection in the classification step. The generalized model is shown in Figure 4.4. There are two strategies: it extracts features but maintains them grouped, the feature selection procedure chooses an optimized subset of groups from the candidate groups. Using grouped features can reduce the number of candidates in the feature list, where the computing time of FSFS is $O(N^2)$ times the number of candidates considered. This would limit the computation time and is helpful to avoid a local maximum since random variables are also added when choosing the best candidate group. The idiosyncratic fish features and the texture features are most frequently selected by FSFS for use. Half of the tree nodes select REHIST normalized

Input: class C_1 to C_n
begin
  $c := \{C_1, ..., C_n\}$
  $level := 0$
  $featureSet :=$ allFeature(F)
  construct$(c, level)$
end
proc construct(c, n) $\equiv$
  if $n >$ MAXDEPTH
    exit
  end
  // Evaluate classification accuracy on each
  // split of classes c in parallel
  parallel for {binary splits of c}
    $r =$ evaluate$(c, featureSet)$
  end
  // The ChooseSplit finds the optimal class
  // subset pair based on the set of r evaluations
  $[cLeft, cRight] :=$ ChooseSplit$(\{r\})$
  $cFeatureSubset :=$ FeatureSelection$(featureSet, cLeft, cRight)$
  // The maximum leaf node subset
  // size is set to 4 to limit max tree depth
  if size$(|cLeft|) > 4$
    construct$(cLeft, n+1)$
  end
  if size$(|cRight|) > 4$
    construct$(cRight, n+1)$
  end
end

Algorithm 1: Algorithm of generating the BGOT tree.

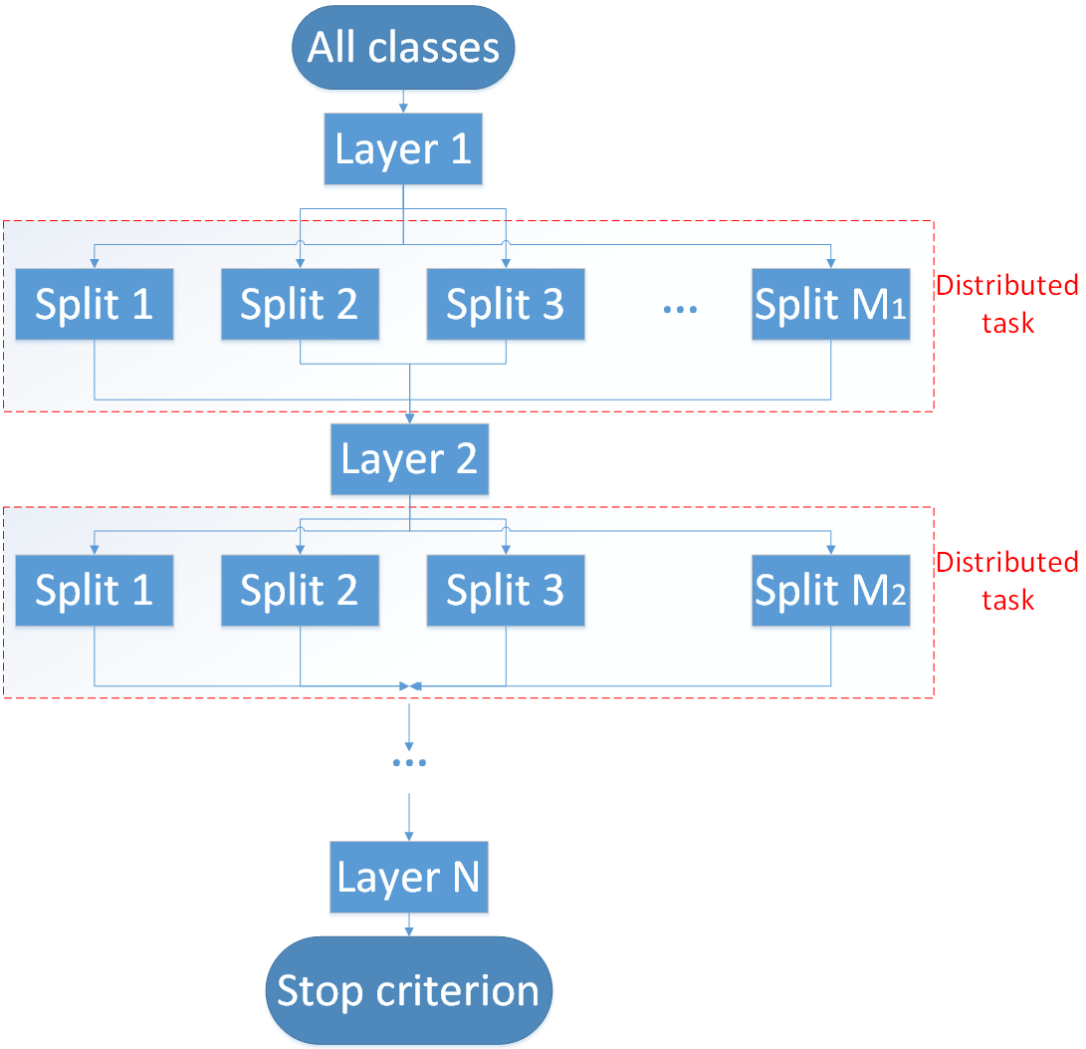


Figure 4.3: The algorithm to generate our balanced hierarchical classification tree. At each tree level, we select the optimal disjoint and balanced class subset split by exhaustively searching all possible splitting combinations. Each set of algorithm stages within a dashed area represents a superstep that is distributed to our cluster in parallel.

colour features and fish boundary descriptors.

We evaluate the performance of individual group of features. The summary table of the relative merits of the features is given in Table 4.3, where each row is the evaluation scores that result when only this group of features is used for classification. As can be seen, some individual feature sets perform reasonably well (e.g. sets 50-64) but not nearly as well as the combined BGOTR results shown in Table 4.4.

4.2.3 Node rejection for misclassified samples

Hierarchical classification is a unidirectional process, in which the test samples go down the hierarchy. Normally, if any sample is misclassified, it moves further down until it reaches the leaf node. As a result, all classification errors are accumulated, and there is no mechanism to correct these mistakes or filter them out of the results. We propose a filter algorithm called node rejection to ease the error accumulation problem, as shown in Figure 4.5. It adds a "-1" branch at each node, and this branch contains all hidden classes which do not appear in this node. Any fish that is classified as "-1" will be re-classified by a flat multi-class SVM. This "-1" branch provides an alternative pipeline so only the fish samples that are similar to the existing classes will be preserved. We examine the hierarchical classification method both with and without node rejection. One advantage of using the re-classification mechanism for misclassified samples is that the feature selection for each tree node can be optimized for improving the accuracy of confident decisions, instead of involving error prone samples. In the case of node $ID2$ in Figure 4.5, the conventional hierarchical classification method creates a binary classifier for the two groups of fish species, where the first group contains species $1, 7, 9, 11$ and species $2, 3, 4, 8$ are assigned to group two. In our method, an additional group is appended so that it filters out the misclassified samples belonging to species $5, 6, 10, 12, 13, 14, 15$. One concern is that node rejection may introduce extra mistakes since it adds an additional group to every node of the existing hierarchy (except the root node). But given the level of average precision/recall ($c.$ $85\% \sim 90\%$) and compared with conventional methods, the node rejection mechanism eliminates sufficient classification mistakes and improves the overall performance. We will justify this approach experimentally in Section 4.3.1.

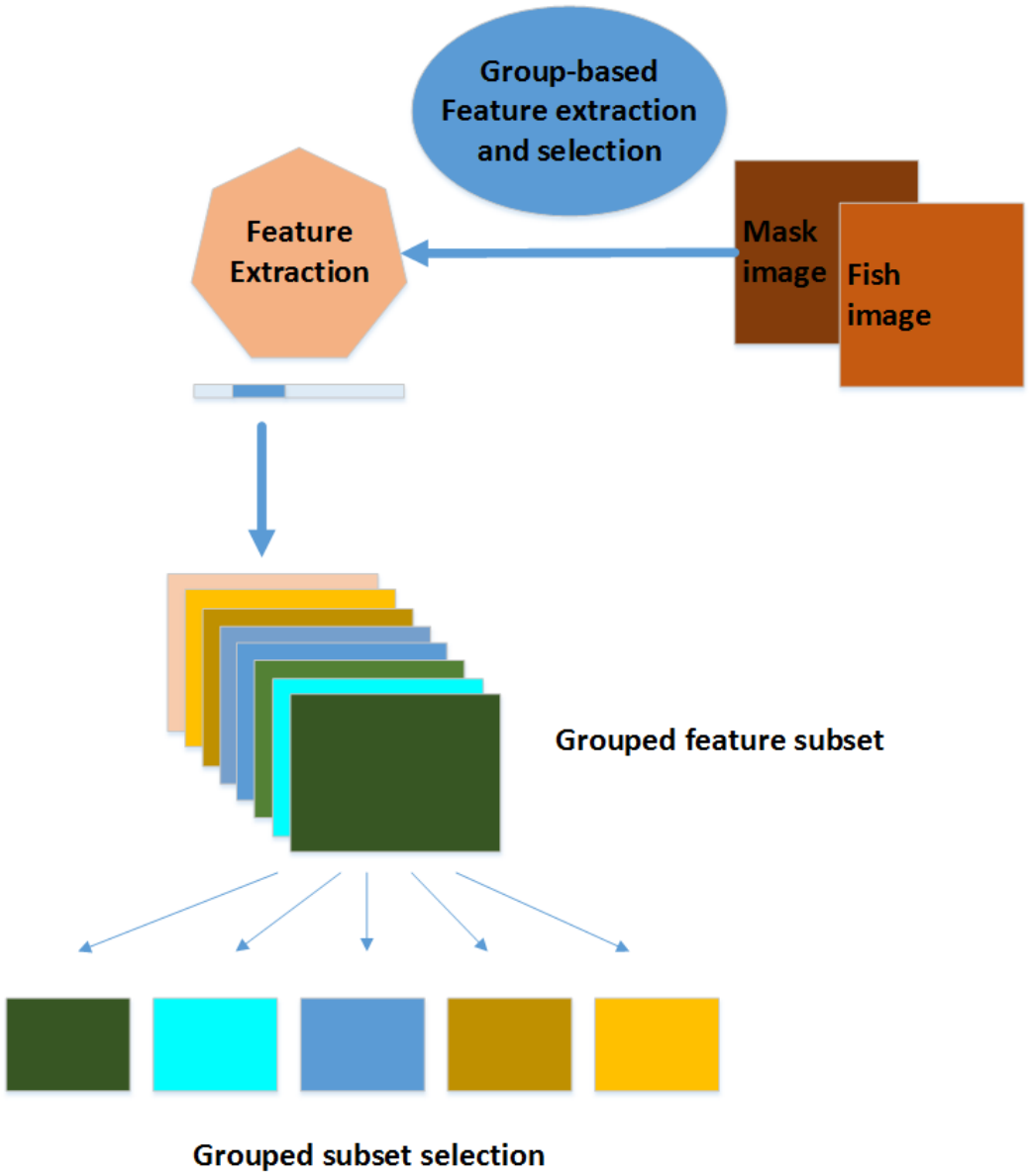


Figure 4.4: Architecture generalizing the group feature selection in Chapter 4. The input fish and mask images are passed to feature extraction component that extracts sets of grouped features. Each set is associated with a number of features that belong to the same type. Feature selection is then restricted to select a whole group of features in each step.

ID	Average Recall (%)	Average Precision (%)	Accuracy by Count (%)
1	17.2	21.9	68.3
2	18.9	26.9	65.7
3	16.5	25.7	65.0
4	17.7	25.2	68.3
5	21.2	27.3	71.8
6	13.7	18.7	62.0
7	13.6	21.5	56.6
8	12.0	23.3	57.6
9	13.8	21.2	59.1
10	16.5	25.6	62.7
11	20.3	26.2	68.1
12	23.5	33.1	68.1
13	21.9	29.6	67.2
14	19.4	22.3	67.7
15	26.0	32.2	72.4
16	15.3	18.0	66.9
17	15.9	22.0	64.1
18	13.7	18.0	62.4
19	15.4	18.2	67.2
20	17.0	20.2	69.7
21	12.4	15.9	58.5
22	9.1	11.0	51.6
23	11.7	15.0	54.5
24	11.1	14.4	53.1
25	13.0	18.6	55.9
26	14.6	15.1	62.8
27	15.4	16.4	62.6
28	16.0	16.4	60.5
29	13.2	12.7	63.0
30	18.5	18.0	65.1
31	10.3	8.4	56.8
32	7.2	6.5	50.9
33	7.0	5.8	50.7
34	8.2	10.3	52.4
35	11.5	13.1	59.2
36	7.1	6.6	50.7
37	6.7	3.4	50.4
38	10.1	13.8	55.5

ID	Average Recall (%)	Average Precision (%)	Accuracy by Count (%)
39	11.1	12.2	57.7
40	8.3	7.3	52.9
41	6.7	3.4	50.4
42	6.7	3.4	50.4
43	9.3	9.0	56.2
44	7.7	6.8	51.6
45	6.7	3.4	50.4
46	6.9	4.0	50.4
47	15.3	21.3	57.3
48	15.5	22.0	57.2
49	28.5	41.6	66.6
50	54.8	63.3	87.3
51	53.9	61.1	87.2
52	53.7	61.3	86.4
53	49.9	56.6	84.7
54	49.8	59.3	83.3
55	49.0	56.3	82.5
56	48.5	56.8	81.1
57	48.2	54.6	80.4
58	47.1	56.3	79.7
59	46.9	57.4	79.2
60	35.0	48.0	81.7
61	41.2	56.8	82.7
62	37.5	56.8	81.2
63	42.7	54.6	80.7
64	44.3	60.6	83.1
65	6.7	3.4	50.4
66	6.7	3.4	50.4
67	6.7	3.4	50.4
68	6.7	3.4	50.4
69	39.4	42.9	82.4

Table 4.3: The summary table of the relative merits of the feature groups. Each entry of this table is the experimental result that when using only this group of features for classification. The feature id is the same as we summarized in the feature chapter.

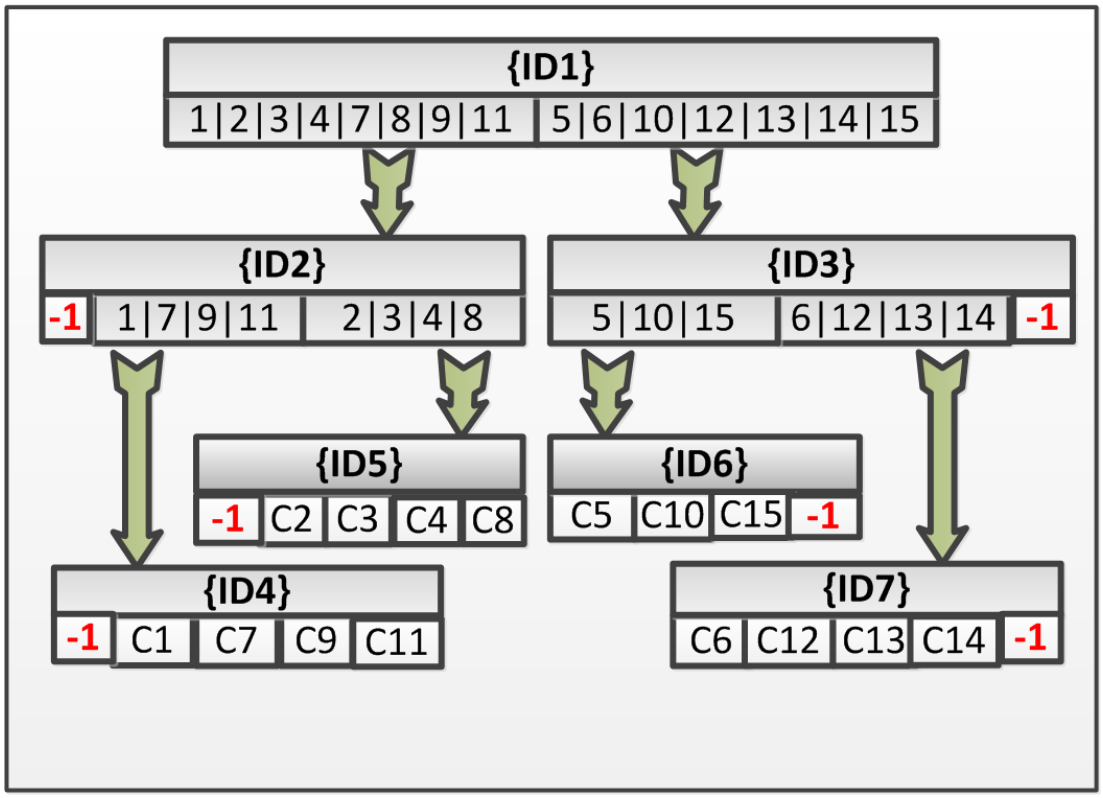


Figure 4.5: A Balance-Guaranteed Optimized Tree computed from training data is shown, where the leaf nodes contain classifiers that either separate the fish into more subclasses or reject the fish for a particular subnode (shown by the "-1" branch), because it is not similar to the fish species in that particular node. Rejected fish are then reclassified by a flat multi-class SVM in this case.

4.2.4 Trajectory voting method

In the view of traditional fish recognition system, the classifier predicts fish species according to individual images. Some classification errors occur due to varying illumination arising either by the fish orientations or light field. We show that fish recognition from consecutive frames of the same trajectory helps eliminate these minor errors and improves the overall accuracy. We have applied the image set classification to the live fish recognition scenario. This method uses a set of observations to recognize test samples. The image set is from a video sequence containing multiple images of the same target. In the literature concerning the image set integration, there are mainly two categories of theories regarding the underlying sequence of result integration: the early integration strategy and the late integration strategy. The former method uses the observations to determine the similarity between image sets, before match-

ing. [Shakhnarovich et al., 2002] consider the features of multiple observations as a whole, and propose a classification based on their distributions. The authors use the relative entropy to compute the covariance matrices of the two input sets and use the Kullback-Leibler divergence metric assuming the input set of vectors form a Gaussian distribution. [Caseiro et al., 2013, Wang et al., 2012] also use the covariance matrix method. However, they represent each image set with its natural second-order statistic (covariance matrix), and then maps it from the Riemannian manifold to a Euclidean space for distance measuring. [Wolf and Shashua, 2003, Yamaguchi et al., 1998] use subspaces (called Mutual Subspace Method, MSM), where the similarity is defined by the minimum principal angle. They use principal angles as a measure for matching two image sequences. The smallest principal angle is defined as the dissimilarity between the two subspaces, and it measures whether the subspaces are similar using a "nearest neighbour" approach. [Kim et al., 2007, Wang et al., 2008] introduce the manifold method that measures the similarity of the common views of the same subject taken from different views. They use a "Manifold to Manifold" distance for the closest subspace pair from the two manifolds. Some other methods like affine/convex hull are used by [Cevikalp and Triggs, 2010, Hu et al., 2011].

On the other hand, the late integration strategy uses likelihoods after matching. These likelihoods could be calculated either by product or by maximizing of the individual decisions. The multiple-instance learning method is also applied to this task and it is introduced by [Maron and Lozano-Pérez, 1998, Zhang and Goldman, 2001, Yang et al., 2005]. Recently, [Shakhnarovich et al., 2002, Everingham et al., 2009] introduced a "min-min" distance for measuring the post-decision similarity.

In our live fish recognition system, we have applied the majority voting algorithm to make use of the temporal information, and it is also used to minimize the environmental influence, as shown in Figure 4.6. This is a late integration strategy. As all fish are freely swimming in a varying illumination environment, the detected fish may have different orientations and appearances. Therefore, the recognition results may vary even for a fish in the same trajectory. A trajectory based winner-take-all voting mechanism is applied after the individual classification. It combines the single frame classification results. The trajectory voting method enhances the fish recognition accuracy by exploiting the consistency in labels expected from tracking each fish individually.

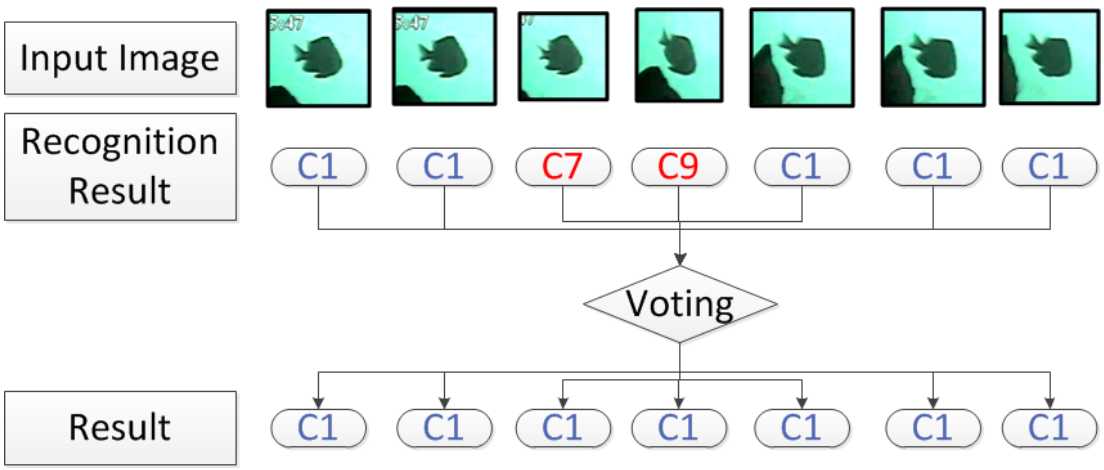


Figure 4.6: An example of trajectory voting is shown. The majority algorithm (also called "winner-take-all" strategy), which counts the votes of each species and uses the highest scores as the final decision, is developed. In this case, two frames of a *Dascyllus reticulatus* fish are misclassified due to a varying illumination arising either by the fish orientations or environmental effects. The proposed trajectory voting method eliminates these minor errors and preserves the majority results.

4.3 Fish recognition experiments

Our data is acquired from a live fish dataset of the 15 different species shown in Figure 4.7. This figure shows the fish species name and the numbers of observations and trajectories in the ground-truth. The data is very imbalanced, where the most frequent species is about 500 times more common than the least one. Note, the images shown here are ideal images as many of the others in the database are a bit blurred, and have fish at different distances and orientations or are against coral or ocean floor backgrounds. Figure 4.8 shows some hard fish examples.

All fish are manually labelled by following instructions from the marine biologists [Boom et al., 2012]. In our experiment, the training and testing sets are isolated so fish images from the same trajectory sequence are not used during both training and testing. We use the pre-processing and feature extraction methods presented in the previous chapter. Pre-processing is undertaken to improve the quality of features. Firstly, the detection and tracking software described in [Nadarajan et al., 2011] is used to obtain the fish and mask images. Then the Grabcut algorithm [Rother et al., 2004] is employed to segment fish from the background, similar to [Edgington et al., 2006, Cline and Edgington, 2010]. Given prior information such as reference frame or pre-

Figure 4.7: Top 15 species of fish in underwater videos, with the number of observations and trajectories in the ground-truth. All in all, there are 24150 observations and 8069 trajectories.

Figure 4.8: Hard fish examples, due to blurred conditions, different distances/orientations, against coral or ocean floor backgrounds.

label foreground area, the graph cut solution gives each pixel a weight between foreground(source) and background(sink), and solves the segmentation problem with a minimum cost cut method to divide the source from the sink. The solution finds the global energy optimum.This approach converts an image processing problem into a graph energy minimization problem, and there is a universal algorithm to tackle the graph cut question. The optimization procedure is based on the similarity between a pixel and its local neighbours. This method could overcome normal image distortion, such as additional noise and water reflection, which triggers segmentation errors in other algorithms.

After feature extraction, 69 types of feature are generated (see Chapter 3). These features are a combination of colour, shape and texture properties in different parts of the fish such as tail, head, top, bottom and the whole fish. All features are normalized by subtracting the mean and dividing by the standard deviation (z-score normalized after 5% outlier removal).

4.3.1 Hierarchical classification for fish recognition

We use the BGOT method for fish recognition. Both flat SVM and hierarchical methods are explored. Both linear and non-linear kernel methods are tested. Based on the multi-class classifier, we designed four other classifiers:

1. A multiclass 1v1 flat SVM classifier, which classifies all 15 classes simultaneously, is implemented as a baseline classifier. Forward sequential feature selection is applied (named flatSVM-fs) to do greedy selection of the features to maximize the average recall among all classes.

2. The Principal Component Analysis (PCA) algorithm is also implemented as a baseline method for feature selection and classification. It uses singular value decomposition (SVD) to reduce the feature dimensions and we preserve 98% of the principal component variance (up to 583 dimensions). The processed features are then classified by a 15-class SVM classifier.

3. The Lasso (L1-constrained fitting) algorithm [Tibshirani, 1996] is a shrinkage and selection method [Zou and Hastie, 2005] for linear regression. It minimizes the usual sum of squared errors, with a bound on the sum of the absolute values of the coefficients. In our experiment, it is implemented as a wrapper procedure

using the scoring function of feature subset. We select features such that the MSE is within one standard error of the minimum (up to 763 dimensions). The selected features are then classified by a 15-class SVM classifier.

4. A classical classification and regression tree method (CART [Hastie et al., 2001]) is provided as another automatically generated hierarchical decision tree to be compared with. It starts with a single node, and then looks for the binary distinction which gives the most information about the class. The generating process continues until it reaches the stopping criterion.

5. A taxonomy tree is constructed according to the fish species taxonomy. This tree is pre-defined. It reflects the homologous similarity between species. All the 15 species of fish belong to the Actinopterygii class (ray-finned fishes), but in different orders, families and genus. This tree splits all classes into 9 groups at the first level according to their family synapomorphies characteristic and leaves a few similar species to deeper layers where the customized multiclass 1v1 SVM classifier is trained (shown in Figure 4.9).

6. An automatically generated tree (BGOT) is designed by recursively choosing a binary split which has the best accuracy over the given classes. Forward sequential feature selection (FSFS) is applied in the BGOT method to select effective subsets of features at each node of the hierarchical tree and the goal of feature selection is to maximize the average accuracy among all classes, which enhances the weight of minority classes. Feature selection typically selects about 300 of the features at each node.

The experiment is based on 24150 fish images with a 5-fold cross validation procedure with a leave-$\frac{1}{5}$-out strategy. The training and testing sets are isolated so fish images from the same trajectory sequence are not used during both training and testing. We applied the majority voting algorithm to make use of the temporal information.

Results for the 5 algorithms are listed in Table 4.4 where the AR and AP are recall/precision averaged over all classes rather than over all fish. This is because of the greatly unbalanced class sizes. Three performance metrics are employed to evaluate the accuracy of the proposed system. The first metric is Average Recall (AR, or Macro-Averaged Recall) over all species. It describes on average how many fish are correctly recognized for each species. This score is more important to our experiment because of the imbalance in the classes. Given True Positive / False Positive / False Negative,

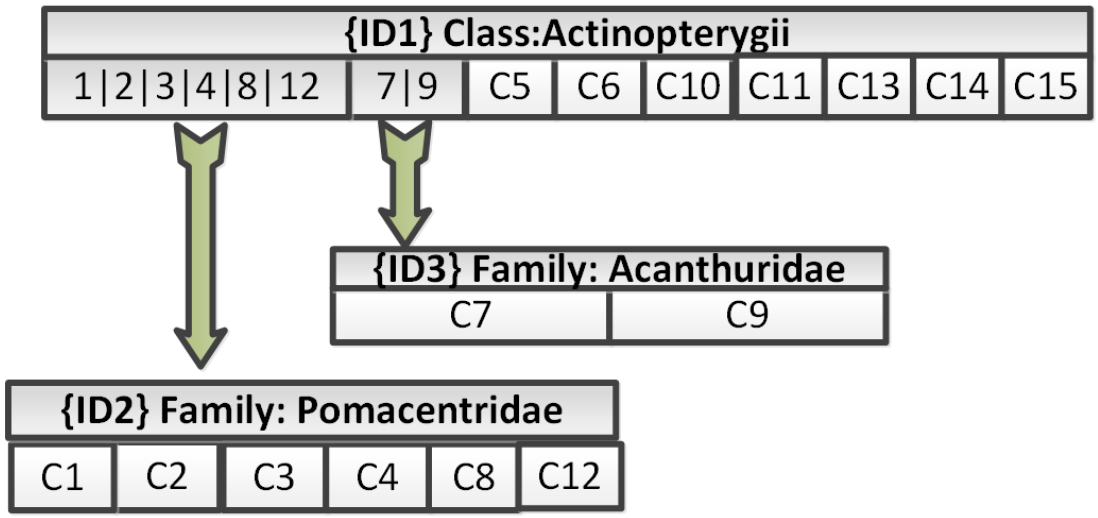


Figure 4.9: A pre-defined taxonomy tree is constructed according to the fish species taxonomy. This tree splits all classes into 9 groups at the first level according to their family synapomorphies characteristic and leaves a few similar species to a deeper layers where the customized multiclass 1v1 SVM classifier is trained.

AR is defined as:

$$AR = \frac{1}{c}\sum_{j=1}^{c}\left(\frac{TruePositive_j}{TruePositive_j + FalseNegative_j}\right) \tag{4.3}$$

where c is the number of classes. The second score is Average Precision (AP, or Macro-Averaged Precision) over all species. It is the probability that the classification results are relevant to the specified species:

$$AP = \frac{1}{c}\sum_{j=1}^{c}\left(\frac{TruePositive_j}{TruePositive_j + FalsePositive_j}\right) \tag{4.4}$$

The third metric is the accuracy over all samples (Accuracy over Count, AC, or Micro-Average Recall), which is defined as the proportion of correct classified samples among the whole dataset. AC is calculated as:

$$AC = \frac{\sum_{j=1}^{c} TruePositive_j}{\sum_{j=1}^{c}(TruePositive_j + FalsePositive_j)} \tag{4.5}$$

We compare the hierarchical classification against the linear SVM classifier (AR = 76.9%). Other non-linear flat SVM methods (polynomial, radial basis function, sigmoid function) are also included but their performances are worse than the linear SVM

Method	AR (%)	AP (%)	AC (%)
SVM (linear)	76.9 ± 4.6	88.5 ± 3.6	95.7 ± 0.5
SVM (polynomial)	61.8 ± 5.0	86.0 ± 7.0	93.0 ± 0.4
SVM (RBF kernel)	70.4 ± 5.6	87.8 ± 6.7	96.0 ± 0.6
SVM (sigmoid)	62.3 ± 5.8	77.1 ± 7.2	85.9 ± 1.0
Lasso	76.6 ± 4.7	85.4 ± 3.3	95.4 ± 0.5
PCA (98%)	77.7 ± 3.8	88.9 ± 4.1	95.4 ± 0.4
flatSVM-fs	78.4 ± 3.7	88.0 ± 5.5	95.9 ± 0.4
CART [Hastie et al., 2001]	53.6 ± 5.1	52.9 ± 4.6	87.0 ± 0.7
Taxonomy	76.1 ± 5.2	87.2 ± 6.7	95.3 ± 0.4
BGOT	**84.8*** ± 3.9	**91.4** ± 2.8	**97.5*** ± 0.6

Table 4.4: Fish recognition results. We add the standard deviation of AR/AP/AC over 5-fold cross validation. * means the score is a significant improvement over other methods at 95% confidence level.

method. PCA is a popular algorithm to reduce feature dimensions. We apply it before an SVM and achieve almost the same score (AR = 77.7%). In the third row, feature selection before use in a SVM produces slightly better results (AR = 78.4%) than the flat SVM using all features. The CART algorithm has the lowest AR (53.6%) among all three hierarchical methods. The taxonomy methodology achieves a better AR of 76.1% than CART but is worse than the automatically generated hierarchical tree (84.8%) which chooses the best splitting by exhaustively searching all possible combinations while remaining balanced. The BGOT method without node rejection has a lower performance (80.1% in AR). Most algorithms achieve high AC score, but this is because the classes are very unbalanced. For example, to simply label all fish as class 1 already achieves an AC = 50.4%.

The individual class recalls/precisions are shown in Figure 4.10 and Figure 4.11. The hierarchical approaches achieve better accuracy than the flat SVM classifier (linear) and other baseline methods because they arrange the similar species into the same group and add fish-tail features to distinguish these species. Species 7,9,11,13 have low scores in part due to confusion with the much larger classes. As shown in Figure 4.7, these species are similar to the most dominate species 1, and our proposed BGOT method presents significant better results in recognizing them than other methods published in the literature.

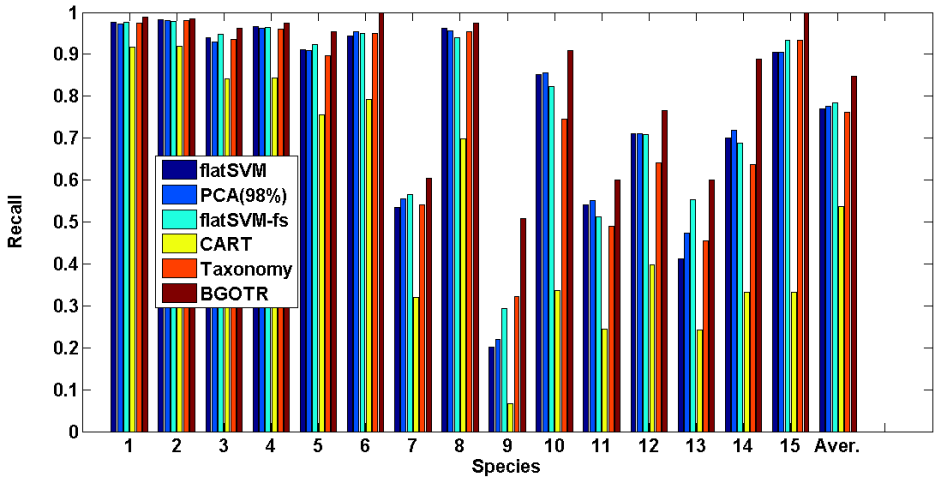


Figure 4.10: Recall of 15 species. These scores are averaged by 5-fold cross validation.

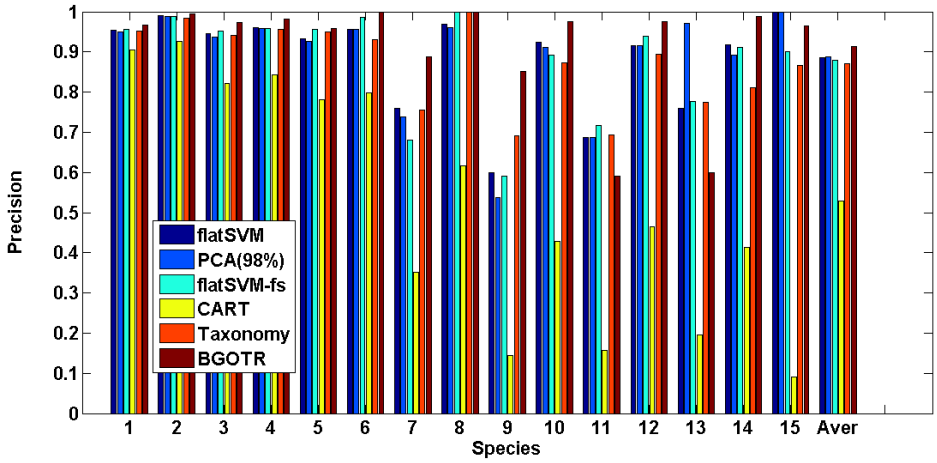


Figure 4.11: Precision of 15 species. These scores are averaged by 5-fold cross validation.

4.4 Discussion

In this chapter, we presented a novel Balance-Guaranteed Optimized Tree (BGOT) classifier for live fish recognition. More specifically, we proposed a set of heuristics which are helpful to construct a hierarchical tree. Although hierarchical classification is widely applied in machine vision applications, BGOT improves the normal hierarchical method by two heuristics for how to organize a single classifier and construct a hierarchical tree with higher accuracy: (1) arranges more accurate classifications at a higher level and leaves similar classes to deeper layers, and thus it searches for the optimal split of the given classes at the current node to minimize the mean misclassification rate between the two child nodes; (2) keeps the hierarchical tree balanced to reduce the max-depth and control error accumulation, so that all possible splits of the classes into two nearly equal sets of classes are tested. In addition, a novel mechanism for classifying confusable samples by training a hidden class in each node and re-classifying these samples in a multiclass SVM is developed to improve the performance of BGOTR. The proposed method is evaluated on a live fish dataset. This dataset of 24k samples over 15 species is the largest and most varied dataset used for fish species recognition research. The strategy of keeping balanced not only balances the number of species, but also help balance the counts of samples. Figure 4.12 shows the counts of training data that go down each path of the BGOT, averaged by 5-fold. In node $ID1$, the ratio of two groups is about 20. In node $ID2$ and $ID3$, the numbers are about equal.

The experimental results demonstrate that the automatically generated hierarchical tree achieves *c.* 6% improvement of the average recall (AR) and *c.* 3% improvement of the average precision (AP) compared to the flat SVM and other hierarchical classifiers (Table 4.4). However, species 7,9,11,13 are similar to the most common fish *Dascyllus reticulatus*, and they are likely to be misclassified. As a result, their performances are worse than other species.

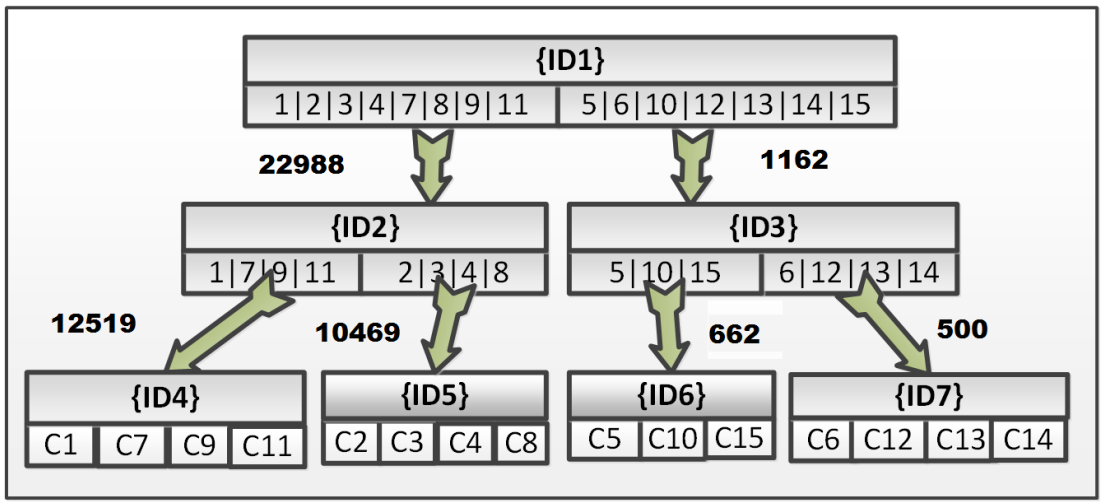
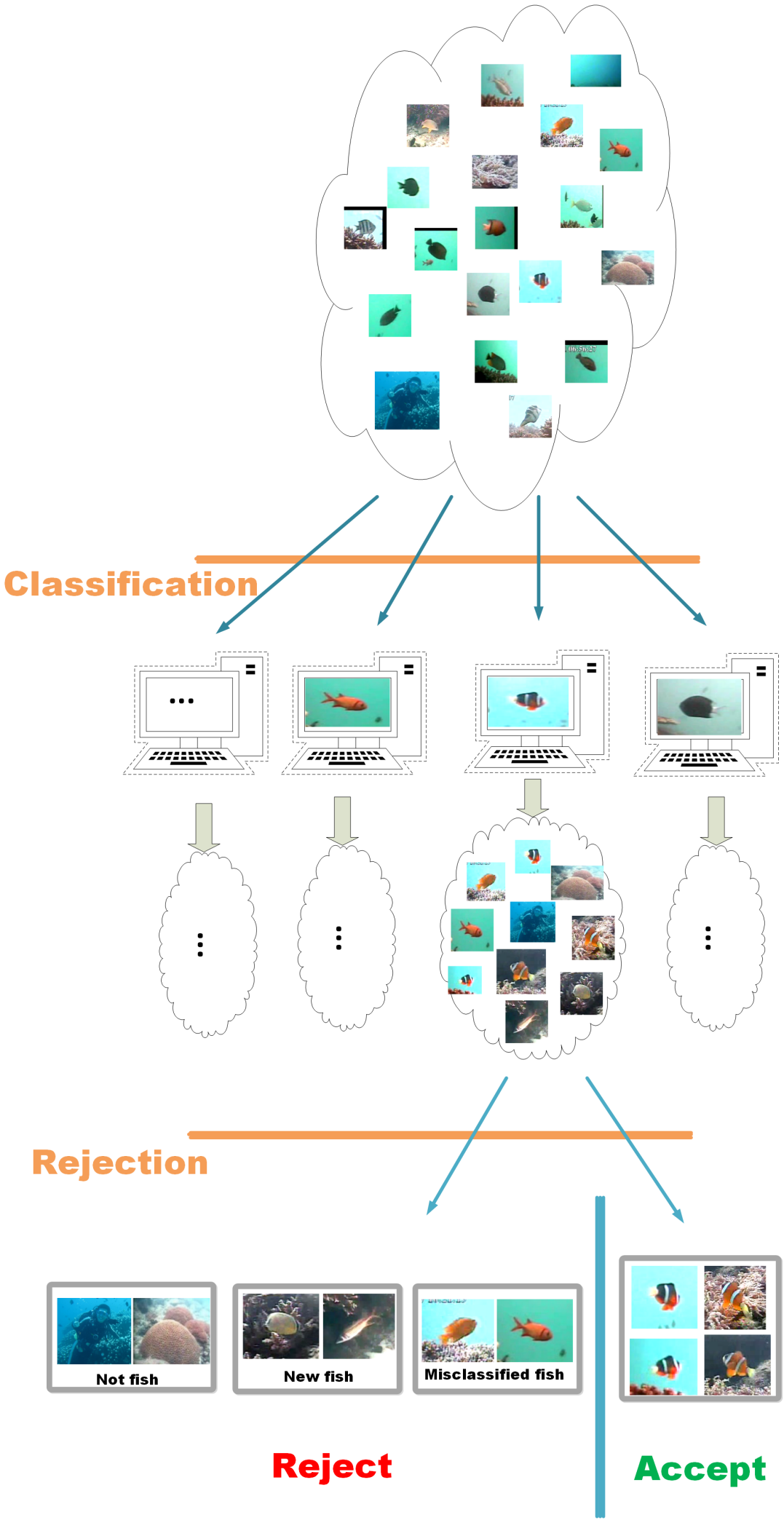


Figure 4.12: The counts of training data that go down each path of the BGOT (averaged by 5-fold).

Chapter 5

Decision refinement after hierarchical classification

This chapter presents a decision refinement method built upon our fish recognition work. The novel innovations of this work arise from the proposed GMM-based reject option. The reject function evaluates the posterior probability of the tested samples given the recognition result. This is a post-recognition step and the rejection is independent of the recognition since it is applied only to the recognition results. The "rejection" term targets the specific application scenarios of: (1) eliminating false positives from the recognition results, and (2) eliminating samples not belonging to the training classes. In the experimental section, we evaluate the performance of our method on these two applications respectively. More specifically, by using a Gaussian Mixture Model (GMM) at each leaf node of the BGOT hierarchical method, a reject option can filter some false detections from the fish detection results (shown in Figure 5.1). It evaluates the *posterior* probability of the classified samples. It produces a lower false positive rate since some misclassification errors in the hierarchical classifier are overcome but at the price of a slightly lower true positive rate due to incorrect rejections. Following the formal description of the proposed model, the experimental results obtained from the manual labelled fish dataset are presented that demonstrate better performance compared to two previous rejection methods.


Classification
Rejection
Not fish
New fish
Misclassified fish
Reject
Accept

5.1 Introduction

As presented in the previous chapter, the fish recognition task is an application of multi-class classification. We used the hierarchical Balance Guaranteed Optimized Tree to overcome the critical drawbacks of the flat classifier and it also improves the conventional hierarchical method. However, a common problem with these hierarchical classification methods is the error accumulation issue. Each level of the hierarchical tree has some classification errors. In fish recognition, especially when our database is extremely imbalanced, misclassified samples are passed into deeper layers and reduce the average accuracy of the final recognition performance. Another issue for a multi-class classifier (not only for hierarchical classification) is that it classifies every test sample into one of the training classes. Although our fish recognition dataset covers the 15 most common species of fish from our videos, there are still many observed fish from unmodeled species. These unknown fish images are classified as known species and the precision is thus decreased. Furthermore, manual annotation work for these minority species is expensive because of the small proportion of these images, when compared to the major species. Thus, the reject option helps the fish recognition application in finding new species.

We address the improvement of rejection in hierarchical classification by calculating the *posterior* probability from Bayes rule. A GMM model is applied at the leaves of a hierarchical tree as the reject option. It evaluates the *posterior* probability of the classified samples and produces a lower false positive rate, since some misclassification errors in the hierarchical classifier can be overcome, but at the price of a slightly lower true positive rate due to incorrect rejections.

In this chapter, we propose a novel rejection system in hierarchical classification for fish species recognition. We also test the proposed rejection algorithm on the Oxford flower dataset. The reject function is integrated with the Balance-Guaranteed Optimized Tree (BGOT) hierarchical method. After a forward sequential feature selection and learning the mixture models, a GMM model is applied to evaluate the *posterior* probability of classified samples and provides a reject option. The rest of the chapter is organized as follows: Section 5.2 briefly introduces the classification reject option. Section 5.3 describes the Gaussian Mixture Model for the reject option. Section 5.4 shows experimental results in an underwater observational system. We also present the experiments and analysis of the proposed method on the Oxford flower dataset.

Conclusions are drawn in Section 5.5.

5.2 Classification with reject option

We are applying a pattern recognition method to recognize fish in underwater videos. This is a multi-class problem with unknown classes. Given the training set $\mathcal{D}$ from p classes, which is a set of n sample points of the form:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{1, ..., p\}\}_{i=1}^{n} \tag{5.1}$$

y_i indicates the class label of m-dimensional vector $\mathbf{x}_i$.

Hierarchical classification has proven effectiveness in imbalanced datasets [Huang et al., 2012], document categorizing [Mathis and Breuel, 2002], and large numbers of classes [Deng et al., 2009]. However, there is a draw-back of the hierarchical classification method: the error accumulation problem. If a sample is misclassified at some intermediate nodes, then it can never be correctly classified. It becomes more critical in an imbalanced data set. The hierarchical algorithm accumulates classification errors when these samples are pushed down the tree. Samples from the minority classes can generate greater cost than the dominant classes if they are misclassified because we optimize the class based accuracy. In order to resolve the error accumulation issue, a reject option eliminates the samples that are dissimilar to the assigned classes. Thus, a p-class SVM has $p+1$ decisions: $\{1, ..., p, Reject\}$. The reject option means either a wrong decision of any of the p classes or the sample is from an unknown class. Platt [Platt, 1999] proposed a rejection method that used an additional sigmoid function $P(y = 1 \mid t) = 1/(1+\exp(at+b))$ to map the SVM outputs into *posterior* probabilities $P(y = \pm 1 \mid t)$ rather than first estimating the class-conditional probabilities $P(t \mid y = \pm 1)$, where t is the SVM output, a and b are parameters trained from validation set. The *posterior* probabilities mapping function can be estimated by using maximum-likelihood method [Wang and Casasent, 2009]:

$$\begin{aligned} < a, b > &= \underset{a,b}{\operatorname{argmax}} P(y = 1 \mid t, a, b) \\ &= \underset{a,b}{\operatorname{argmax}} \prod_i P(y = 1 \mid t^i, a, b), \forall_i \end{aligned} \tag{5.2}$$

where t^i denotes the output for the ith validation sample, and y is the class label. Another common way to give a score to the classifier decisions is the Soft-Decision hierarchical classifier. In [Wang and Casasent, 2009], Wang *et al.* present an implementation

using the SVRDM classifier. The significant change is that there is no constraint that the outputs of each node should sum to one. Given evidence X and the classification result for each sub-branch m, each node i in the classification path generates a probability output $P_i(C = m \mid X)$. The final *posterior* probability P is the product of the corresponding P_i along each path.

5.3 Gaussian mixture model for reject option

A Gaussian Mixture Model (GMM) is a semi-parametric density model which is comprised by a number of Gaussian components [Bishop, 1995]. A GMM model assumes that the data features are originally sampled from a weighted sum of multiple Gaussian functions. In feature space, a GMM provides more flexibility and precision in modelling the underlying statistics of sample data [Mckenna et al., 1998].

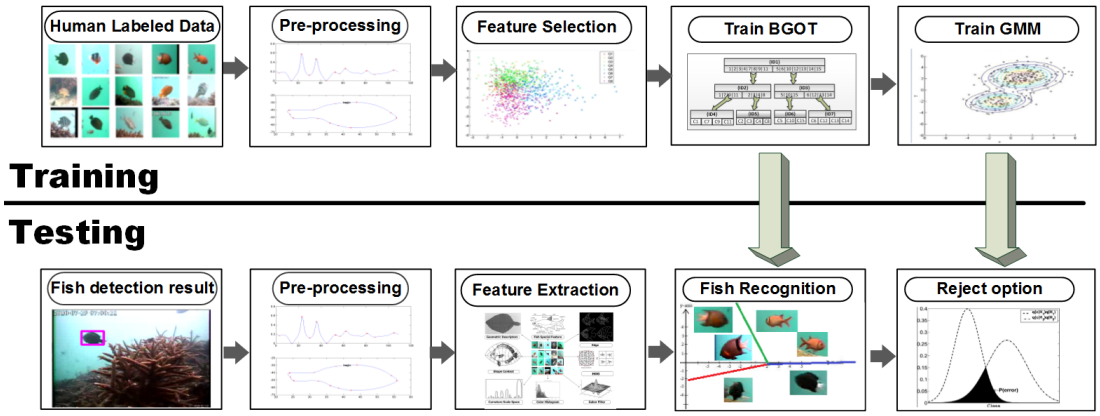


Figure 5.2: Result rejection for fish recognition, framework.

The conditional density for a sample belonging to a given class C in the training set is a mixture with M components of Gaussian densities [Bishop, 1995]:

$$p(\mathbf{x} \mid \theta) = \sum_{i=1}^{M} \omega_i \, g(\mathbf{x} \mid \mu_i, \Sigma_i)$$

$$= \sum_{i=1}^{M} \omega_i \, \frac{1}{(2\pi)^{\frac{D}{2}} \mid \Sigma_i \mid^{\frac{1}{2}}} exp\{-\frac{1}{2}(\mathbf{x}-\mu_i)' \Sigma_i^{-1} (\mathbf{x}-\mu_i)\} \quad (5.3)$$

where x is a D-dimensional continuous-valued data, θ is the parameters of the infinite mixture model, including ω_i and μ_i and Σ_i, $g(\mathbf{x} \mid \mu_i, \Sigma_i)$ is the component Gaussian density, while each component is a Gaussian with mean $\mu_\mathbf{i}$ and covariance matrix Σ_i, ω_i is the mixture weight and satisfies the constraint that $\sum_{i=1}^{M} \omega_i = 1$.

A GMM is employed to represent the hypothetical clusters of density distributions in feature space because individual component Gaussian functions were not sufficient to model the underlying characteristics of the given classes. For example, in fish recognition, some species of fish have specific colours, fin shapes, stripes or texture. It is reasonable to assume that the extracted features represent the domain knowledge and represent them by the density distributions. Each characteristic is expressed both by the mean value μ_i and the covariance matrix Σ_i. The training procedure is unsupervised (after assigned the training class), the GMM captures the prominent density distributions and is not constrained by the label information. In equation 5.3, there are several variables to be fit in this step, like μ_i, Σ_i. The Expectation Maximization (EM) algorithm [Shental et al., 2003], which is guaranteed to converge to a local maximum by iteratively searching, is applied to optimize the Gaussian mixture model. Figueiredo *et al.* [Figueiredo and Jain, 2002] present an unsupervised learning algorithm to learn a proper mixture model from multivariate data. It could automatically select the finite mixture model by using the minimum message length (MML) with advantages compared to other deterministic criteria, *e.g.* BIC, MDL: less sensitive to the initialization, avoids the boundary of the parameter space. This method inherits from the MDL criterion:

$$\hat{m}_{MDL} = arg\min_{m}\{-\log p(\mathcal{X} \mid \hat{\theta}(m)) + \frac{m}{2}\log n)\} \tag{5.4}$$

where $\mathcal{X}$ is the random variable, θ is the parameter vector, $-\log p(\mathcal{X} \mid \hat{\theta}(m))$ is the data code-length and $\frac{m}{2}\log n$ stands for the code-length proportional requirement for each of the m components of $\hat{\theta}(m)$. After replacing the expected Fisher information matrix $I(\theta) \equiv -E[D^2_\theta \log p(\mathcal{X} \mid \hat{\theta})]$ by the complete-data information matrix $I_c(\theta) \equiv -E[D^2_\theta \log p(\mathcal{X}, \mathcal{Y} \mid \hat{\theta})]$, which upper-bounds $I(\theta)$ and requires the exact limit of non-overlapping components, the objective function of MML becomes:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\theta, \mathcal{X}) = \underset{\theta}{\operatorname{argmin}} \frac{\tilde{N}}{2}\sum_{i=1}^{\tilde{M}} \log(\frac{n\omega_i}{12}) + \frac{\tilde{M}}{2}\log\frac{n}{12} + \frac{\tilde{M}(\tilde{N}+1)}{2} - \log p(\mathcal{X} \mid \theta) \tag{5.5}$$

given a set of n independent and identically distributed samples, where $\tilde{M}$ is the upper-bound of all possible m-component mixtures, $\tilde{N}$ is the number of parameters specifying each component.

One difficulty for rejection in a hierarchical method is how to evaluate a probability score based on the intermediate classification results at different layers. Instead of in-

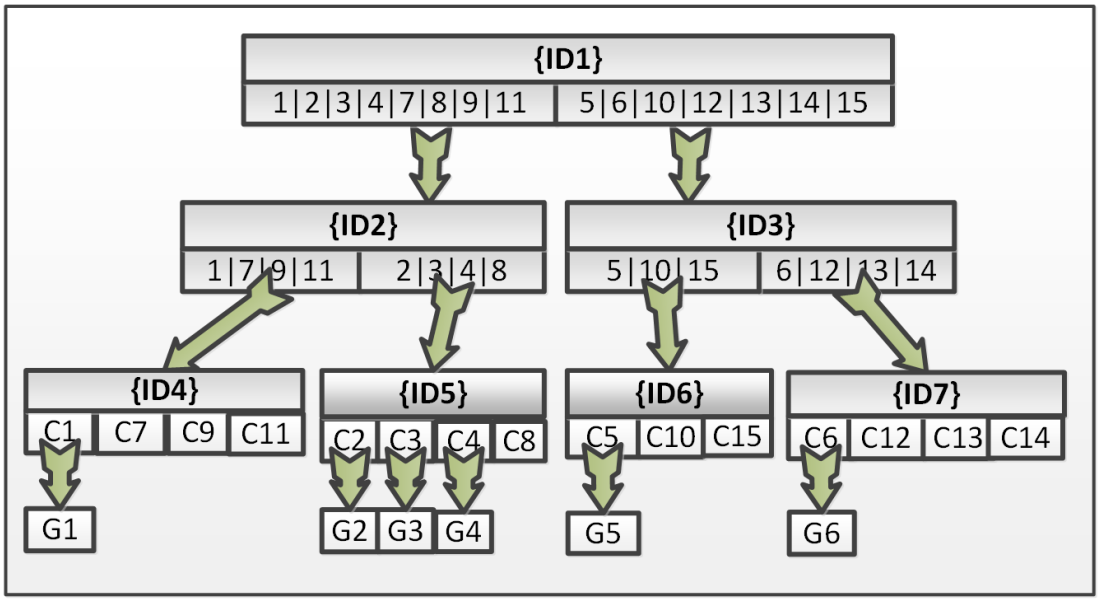


Figure 5.3: GMM for rejection in hierarchical classification, integrated with a BGOT method.

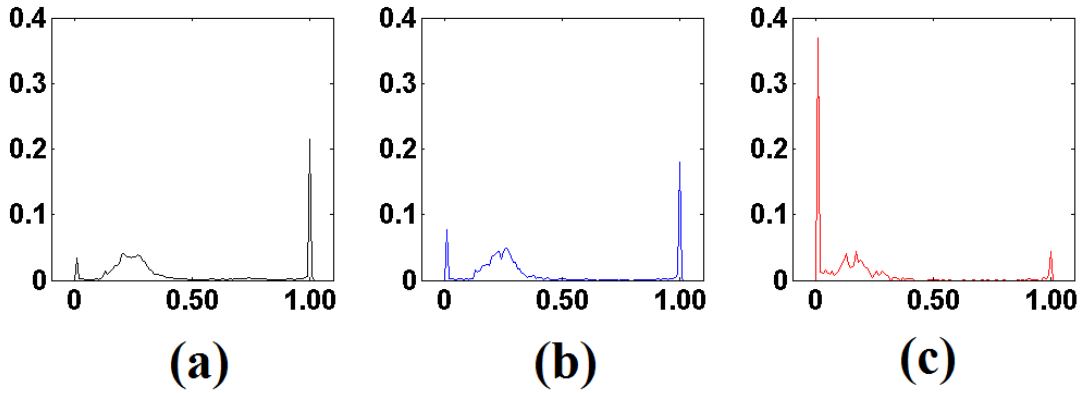


Figure 5.4: (a) Distribution of *posterior* probability of the training samples of species *Chromis chrysura*. (b) Distribution of *posterior* probability of test sample True Positives. (c) Distribution of *posterior* probability of test sample False Positives. See text for details.

tegrating the result score along the path of the hierarchy, here a GMM model is applied after the BGOT classification to implement the reject option (Figure 5.3). The GMM model is trained by a subset of features by using the forward sequential selection method. For each BGOT result, the final $P(C \mid x)$ for that input is estimated according to the GMM likelihood score. More specifically, the rejection uses the *posterior* probability for the predicted class C_i giving evidence X:

$$p(C_i \mid X) = \frac{p(C_i)p(X \mid C_i)}{p(X)} = \frac{p(C_i)p(X \mid C_i)}{\sum_j p(C_j)p(X \mid C_j)} \tag{5.6}$$

where the *prior* knowledge $p(C_i)$ is calculated from the training samples. The features used for training the GMM are the same as for BGOT but a different subset was selected. In [Chib, 1995], Chib and Siddhartha express the marginal density as the *prior* times the likelihood function over the *posterior* density. They found comparable performance of the marginal likelihood with an estimation of the *posterior* density. Since we address the improvement of rejection in hierarchical classification, we also calculate the *posterior* density of the testing samples by Bayes rule. For each sample with evidence X and BGOT prediction C_i, we calculate its *posterior* probability $P(C_i \mid X)$ from Equation 5.6 and set a small threshold (*i.e.* 0.01) to reject all samples whose *posterior* probabilities are below the threshold. Figure 5.4 illustrates the distribution of the *posterior* probability $p(C_i \mid X)$ of all samples that are classified as species *Chromis chrysura*. These samples are either correctly classified (True Positives, Figure 5.4 b) or misclassified (False Positives, Figure 5.4 c). The distribution of the *posterior* probability of False Positives (as shown in Figure 5.4 c) has a peak distribution (about 38%) around the value of zero while most of the True Positives have higher *posterior* probability (Figure 5.4 b). The diversity between these two distributions is exploited to distinguish False Positives. This algorithm rejects a substantial portion of the misclassified samples with the cost of also rejecting a small proportion of True Positives (see experiment section for details).

5.4 Experiments

We evaluate the reject option with an application for fish recognition. The experiments are carried out by comparing our GMM-based method with two state-of-the-art methods: 1) relating SVM outputs to probabilities, and 2) soft-decision hierarchical

classification with a reject option. We also test the proposed rejection algorithm on the Oxford flower dataset.

Figure 5.5: Fish data: 15 species, 24150 fish detections. This figure is identical to Figure 4.7. We duplicate it here. The images shown here are ideal image as many of the others in the database are a bit blurry, and have fish at different distances, and orientations or are against coral or ocean floor backgrounds.

5.4.1 Fish database

The data is acquired from underwater cameras placed in the Taiwan sea with 24150 fish images of the top 15 most common species as shown in Figure 5.5. This is a challenging task due to low quality of images, blur, varying range/orientations and diverse backgrounds. Fish can move freely and illumination levels change frequently both lo-

cally from caustics arising from ocean surface waves and globally according to sun and cloud positions. The fish species are manually labelled by following instructions from marine biologists. This figure shows the fish species name and the numbers of images. The fish detection and tracking software described in [Nadarajan et al., 2011] is used to obtain the fish images. 5-fold cross validation is applied. 24150 images of 15 species are split for 5-fold cross-validation. Approximately, 14490 images are for training, 4830 for validation, and 4830 for testing. Each species is sampled in the same proportion. The training and testing sets are isolated so fish images from the same trajectory sequence are not used during both training and testing. The GMM needs estimated covariance matrices and species 7-15 did not have enough training samples for that estimation, given the number of features selected. Thus, we only apply the reject option to the top 6 species (shown in Figure 5.6). In addition 3220 images from 8 new species (shown in Figure 5.7) are added to the test set to test the performance in probing unknown classes. None of these new samples are from the top 15 species, thus the trained model has no *prior* knowledge about these new classes.

5.4.2 Result rejection in fish recognition

We used the hierarchical classification method BGOT [Huang et al., 2012] for this imbalanced data set. It applies two strategies to help control the error accumulation: arranges more accurate classifications at a higher level and leaves similar classes to deeper layers, while it keeps the hierarchical tree balanced by class to minimize the max-depth. Some pre-processing procedures like fish orientation and fish mask enhancement are undertaken to improve the recognition rate. Next is the feature extracting step. Altogether, 2626 dimensions of features are acquired. They are a combination of colour, shape and texture properties in different parts of the fish such as tail/head/top/bottom, as well as the whole fish. All features are normalized by subtracting the mean and dividing by the standard deviation (z-score normalized after 5% outlier removal). For each fish species, we trained a GMM with the selected feature subset by the forward sequential selection method and the feature selection typically selects about 10-30 features. We then used the learning method presented by [Figueiredo and Jain, 2002] to select the number of mixture models where the maximum Gaussian density component is set as 7. Individual GMMs are trained for each of the top 6 species, which dominate the data set, by using EM algorithm. We classify all fish images and apply the reject option to the classification results that are predicted

as one of the top 6 species. For each sample being classified, the final $P(C \mid x)$ for that sample is evaluated to estimate the classification probability according to the GMM likelihood score. Samples with a low probability are rejected.

Figure 5.6: Dominant fish species used in experiments. We apply the reject option to these species as the dataset is imbalanced and the other species do not have adequate samples to train the rejection model after feature selection.

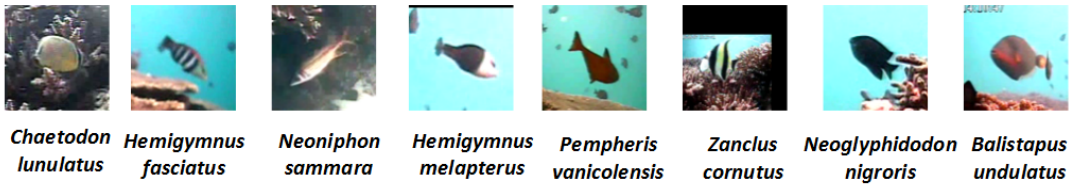


Figure 5.7: 8 new species of fish. They do not belong to any of the training species used in the experiments.

5.4.3 Result analysis and discussions

Figure 5.4 illustrates the different distributions between misclassified and correctly classified samples. After BGOT classification, we eliminate the test samples whose *posterior* probability is lower than the threshold T. This method rejects a significant portion of the misclassified samples (True Rejection, TR) while the cost is that it also rejects a smaller proportion of correctly classified samples (False Rejection, FR). We evaluate the performance of rejection (over 5-fold cross validation) by three factors: True Rejection rate of known classes (the test samples from top 15 classes, which are misclassified and correctly rejected), True rejection rate of unknown classes (the test samples from new classes, which are necessarily classified into one of the top 15 classes and then correctly rejected), False Rejection rate (correctly classified samples but falsely rejected).

Tables 5.1 and 5.2 demonstrate that using the GMM effectively improves the reject option in hierarchical classification for fish recognition. In Table 5.1, the second and

Species	TRs (known class)		TRs (new class)	
	rate(%)	number	rate(%)	number
D. reticulatus	13.7	15	11.2	33
A. clarkii	20.3	4	11.4	212
C. chrysura	32.8	15	51.2	53
P. dickii	13.9	6	14.8	19
M. kuntee	41.7	6	80.6	13
L. fulvus	65.7	4	48.6	106

Table 5.1: Rejection result of incorrect classifications from either trained 15 species (cols 2,3) or new 8 species (cols 4,5), averaged by 5-fold cross validation. (TR=True Rejection). For *D. reticulatus*, the algorithm rejects 13.7% (15) of the known classes that were incorrectly classified as *D. reticulatus*. Similarly, 11.2% (33) of the unknown species classified as *D. reticulatus* were rejected.

Species	True Positives		False Rejections	
	rate(%)	number	rate(%)	number
D. reticulatus	91.9	2237	4.1	95
A. clarkii	95.7	775	0.7	6
C. chrysura	85.2	606	8.0	53
P. dickii	92.5	496	1.8	9
M. kuntee	80.4	74	2.1	1
L. fulvus	84.2	35	1.7	1

Table 5.2: True positive rate among 15 classes after rejection (cols 2,3) and additional false rejections due to the rejection step (cols 4,5), averaged by 5-fold cross validation.

third columns indicate how many misclassified samples from the top 15 species are correctly rejected while the fourth and fifth columns display correctly rejected samples from the new species. In Table 5.2, the last two columns show how many correctly classified fish are thrown out (False Rejection rate) after we have applied the reject option. In a preferable example, *e.g.*, for all test samples that are classified as *Lutjanus fulvus*, 65.7% of misclassified known species samples and 48.6% of new species samples are identified and truly rejected, while only 1.7% of the correctly classified samples are falsely rejected. However, as fish can move freely and illumination levels change frequently in such environments, fish images, even from the same fish, have enormous variations. There are some test samples whose feature distributions are not effectively captured by the GMM. We need to keep a cautious attitude and only filter out samples whose *posterior* probabilities are significantly low. We have to balance the tradeoff between more rejection and more remaining. For example, the cost of the reject option for *Chromis chrysura* is that we throw away 8.0% (53 images) of correct fish while we have correctly rejected 32.8% and 51.2% of the wrongly classified fish from training species and new species, respectively.

The system performance of fish recognition is evaluated by Average Recall (AR) and Average Precision (AP). The experiment result table 5.3 demonstrates that our method rejects a substantial portion of the misclassified samples (significant improvement in AP) while the cost is that it also rejects a small proportion of correctly classified samples (small reduction in AR). We compare it to two other rejection algorithms. As presented by [Platt, 1999], the author fit a sigmoid function on the validation set to the discriminant values produced in the classification step, and used the sigmoid to predict the rejections for test samples. In [Wang and Casasent, 2009], the authors compute the final *posterior* probability as the product of the corresponding discriminant values along the hierarchical decision path. The test samples with low *posterior* probability are rejected. The experimental results show that our method achieves significantly better performance in AP. The proposed method improves BGOT hierarchical classification in two aspects: 1) filters out part of the misclassified samples and increase the averaged precision with a small reduction of the average recall, 2) finds potential new samples which do not belong to the training classes. It detects a set of samples which have a higher probability of coming from new species, and therefore, reduces the work of finding the new fish, especially in a large database of underwater videos.

To summarize our result, we use the F-score to consider both the average recall and

Algorithm	AP (%)	AR (%)
BGOT baseline (no rejection) [Huang et al., 2012]	56.5	**91.1**
BGOT+SVM probabilities [Platt, 1999]	59.0	90.9
BGOT+soft-decision hierarchy [Wang and Casasent, 2009]	58.9	90.7
BGOT+GMM (proposed method)	**65.0***	88.3

Table 5.3: Fish recognition result averaged by species with reject option, averaged by 5-fold cross validation. * means significant improvement with 95% confidence by t-test.

the average precision of the test. The general formula of the F-score for a positive real β is:

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{5.7}$$

We use the F_1 measure, which is the harmonic mean of precision and recall, as shown in table 5.4. The addition of the rejection mechanism gives a significant improvement.

Algorithm	F_1-score
BGOT baseline (no rejection) [Huang et al., 2012]	0.7135 ± 0.0227
BGOT+SVM probabilities [Platt, 1999]	0.7150 ± 0.0222
BGOT+soft-decision hierarchy [Wang and Casasent, 2009]	0.7140 ± 0.0225
BGOT+GMM (proposed method)	$\mathbf{0.7485 \pm 0.0194}$ *

Table 5.4: F-score result averaged by species with reject option, averaged by 5-fold cross validation. * means significant improvement with 95% confidence.

We have also integrated the GMM with a naive Bayes classifier as alternative rejection method, instead of using GMM to evaluate the *posterior* probability given the BGOT prediction. The experiment is evaluated on the top 6 species. The result scores demonstrate that the BGOT+GMM method performs better than the GMM+naive Bayes method in both AR and AP scores. More specifically, the BGOT+GMM method achieves 73.8% in AR and 88.0% in AP, while the GMM+naive Bayes method has 64.7% and 60.7%, respectively. The result analysis shows that the GMM scores could demonstrate the likelihood that whether the test sample is likely a fish, however, these scores are difficult to illustrate the differences between fish species. As a result, the GMM and naive Bayes method shows a worse performance compared to our proposed method.

5.4.4 BGOTR application to new real fish videos

Figure 5.8: Invalid fish images, chosen from 3 underwater videos. In a normal classifier without a reject option, these images would be classified and cause unexpected results. Our rejection algorithm aims at eliminating them while preserving most valid fish images.

Our fish recognition system depends on the detection results. Due to the complex environment (*e.g.* light distortion, fish occlusions and illumination transformation), the fish detection algorithm produces errors that are input to the classification procedure and cause unexpected recognition results. The previous experiments are evaluated on a "clean" dataset where all tested images are valid fish from either known or unknown species. However, in real applications, the acquired data may contain false detections, *e.g.* blurred images, occlusion by other fish or background objects, non-fish objects (coral, sea flowers, *etc.*). Some examples of false detections are shown in Figure 5.8. In this section we experimentally evaluate how many false detections our BGOT system can reject while preserving the valid ones. We choose 3 underwater videos and have labelled 1000 detections from each video.

ID	Average Recall (AR)	Averaged Precision (AP)
video1	0.815	0.412
video2	0.804	0.448
video3	0.725	0.557
average	0.781	0.472

Table 5.5: Experiment result for real videos. In each video we select the first 1000 detections and manually label all samples.

The recognition results are shown in Tables 5.5 and 5.6. We use BGOT to classify the test images and calculate the Average Recall (AR, macro recall) and Averaged Precision (AP, macro precision) among all 15 species. The AR score demonstrates that the BGOT method recognizes about 78% of the real, untrained valid fish images correctly. The test images include many invalid detections (692, 892, 487, respectively). The

ID	True detections	False detections	Rejections	TR	FR
video1	308	692	390	378	12
video2	148	852	734	705	29
video3	513	487	380	312	60
average	323	677	501	465	34

Table 5.6: Experiment of rejection result in real videos. TR = True Rejection, FR = False Rejection.

BGOT method filters more than half of these false detections (378, 705, 312, respectively) while it retains most of the valid inputs. Some false detections are not rejected and these inputs lower the average precision score (*c.* 47%).

5.4.5 Application of the reject option to flower image classification

We applied the proposed rejection algorithm on a popular dataset: the Oxford flower datasets with 17 classes of common flowers in the UK [Nilsback and Zisserman, 2006] (as shown in Figure 5.9). This task is also difficult because the images have large scale, pose and light variations. Some classes are quite similar to others and they both have enormous variations. The authors, Nilsback and Zisserman, used a visual vocabulary method for the flower classification and they produced an accuracy of 81.4% over all samples with 3-fold cross validation. In [Nilsback and Zisserman, 2007], the authors applied a segmentation algorithm to 13 categories of flowers (753 flower images), while the other 4 classes (snowdrops, lily of the valley, cowslips and bluebells) are omitted because their foreground objects are too small for segmentation. We exploited the segmentation results and used the same feature extraction and hierarchical classification of [Huang et al., 2012]. We used the BGOT method as described in previous sections. We trained a 13-class BGOT tree and it achieves an accuracy score of 83.2%, which is better than a flat SVM with forward sequential feature selection (82.0%). Note, the visual vocabulary method is based on all 17 classes while our training classes cover the 13-class subsets which have the segmentation results.

To evaluate the performance of the reject option on the flower dataset, we chose another 7399 samples of 90 different classes from an extended flower dataset which is provided by the same authors [Nilsback and Zisserman, 2008]. This dataset consists of 102 categories of flowers and we exclude the 12 classes which already exist in the

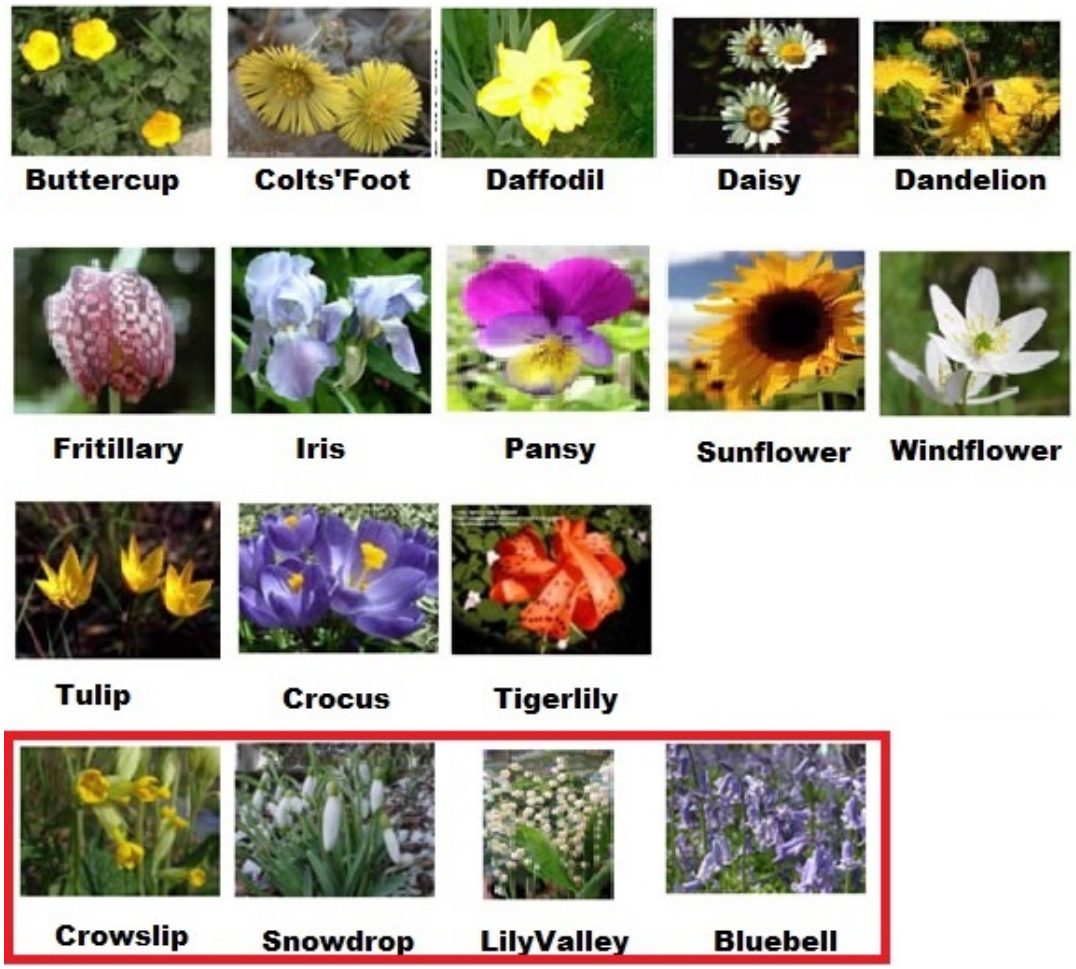


Figure 5.9: Flower dataset of common flowers in the UK. Four classes (snowdrops, lily of the valleys, cowslips and bluebells, as marked within the red box) are not segmented due to the small size of the foreground objects.

TRs (known class)		TRs (new class)	
rate(%)	number	rate(%)	number
21.5	7	37.4	2764
True Positives		False Rejections	
rate(%)	number	rate(%)	number
83.2	158	4.0	6

Table 5.7: Rejection performance of classification result, averaged over 3-fold cross validation. (TR=True Rejection)

training set. We repeated our proposed rejection algorithm after the classification and calculated the *posterior* probability of these results by using a GMM for each of the 13 classes (the same as we did for the fish dataset). Each GMM is trained on an FSFS selected subset of features where the feature selection algorithm maximizes the accuracy of classifying the given class from all other classes. The distributions of the *posterior* probability of the three different groups (True Positives, False Positives, New classes) are shown in Figure 5.10. We set a small threshold (*i.e.* 0.01) and reject all test samples whose *posterior* probabilities are below the threshold. As a result, the proposed method filters out a significant portion of True Rejections (misclassified samples, either False Positives or samples from new classes, shown by the scores of 21.5% & 37.4%, respectively) with a small cost (4.0%) of the False Rejections (correctly classified samples but falsely rejected), as shown in Table 5.7. This task is challenging since the trained GMM has no *prior* knowledge about any of the new classes. The proposed rejection method has rejected more than one third (37.4%) of the test samples from the new classes, at the cost of a slight reduction of accuracy (4% True Positives are falsely rejected).

In this experiment, we added a reject option to the normal multiclass classifier, at the price of a slightly lower accuracy due to incorrect rejections. In previous research, [Gehler and Nowozin, 2009] has a better accuracy by about 2%, but our experimental result could be improved if we optimised our features for the flower dataset, instead of using the same features for fish recognition.

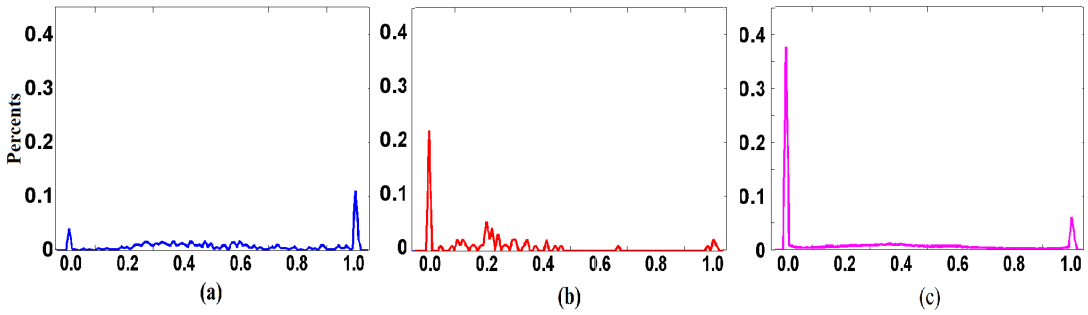


Figure 5.10: *Posterior* probability of the samples of True Positives (a), False Positives (b), test samples from new classes (c). The average *posterior* probabilities of both the False Positives and test samples from new classes are lower than the True Positives. We set a small threshold (*i.e.* 0.01) and reject a significant portion of misclassified samples (the rear peaks in b & c).

Algorithm	Accuracy (%)
Visual Vocabulary [Nilsback and Zisserman, 2006]	81.3 *
SVM-fs	82.0 ± 2.0
SVM (1-vs-All) [Varma and Ray, 2007]	82.6 ± 0.3
LPBoost [Gehler and Nowozin, 2009]	85.4 ± 2.4
BGOTR	83.2 ± 2.6

Table 5.8: Flower recognition results from the literature and our BGOTR method, averaged by 3-fold cross validation. * means that literature [Nilsback and Zisserman, 2006] uses overall accuracy, no standard deviation.

5.5 Conclusion

This section adds a novel rejection system to the hierarchical classification algorithm as applied for fish species recognition. We apply a GMM model at the leaves of the hierarchical tree as a reject option. We use feature selection to select a subset of effective features that distinguishes the samples of a given class from others. After learning the mixture models, the reject function is integrated with a BGOT hierarchical method. It evaluates the *posterior* probability of the testing samples and reduces the false positive rate, since some misclassification errors in the BGOT classifier can be overcome at the price of a slightly lower true positive rate due to incorrect rejections. The experimental results demonstrate a reduction in the accumulated errors from hierarchical classification and an improvement in discovering unknown classes in comparison to two other

rejection algorithms.

Chapter 6

Individual feature selection for one-versus-one classifier improves multiclass SVM performance

In this chapter, we investigate the process of feature selection and constructing a multiclass SVM. We propose that an Individual Feature Selection (IFS) procedure can be directly exploited to binary One-versus-One (OvO) SVMs before assembling the full multiclass SVM, and this approach gives better performance than globally selecting the features. The usual Multiclass Feature Selection (MFS) algorithm chooses an identical subset of features for every OvO SVM. The proposed method selects different subsets of features for each OvO SVM inside the multiclass classifier so that each vote is optimized to discriminate between the two specific classes. While this is a simple and seemingly obvious variation, we have not found any report of it in the literature. Forward sequential feature selection (FSFS) is taken as the generic mechanism, so the comparison focuses on the differences between the MFS and IFS methods. Following the technique discussion of the proposed IFS framework, this chapter gives a formal estimate of the computational complexity. The proposed IFS method is tested on four different datasets for comparing the performance and time cost. Experimental results demonstrate significant improvements compared to the normal MFS method on all four datasets.

Multiclass classifiers (that categorize objects into more than two specific classes) are important tools since they are widely applied to machine vision and pattern recogni-

tion applications. Over the last decade, SVM has shown impressive accuracy in resolving both linear and nonlinear problems by maximizing the margin between classes [Suykens and Vandewalle, 1999]. Although SVM was originally designed for a binary task, additional mechanisms can create a multiclass SVM by decomposing it into several binary problems such as One-vs-Rest (OvR) and One-versus-One (OvO) [Platt et al., 2000]. A precise definition of the algorithm is given in the next section.

Multiclass SVM is often treated as a black-box within more complicated applications, such as object recognition ([Hsu and Lin, 2002, Gehler and Nowozin, 2009]) and bioinformatics ([Guyon et al., 2002, Furey et al., 2000]) and text classification ([Forman, 2003, Tong and Koller, 2002]), which hides the process that the multiclass SVM generates results by using a group of assembled binary classifiers. In practice, feature selection is necessary for applications that have an abundant number of features. It not only eliminates redundant features to reduce computation and storage requirements, but also chooses appropriate feature subsets that improve the prediction accuracy. [Guyon and Elisseeff, 2003] categorizes the feature selection methods into three types: filter, wrapper and embedded. The filtering method evaluates the correlation of every feature and ranks them by their coefficients, so the selection algorithm chooses new features that have lower correlations to the existing features. The wrapper method, which tests the prediction power of single features, investigates the independent usefulness of features and the selection strategy is according to the order of power. The embedded method integrates both feature selection and training. It selects features while building the model. Figure 6.1 illustrates a typical example of the feature selection performance on a multiclass application. Firstly, the classification performance increases as more features are selected, because more features provide more discriminative ability in the feature space. After the number of selected feature reaches 15, the accuracy score fluctuates near a specific level. Then the score starts to drop due to redundancy and over-fitting when more than 30 features are selected.

Normally, the Multiclass Feature Selection (MFS) procedure is applied to the black box of multiclass SVM, and it selects the same feature subset for every binary classifier to maximize the average accuracy over all classes [Shieh and Yang, 2008, Saeys et al., 2007, Chen et al., 2006]. Here we investigate the sequence of feature selection and constructing a multi-class SVM. We propose that an Individual Feature Selection (IFS) procedure can be directly exploited to the binary OvO SVMs before assembling the full multiclass SVM. Given samples from every pair of classes, the selected subset

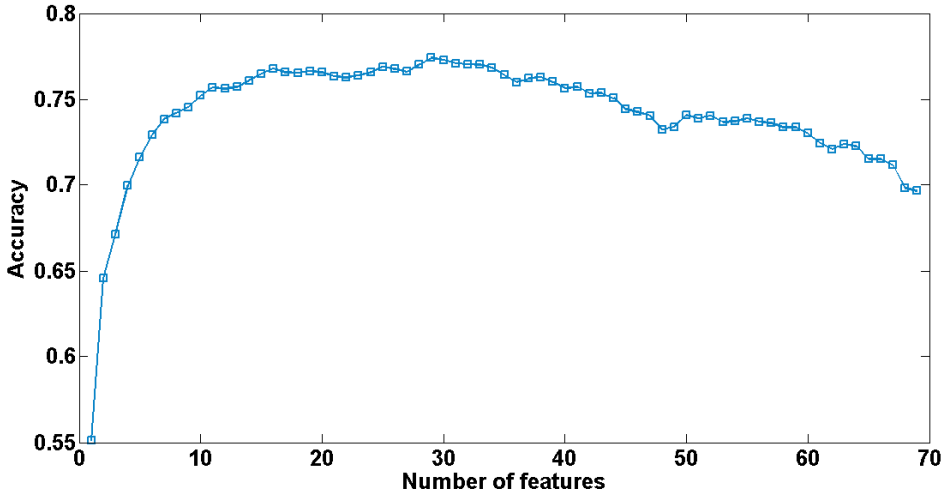


Figure 6.1: An example of the feature selection result in a multiclass application. The accuracy score increases in the beginning but it drops after 30 feature are selected. This example indicates that feature selection reduces the size of the feature space and also improves the accuracy by choosing an appropriate feature subset, instead of using all features.

of features maximizes the accuracy of classifying these classes. After then, we use these optimized OvO SVMs to construct a multi-class classifier. One can hypothesize that the classification performance would be better under the second scheme because each vote is now optimized to discriminate between two specific classes. The experimental results show that this small change to the normal multiclass SVM significantly improves performance with a decreased computing cost.

In this chapter, we propose a novel practical mechanism that applies individual feature selection to each binary OvO SVM, called IFS-SVM. After forward sequential feature selection and training each SVM model, IFS-SVM classifies each test sample by counting votes for each specific class and selects the class with most votes. The proposed method is evaluated on four different datasets to compare the performance and computing time. We note that other feature selection and vote combination methods could be used. This thesis only addresses the issue of when to do the feature selection. The rest of the chapter is organized as follows: Section 2 introduces the multiclass SVM with OvO strategy. Section 3 describes individual feature selection for multiclass SVM. Section 4 shows experimental results of four datasets: two underwater fish image datasets, the Oxford flower dataset and a skin lesion image dataset.

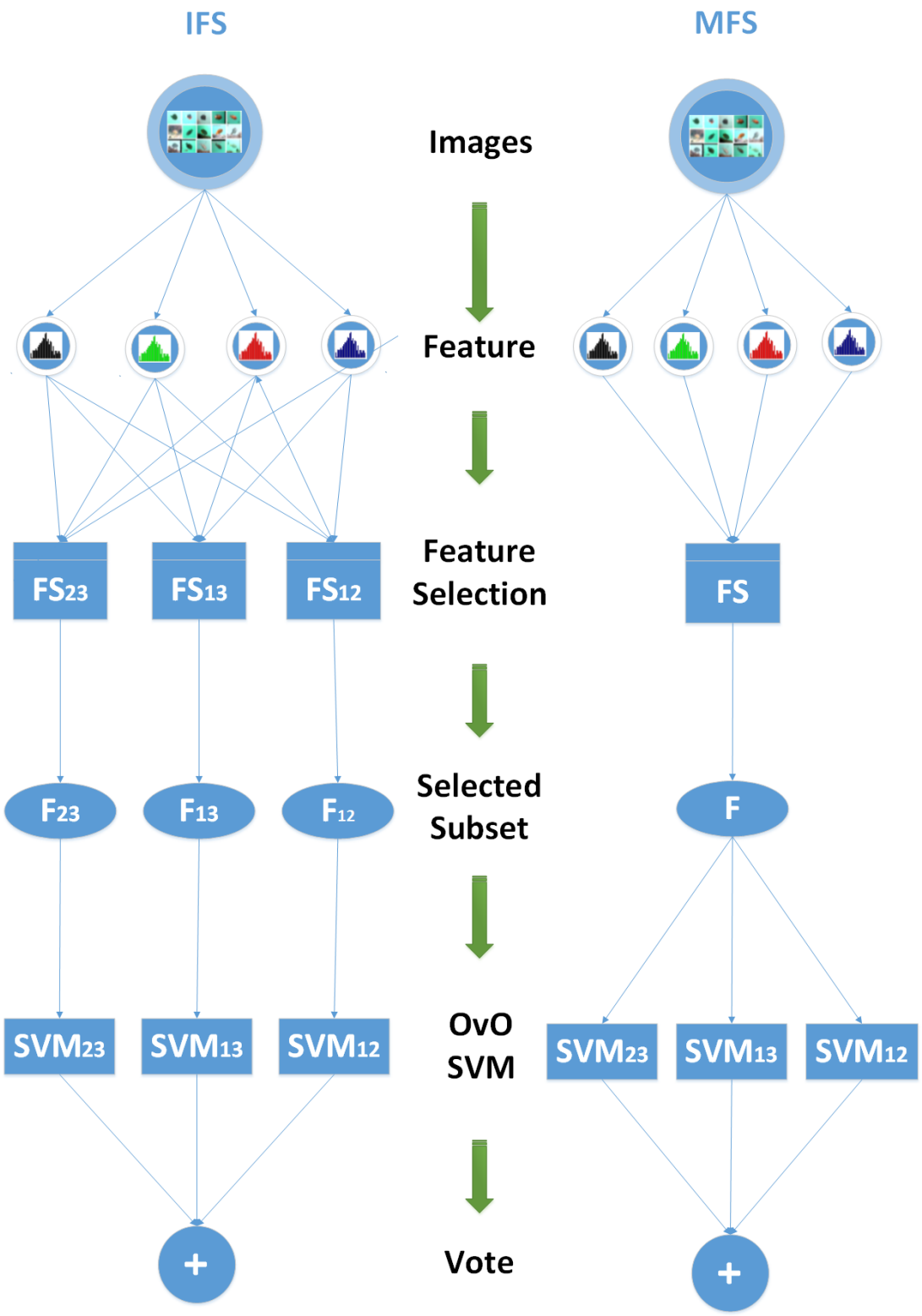


Figure 6.2: Comparing the workflows of MFS and IFS. We choose an example that classifies three classes so the final prediction is calculated by voting from three OvO SVMs. In the second column, the MFS method selects the same subset of features for all binary OvO SVMs while the IFS method chooses an individual feature subset for each OvO classifier.

6.1 Multiclass SVM with OvO strategy

Given a training set $\mathcal{D}$ from p classes, which is a set of n sample points of the form:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{1, ..., p\}\}_{i=1}^{n} \tag{6.1}$$

y_i indicates the class label of m-dimensional feature vector $\mathbf{x}_i$. Considering the two-class task (p = 2), the maximum margin classifier, a Support Vector Machine (SVM) [Cortes and Vapnik, 1995], is optimized to find a hyperplane, called maximum-margin hyperplane, which maximizes the margin between the two classes. A binary SVM minimizes

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ & s.t. \quad y_i(\mathbf{x}_i \cdot \mathbf{w}^T + b) \geq 1 - \xi_i \quad and \quad \xi_i \geq 0 \quad \forall y_i \in \{-1, 1\} \end{aligned} \tag{6.2}$$

where $\mathbf{w}$ is the normal vector to the hyperplane, b is the bias. This equation could be transformed into a convex quadratic programming optimization problem, and $\mathbf{w}$,b could be calculated by a Quadratic Programming solver.

A multiclass classification task can be decomposed into a set of two-class problems where the binary SVMs are applicable. One strategy is to train p One-versus-Rest (OvR) classifiers and they are used to classify one class from all the other classes. The final classification is determined by the highest score (winner-takes-all). The second strategy pairs each two of the classes and trains an SVM classifier for each pair, named as One-versus-One (OvO) strategy. Each binary classifier is trained on only two classes, thus the method constructs $p*(p-1)/2$ binary OvO SVMs. These binary classifiers process the test sample, and the winning class is added a vote. The class with the most votes determines the final prediction. Both strategies are widely used and have their own pros and cons. OvR uses fewer binary classifiers and the training cost is linear with p but it is criticized for no bound on the generalization error [Platt et al., 2000] and resolving potentially asymmetric problems using a symmetric approach [Li et al., 2004]. OvO is easy to train because each classifier only resolves a binary classification problem with two classes, but the computation cost is bigger since the number of binary classifiers grows as $p*(p-1)/2$.

6.2 Individual feature selection for binary OvO-SVMs

After constructing the multiclass SVM using the OvO strategy, the Multiclass Feature Selection (MFS) method chooses a subset of features by either filtering features according to their correlation coefficients or wrapping them in proportion to their usefulness to a given SVM predictor [Guyon and Elisseeff, 2003]. In contrast to the MFS criteria that treats the multi-class SVM as a black-box and selects features such that all binary classifiers use the same subset of features, our proposed work investigates applying feature selection to each binary classifier individually so that each OvO vote is optimized. An example of comparing the different workflows of MFS and IFS is shown in Figure 6.2. Both methods use the same forward sequential feature selection algorithm. The complete proposed training procedure is described as follows:

(1) For every two classes i, j ($i, j \in \{1, ..., p\}$ and $i \neq j$), start with an empty feature set $\tilde{F}_{ij} = \phi$ and m features $\{f_t\} = F$. The evaluation function is named as E.
(2) Repeat until all features are evaluated, step $s \in \{1, ..., m\}$:

- select every $\{f_t\} \in F$ and evaluate $e_{s,t} = E([\tilde{F}_{ij}, f_t])$
- choose the maximum of all evaluations $\tilde{e}_s = \arg\max_t e_{s,t}$, record $\tilde{e}_s$.
- add the corresponding feature $\tilde{f}_s$ to the feature set $\tilde{F}_{ij}$ as the selected feature of step s: $\tilde{F}_{ij} = E([\tilde{F}_{ij}, \tilde{f}_s])$.
- remove the feature $\tilde{f}_s$ from the feature pool F: $F = F$ - $[\tilde{f}_s]$.

(3) Choose the feature subset $F_{ij} = [\tilde{f}_1, ..., \tilde{f}_{\tilde{s}}]$ that produce the highest evaluation score for each i, j, where $\tilde{s} = \arg\max_s \tilde{e}_s$. Note: other stopping criteria could be used.

After feature selection, these binary SVMs are trained using their corresponding feature subsets $\tilde{F}_{ij}$ on the training samples. In the evaluation step, binary SVMs also extract the $\tilde{F}_{ij}$ features of the test samples, and they vote for the final prediction. It is reasonable to assume that each vote is optimized so the prediction is more accurate.

One concern is the computational complexity. But given the assumption that the computing time of classification only depends on the number of features, we can show that our proposed method (IFS-SVM) requires no more computing time in feature selection than the common MFS method (both using the forward sequential feature selection algorithm):

Assumption 1: The computation time of a binary classifier only depends on the number of input features, *i.e.* $f(D_{m*n}) = f(m,n)$ where function f is the computation time, D_{m*n} is the input features, m is the number of samples, n is the number of features.

This assumption eliminates nonessential details so we can focus on comparing the time cost itself. The computation time of feature selection using MFS is:

$$T_{MFS} = \sum_{n=1}^{\tilde{N}} [(N-n+1) * (T_v(c) + \sum_{i \neq j \& i,j \leq c} f(M_i + M_j, n))] \tag{6.3}$$

where M_i is the number of samples from class i, $i \in \{1,2,...,c\}$, c is the number of classes, N is the number of input features F and $\tilde{N}$ is the number of features to select, T_v is the computing time of voting. The computation time of feature selection of our proposed IFS method is:

$$\begin{aligned} T_{IFS} &= \sum_{i \neq j \& i,j \leq c} \sum_{n=1}^{\tilde{N}} [(N-n+1) * f(M_i + M_j, n)] \\ &= \sum_{n=1}^{\tilde{N}} [(N-n+1) * \sum_{i \neq j \& i,j \leq c} f(M_i + M_j, n)] \leq T_{MFS} \end{aligned} \tag{6.4}$$

Although the IFS-SVM method conducts p^2 times individual feature selections, the size of samples in each individual one is decreased to $2/p$ (two out of p classes). Thus the computing complexity is still $O(p^2)$. On the other hand, equations 6.3 and 6.4 show that the IFS-SVM method avoids the voting procedure when selecting features. We have conducted experiments on four datasets to compare the computation time of both methods, as shown in Figure 6.5. This experiment varies the number of classes p and records the computing time of feature selection as describe in Section 6.1. Both curves fluctuate since the number of selected features may vary from different number of classes. The general trend indicates that the proposed method (IFS-SVM) spends less time for training than the MFS method. See experimental section for more details.

6.3 Experimental evaluation

We test both feature selection mechanisms on four datasets using cross validation. The binary OvO SVM classifier is implemented by LIBSVM [Chih-Chung and Chih-Jen, 2011].

We use the same forward sequential feature selection for all tests so the results are comparable. All experiments are programming in Matlab. The code is compiled and deployed on a cluster of machines. The performance is evaluated by Average Recall (AR), Average Precision (AP) and Accuracy over Count (AC). AR and AP describe the recall/precision that are averaged over all classes so the minority classes have equal importance to the major ones. AC is the accuracy over all samples, and it is defined as the proportion of correctly classified samples among all samples. These scores illustrate a comprehensive analysis of the experimental results regardless of whether the dataset is balanced or not. In each experiment, we compare AR/AP/AC scores of three methods: multiclass SVM without feature selection (M-SVM), multiclass feature selection for SVM (MFS-SVM), individual feature selection for multiclass SVM (IFS-SVM).

6.3.1 Underwater fish image dataset

The fish images are acquired from underwater cameras placed in the Taiwan sea with 24150 fish images (Fish24K dataset) of the top 15 most common species [Boom et al., 2012]. The training and testing sets are isolated so fish images from the same trajectory sequence are not used during both training and testing. We use the same method of feature extraction as in [Huang et al., 2012]. These features are combinations of 69 types (2626 dimensions) including colour, shape and texture properties in different parts of the fish such as tail/head/top/bottom, as well as the whole fish. All features are normalized by subtracting the mean and dividing by the standard deviation (z-score normalized after 5% outlier removal).

The classification results after feature selection with 5 fold cross-validation are shown in Table 6.1. This dataset is very imbalanced, thus the averaged recall and precision are lower than the accuracy over all samples. The first row shows the result of multiclass SVM using all features, where the averaged recall (AR) is increased after the feature selection with the cost of reduced AP and AC (the second row). In the third row, individual feature selection (IFS-SVM) improves the classification performance in all three measures.

The Fish24K dataset is so imbalanced that the samples of the most common species are 500 times larger than the samples of the least common species. We conduct another experiment on a similar dataset of 6874 fish images (Fish7K dataset) to evaluate the performance when the dataset is less imbalanced. The result is shown in Table 6.2. The

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	76.9 ± 4.0	88.5 ± 3.6	95.7 ± 0.5
MFS-SVM	79.0 ± 3.6	86.4 ± 5.3	95.3 ± 0.3
IFS-SVM	$\mathbf{81.6 \pm 4.7}$	$\mathbf{90.9 \pm 5.0}$	$\mathbf{96.4 \pm 0.5}$*

Table 6.1: Experiment results on the whole fish image dataset, all results are averaged by 5-fold cross-validation. * means significant improvement with 95% confidence.

MFS method reduces the feature dimensions with the cost of slightly decreasing the performance, while the proposed IFS method significantly improves the performance.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	72.6 ± 6.1	77.7 ± 3.3	93.2 ± 0.9
MFS-SVM	72.3 ± 8.8	77.5 ± 7.4	92.9 ± 1.1
IFS-SVM	$\mathbf{80.2 \pm 3.0}$	$\mathbf{89.8 \pm 5.4}$*	$\mathbf{94.9 \pm 1.3}$*

Table 6.2: Experiment results on more balanced fish dataset of 6874 images, all results are averaged by 5-fold cross-validation. * means significant improvement with 95% confidence.

6.3.2 Oxford flower dataset

The Oxford flower dataset [Nilsback and Zisserman, 2007] consists of 13 categories (753 segmented flower images) of common flowers in the UK (Figure 6.3). We exploit the segmentation results and use the same features as described in the previous section. The whole dataset is split into three parts for cross-validation. Half of the images are used for training while the validation and test set divide the remaining images equally.

As shown in Table 6.3, feature selection improves the classification accuracy, while the proposed method (IFS-SVM) achieves the highest performance. In this experiment, AR, AP and AC scores are close since this dataset is more balanced. Other features and machine learning methods might achieve better results. However, we only introduced the improvement of using forward sequential method with a linear SVM, so the result focuses on the variations introduced by different feature selection methods.

Figure 6.3: Flower dataset of 13 common categories in the UK. This task is difficult because the images have large scale, pose and light variations. Some classes are quite similar to others and they both have enormous variations.

6.3.3 Medical image dataset

The third dataset is consists of 1300 medical images of skin lesions, belonging to 10 classes [Ballerini et al., 2012]. 17079 dimensions of colour and texture features are extracted and normalized to zero mean and unit variance. PCA is used for feature reduction which preserves the top 98% energy of components' coefficients. It reduces the dimension of features to 197 but loses about 9% accuracy (from 76% to 67%). The result in Table 6.4 demonstrates improvements for both feature selection methods (MFS and IFS). The proposed IFS method is significantly better than the other two methods for all three evaluation criteria with 5-fold cross-validation.

6.3.4 Experiment overview

In Figure 6.4, we give an overview of the performance of the three methods when the number of classes changes. The first row shows the results of the Fish24K dataset. AR, AP and AC (first three columns) are all decreasing as the number of classes increases.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	76.6 ± 3.7	78.0 ± 3.5	77.7 ± 3.6
MFS-SVM	81.4 ± 2.2	83.5 ± 2.9	83.3 ± 1.9
IFSSVM	$\mathbf{82.8 \pm 1.4}$	$\mathbf{85.5 \pm 0.2}$	$\mathbf{83.8 \pm 1.6}$

Table 6.3: Experiment results on flower dataset. All results are averaged by 3-fold cross-validation.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	58.8 ± 2.5	66.2 ± 3.3	66.9 ± 2.9
MFS-SVM	61.8 ± 4.0	64.4 ± 5.1	70.2 ± 2.9
IFS-SVM	$\mathbf{73.0 \pm 5.0}$*	$\mathbf{76.3 \pm 4.0}$*	$\mathbf{77.0 \pm 3.2}$*

Table 6.4: Experiment results on skin image dataset. All results are averaged by 5-fold cross-validation. * means significant improvement with 95% confidence.

The MFS method (red line) is sometimes worse than the baseline M-SVM method (black line) due to over-fitting. It achieves significant improvement in the validation set, but the performance drops when it is generalized to the test set. The same trend is also observed in the following experiments: the Fish7K dataset, the Oxford flower dataset, the skin lesions dataset. Our proposed IFS method (blue line) outperforms the other two methods and achieves higher performance in all experiments. Figure 6.5 shows the computing time of feature selection, which illustrates that the IFS method reduces the time cost while having superior accuracy.

6.3.5 Optimization in computing time

LIBSVM provides its own implementation of multi-class SVM that also uses the OvO strategy. In our experiment here, we use the multiclass LIBSVM, instead of using its binary SVM utility and wrapping to a multiclass SVM in Matlab (MFS-SVM), to process the same forward sequential feature selection method on the datasets. The results are listed in Table 6.5, comparing to the computing time of MFS and IFS methods.

IFS-SVM is faster in the Fish24K dataset because it only selects features for two classes so the size of the feature subset is smaller, while the other two methods have to choose more features to balance the accuracy over all classes. This factor becomes

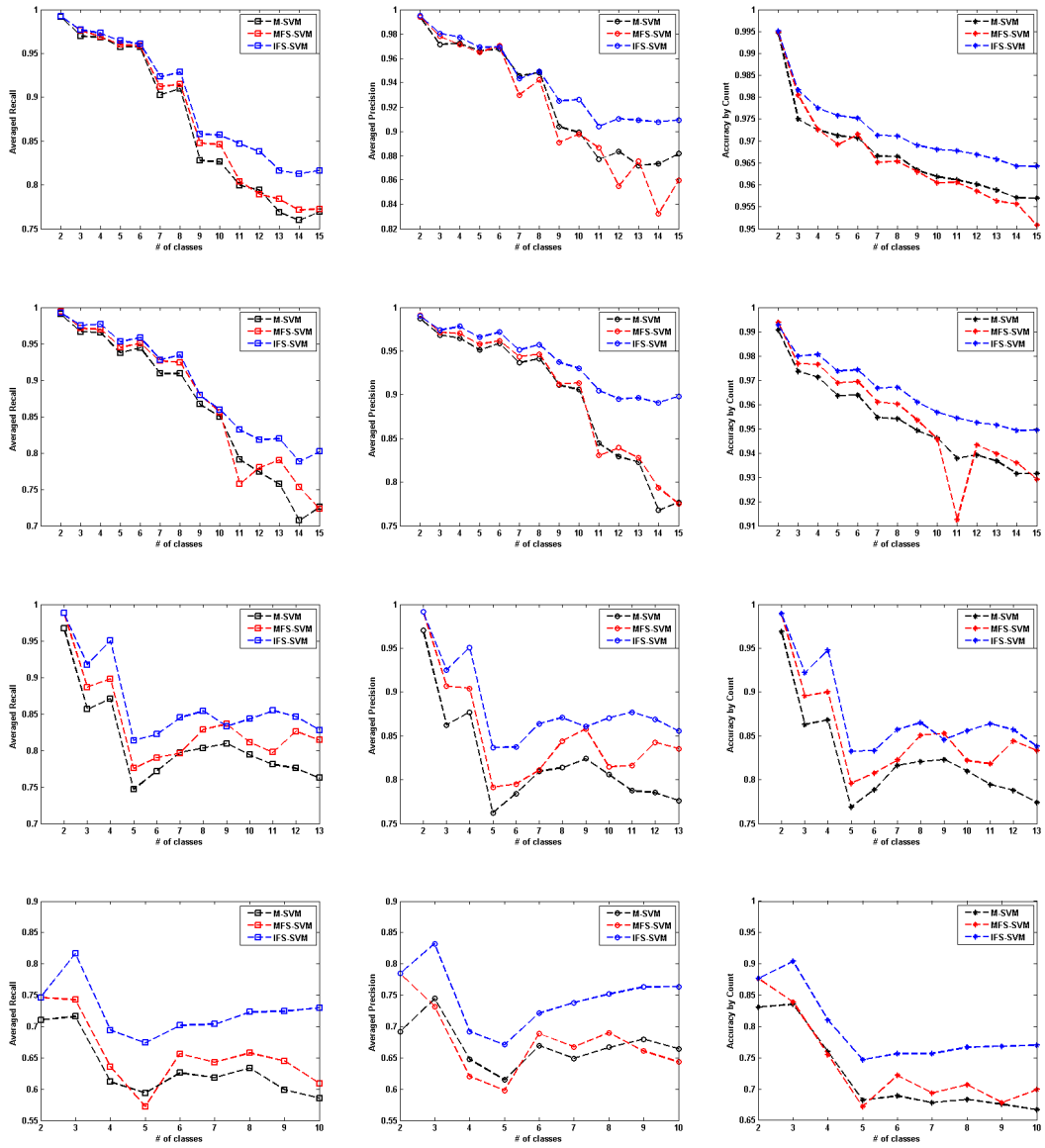


Figure 6.4: Performance overview comparing the three methods as the number of classes increases. From left to right: Averaged Recall, Averaged Precision, Accuracy by Count. From top to bottom: the Fish24K dataset (24150 images), the Fish7K dataset (6874 images), the Oxford flower dataset (753 images), and the skin lesions dataset (1300 images). Note, in the result on Oxford flower dataset, MFS-SVM performs slightly better than the IFS-SVM when classifying 9 classes.

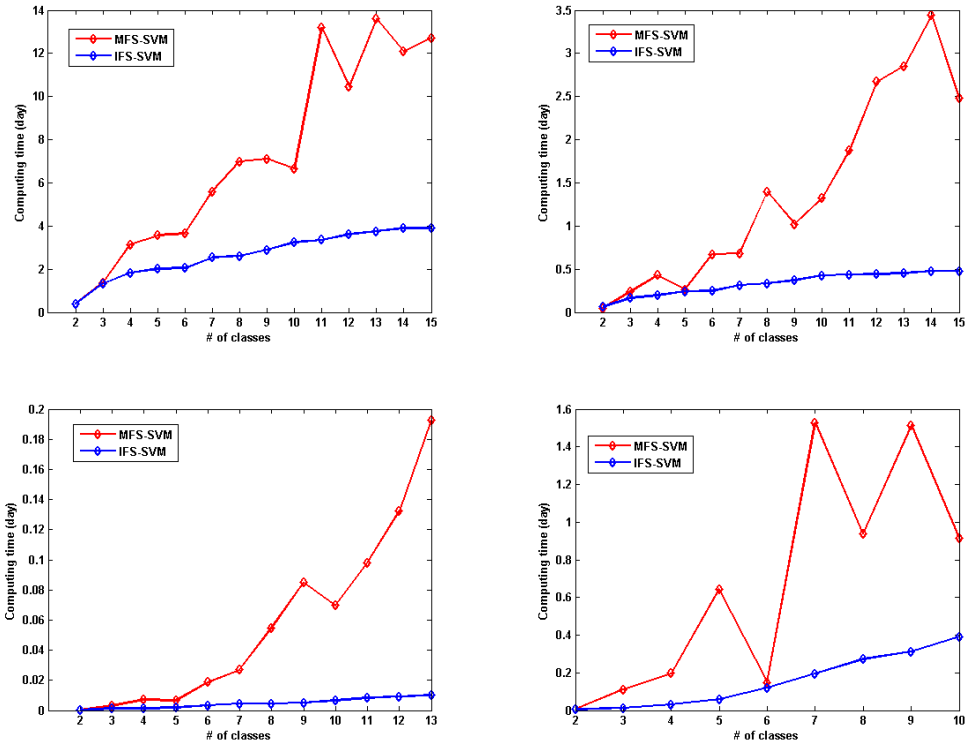


Figure 6.5: Computing time (training) of three methods as the number of classes increases. From left to right, row one: the Fish24K dataset (24150 images), the Fish7K dataset (6874 images); row two: the Oxford flower dataset (753 images), and the skin lesions dataset (1300 images).

more significant when the dataset is large. The LIBSVM method spends less computing time than IFS-SVM in the other three experiments. The LIBSVM uses the same procedure as MFS-SVM, but it is more efficient since it implements the multiclass SVM in C++. This experiment also provides an estimate of the potential optimization (2-50x improvement) of the IFS method if it were re-implemented in C++.

6.4 Conclusion

In this chapter, we showed that individual feature selection in each one-versus-one classifier improves the performance of multiclass SVM. This method could be adapted into any multiclass classifier that is constructed by assembling binary classifiers. We tested the proposed method on four different datasets, comparing to the multiclass SVM with forward sequential feature selection. The results demonstrate a significant

method	Fish24K	Fish7K	Flower	Skin
MFS-SVM	14.34	2.48	0.19	0.92
LIBSVM	5.57	0.24	2.73e-3	0.18
IFS-SVM	3.90	0.48	0.01	0.39

Table 6.5: Computing time comparison. The experiment used the datasets described above. The LIBSVM method uses the same OvO strategy as MFS-SVM but is optimized. Thus it provides an estimate of the potential optimization of our proposed method.

improvement on all experiments. We also compare the computing time and show the proposed method is more efficient than the normal feature selection mechanism.

Chapter 7

Conclusions

This last chapter states the conclusions of the work presented in this thesis, which includes the novel contributions and experimental achievements. Unsolved issues and potential future investigations that come from this work are then discussed.

Live fish recognition in the open sea has been investigated to help understand the marine ecosystem, which is vital for studying the marine environments and promoting commercial applications. This recognition task is fundamentally challenging because of its complex situation where the illumination changes frequently. Prior research is mainly restricted to constrained environments (fish in the tank or on a conveyor system) or dead fish, and these machine vision systems have only explored applications for a limited number of fish species. These methods perform worse when they deal with unconstrained fish in a real-world underwater environment, especially when the dataset is greatly imbalanced.

In contrast, our work investigates novel techniques to perform effective live fish recognition in an unrestricted natural environment and presents an application of machine vision and learning for free swimming fish. This so-called Balance-Guaranteed Optimized Tree with Reject option (BGOTR) system adopts a hierarchical classification that is based on inter-class similarities to improve the normal hierarchical method and to integrate computer vision techniques and marine biological knowledge. Multiclass classifier and feature selection are built together into a hierarchical tree and optimized to maximize the classification accuracy of grouped classes. BGOTR exploits a novel rejection mechanism to re-classify samples that tend to be confusable with other classes. Meanwhile, trajectory voting combines temporal information with the classi-

fication results so that majority results of the same species are preserved while potential outliers produced by occasional illumination changes or fish postures are eliminated. Conflicting decisions resulting from several confusable species are effectively dealt with by voting using each fish detection that appears in multiple frames of a video shot. The reject option after hierarchical classification is conducted by applying the Gaussian Mixture Model (GMM) method to model the feature distribution of the training images. Low confidence decisions of test samples are rejected so that a substantial proportion of classification errors and new species are thrown out although a small number of correctly recognized fish are also removed due to incorrect rejection. A novel practical mechanism that applies individual feature selection to the binary One-versus-One SVM, called IFS-SVM, is presented. After forward sequential feature selection and training each SVM, IFS-SVM classifies each test sample by counting votes that are optimized for every pair of specific classes. Tested on a manually labelled fish dataset of 24150 images, which is the largest and most varied dataset used for fish species recognition, BGOTR demonstrates better accuracy averaged both by all images and by all classes, compared with other previous research. This is the first time that the hierarchical classification method with reject option has been implemented in a live fish recognition system.

The rest of this chapter summarizes the novel contributions of the Balance-Guaranteed Optimized Tree with Reject option and then discusses future work that is extended from existing results.

7.1 Contributions

The following paragraphs describe the novel contributions that distinguish the proposed BGOTR system from prior fish recognition studies:

- **New and more effective classification method for free swimming fish, in Chapter 4**
  We introduce the BGOTR framework and trajectory voting method to identify the top common species to extend the fish recognition works to free swimming fish in an unrestricted natural environment. Our Balance-Guaranteed Optimized Tree with Reject option (BGOTR) is the first machine vision application to integrate these methods. We show BGOTR has a higher classification accuracy than

common alternative classifiers.

- **New features suitable for fish classification, in Chapter 3**
  The thesis also introduces several new types of fish descriptors that are shown to be effective and invariant to environmental changes. These features are a combination of colour, shape and texture properties in different parts of fish such as tail, head, top, bottom and the whole fish. They are designed to integrate domain knowledge with machine vision methods and considered altogether in the pool for feature selection in the classification step. A novel streamline method is implemented to align the fish images in the same direction before further processing. In the feature selection step, our idiosyncratic fish features and the texture features are most frequently selected by FSFS. Half of the tree nodes select REHIST normalized colour features and fish boundary descriptors for use.

- **A trajectory voting strategy to exploit temporal information for refining classification results, in Section 4.2.4**
  Single image classification together with fish tracking results has not been investigated in previous machine vision applications. We give the first implementation of this algorithm in the hierarchical fish recognition framework BGOTR. The low quality of video images greatly limits the fish recognition accuracy. As each fish appears in multiple frames from a video shot, we use the trajectory analysis to combine the results from all frames, and use the winner-take-all strategy to accept the majority decision. The challenging task of live fish recognition produces high variability in the fish images. Trajectory voting operates over multiple frames to combine possible conflicting decisions about confusable species. By employing this voting approach, potential outliers produced by occasional illumination changes or fish postures are also eliminated.

- **A classification-rejection method to clear up decisions and reject unknown classes, in Chapter 5**
  The reject option is designed to clear up decisions and reject unknown classes. With the reject function after hierarchical classification, BGOTR is a new and more effective approach to suppress the error accumulation problem of the hierarchical method and to eliminate false detections as well as samples from unknown classes. A Gaussian Mixture Model and Bayesian *posterior* probability are the basis of the reject option after the hierarchical classification. It evaluates the recognition result and calculates the probability of being a certain

species to filter less confident decisions. The experimental results obtained from a manually labelled fish dataset show a lower false positive rate since some misclassification errors can be overcome but at the price of a slightly lower true positive rate due to incorrect rejections.

- **An individual feature selection mechanism for optimizing One-versus-One SVM performance, in Chapter 6**
  Previous studies treat the Multiclass One-versus-One (OvO) SVM, which is constructed by assembling a group of binary classifiers, as a black-box. We propose a novel mechanism where an Individual Feature Selection (IFS) procedure can be directly applied to binary One-versus-One SVM before assembling the full multiclass SVM. The IFS method selects different subsets of features for each One-versus-One SVM inside the multiclass classifier so that each vote is believed to be optimized for better discriminating the two specific classes. The proposed IFS method is tested on four different datasets for comparing the performance and time cost. Experimental results demonstrate significant improvements compared to the normal MFS method on all datasets.

7.2 Future work

The work presented in this thesis presents a higher performance fish recognition algorithm for free swimming fish in an unrestricted natural environment. However, there are still many potential extensions related to our presented studies. Some of the improvements are discussed in the relevant chapters. We highlight some interesting topics of the research below.

- **Alternative multiclass classifier**
  We used the multiclass SVM classifier. An alternative classifier is a weighted K-Nearest Neighbour (KNN) model. According to the distance between positive and negative samples, each of the feature vectors is assigned a weight to express its reliability. In the classification step, the weighted KNN classifier is applied to the whole data set. This method is combined with the hierarchical tree, and a new fish sample is classified as the same class as its nearest labelled data. The KNN classifier is more efficient for datasets of a significant size. In contrast to the normal “hard assign” classification, a “soft assign” algorithm can

be employed, where the classification results consist of the similarity scores to all species. It also generates a result vector to identify to which category the data sample belongs. But instead of the binary value produced by the "hard assignment" strategy, each classification is calculated as a similarity in the "soft assign" mechanism. All members of the clustering group vote for species by adding their similarities to the result histogram which determines the final result. The highest score in the histogram decides the class label. This winner-take-all strategy reduces the effect of noise and hopefully overcomes the limitations of a single noisy data classification. We did explore the use of random forest classifiers, but they gave worse results. A kernel SVM or Bayesian logistical regression (or other multinomial logistic regression) could also be considered as a future classification technology substituting for an SVM. Alternatively, the fish recognition application on dataset of better quality images could consider a deformable shape model that incorporates two steps. The first step is a coarse description which shows the entire fish's shape. In this step, the fish is represented with global features such as contour, shape, colour histogram and so on. The second step illustrates different parts such as the head, tail and fin. These parts are represented by using local features like Histogram of Oriented Gradient (HOG) and Scale-Invariant Feature Transform (SIFT). Furthermore, as the fish changes its distance from the camera and alters the shape size, the deformable model can be trained at different resolutions.

- **Alternative hierarchical classification tree**

  The hierarchical classification method could benefit from speeding up the construction of the hierarchical classification tree, because the procedure of choosing the best split has to exhaustively search all of the possible combinations, which is time-consuming and not affordable when number of classes grows to be huge, especially considering the splits into multiple branches. In a large scale database, the taxonomical representation helps describe the hierarchical structure of fish species. The taxonomy ontology is an academic subject which aims to construct a scientific methodology to systematize animals into their hierarchical categories. This methodology is based on the synapomorphies characteristic from both fossil and the extent to which the taxon is monophyletic. The representation indicates the distinction between species, *e.g.*, the presence or absence of components, specific numbers and particular shapes. A possible way to improve

the hierarchical classification algorithm with pre-defined taxonomic structure to exploit the knowledge of the biologists on a huge database which contains a large number of fish images. We could also extend the fish species recognition method to include fish component distribution (*e.g.*, head, tail and fin) which might improve the robustness in distorted environments. In our BGOT system, we only consider fish as a whole object. However, some video frames are not sufficient to determine the fish species. Some components, as well as the global fish features, may not be observable in a single frame. To improve the recognition accuracy under such circumstances, the component distribution model acts as a partial recognition system based on the tracking result. By giving evidence of component C_i, the fish species might be estimated as:

$$S = \underset{S}{\operatorname{argmax}} \sum_{i=0}^{N} w_i log(p(C_i = \alpha_i^S \mid V)) \tag{7.1}$$

where w_i indicates the weight of each component, C_i is model type of component i, α_i^S is the type of component i for species S, V is the visual evidence. By using Bayes Rule, the formula is transformed to:

$$S = \underset{S}{\operatorname{argmax}} \sum_{i=0}^{N} w_i log(\frac{p(V \mid C_i = \alpha_i^S) p(\alpha_i^S)}{p(V)}) \tag{7.2}$$

where $p(\alpha_i^S)$ is the *a priori* probability of this specific component i shape over all classes, $p(V \mid C_i = \alpha_i^S)$ is the statistical result of evidence for a given component type, $p(V_i)$ is constant over all classes S and is omitted. A fish is species S if it has the highest probability of having the right component C_i types for all N components. This component-based evidence can also be combined with the integrated evidence used above.

- **Learning to predict trajectory set**
  In our experiments, the improvement arising from trajectory-based integration shows that using temporal information improves individual classification based on single frames. The advantage is that mistakes arising from random noise or various pose/orientations can be eliminated if the majority of samples are correctly recognized. The trajectory recognition method used in Chapter 4 uses a majority vote (*i.e.* takes the class of the most votes as the final prediction). The winner-take-all strategy is performed separately for each tracked fish. The trajectory based recognition result could be computed from the results of any

individual classifier after a certain number of frames, wherever a set of observations is tracked and recorded. We have presented this mechanism as an post-recognition result refinement strategy. One can also consider a before-matching process that uses the trajectory data for probability inference, *i.e.* use multiple observations to determine the similarity between image sets, before matching. This is true of any method of image set classification that involves more than one frame of grouped input data, *e.g.* joint distributions, second-order statistical representations (covariance matrix), subspaces (Mutual Subspace Method, MSM), manifold method and affine/convex hull methods. With the consideration of Chapter 4, future work could assign a *posterior* probability P(classification | all trajectory data), given prior knowledge that the observations of the same fish are replicated over N tracked frames.

- **Investigation of feature selection approaches**
  The feature selection algorithm described in the previous chapters uses forward sequential feature selection based on grouped subsets of features. Recent work such as [Tibshirani, 1996] introduces a shrinkage and selection method for linear regression (Lasso). Similar to the soft-thresholding method based on wavelet coefficients, the Lasso algorithm minimizes the sum of squared errors. We implement this method on the live fish dataset, and it selected 760 out of the 2626 features. But the results when using our group-based feature selection method were better than the Lasso method. Other feature selection methods including filter, wrapper and embedded methods could also be examined in the future. One might also look at methods of pruning membership in each feature group.

- **Computer-assisted labelling**
  The Fish4Knowledge project is concerned with a significant video data quantity, with more than 10E+9 fish detections. It is a tough job to label ground-truth data using marine biologists in the traditional way. We have introduced a clustering-based annotation process for labelling fish images. In the future, perhaps unlabelled data can be processed as data with incomplete labelling, *e.g.*, semi-supervised learning, which aims to effectively apply the small amount of labelled data to a huge amount of unlabelled samples. After the clustering and the group labelling, the knowledge from labelled samples can be applied to the unlabelled group while the unlabelled data's features may also enhance classification performance. Alternatively, the annotation work can be applied to a whole

set of clustered samples. Labelling every individual is a difficult mission, but to label the group (*e.g.*, fish from the same trajectory) is faster by contrast. Further investigation into methods to integrate the BGOTR system and the labelling tool will be very useful since the fish recognition system provides a coarse labelling result where sets of manual annotations only require reviewing. Labelling samples rejected by BGOTR will also be an exciting topic for machine vision and marine biology research, under the assumption that the rejected observations will contain a higher proportion of new species.

- **Image quality enhancement**
  We have not implemented any algorithm to improve the quality of underwater images. Enhancement of underwater image quality includes image restoration from the water impurity, colour correction below the water surface, and image enhancement by applying super-resolution methods. Firstly, image distortion is always irregular and complex, and distortion factors include two aspects. One is the turbidity affected by impurity and floating particles. The other one is the light propagation properties by the media attributes. The image distortion model formulates the observed image as a degradation function convoluted with the original image adding noise. The system response function can be described as the combination of optical transfer function and the modulation transfer function. The restored image is de-convoluted by the de-convolution model. Secondly, colour correction in underwater video is more complex, the strengths of light decrease as depth growth according to their wavelengths. The colour correction model generates a comparable underwater scene which is beneficial to fish recognition by considering the spatial variations and by achieving proper colour compensation as the distance increases. A possible method of colour correction is to employ the prior knowledge of some objects (sand for example), so the colour correction model can be estimated by calculating the distortion rate of three primary light components. In our underwater frames, there are many background objects which can be used to identify the parameters of the distortion model. The discussed colour correction method can be applied to restore the colour features while avoiding importing lighting issues. Thirdly, the image enhancement methodologies, such as Super Resolution (SR), could also be applied to improve the quality of underwater videos. There are two categories of SR methodologies: multi-frame SR and single-frame SR. In the underwater

environment, recorded videos are often affected by many distortion factors. The video quality can be improved by implementing the multiple-frame based super-resolution method and by reconstructing the fish details from distorted images. We consider the investigation work of image quality enhancement as a future work that could benefit both the quality of fish features and the performance of fish recognition system.

References

[Ballerini et al., 2012] Ballerini, L., Fisher, R., Aldridge, B., and Rees, J. (2012). Non-melanoma skin lesion classification using colour image data in a hierarchical k-NN classifier. In *9th IEEE International Symposium on Biomedical Imaging*, pages 358–361.

[Begg and Waldman, 1999] Begg, G. A. and Waldman, J. R. (1999). An holistic approach to fish stock identification. *Fisheries research*, 43(1):35–44.

[Belongie and Malik, 2000] Belongie, S. and Malik, J. (2000). Matching with shape contexts. In *IEEE Workshop on Content-based access of Image and Video-Libraries*, pages 20–26.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

[Benson et al., 2009] Benson, B., Cho, J., Goshorn, D., and Kastner, R. (2009). Field programmable gate array (FPGA) based fish detection using Haar classifiers. In *American Association of Underwater Sciences Symposium*, pages 160–167.

[Bishop, 1995] Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

[Blidberg, 2001] Blidberg, D. R. (2001). The development of autonomous underwater vehicles (AUVs); a brief summary. In *IEEE International Conference on Robotics and Automation*, volume 4.

[Boom and Fisher, 2011] Boom, B. and Fisher, R. (2011). Fish4knowledge deliverable d5.1 component interface and integration plan.

[Boom et al., 2012] Boom, B., Huang, P., He, J., and Fisher, R. B. (2012). Supporting ground-truth annotation of image datasets using clustering. In *Proceedings of 21st International Conference on Pattern Recognition (ICPR)*, pages 1542–1545.

[Bosch et al., 2007] Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 401–408, New York, NY, USA. ACM.

[Brehmer et al., 2006] Brehmer, P., Chi, T. D., and Mouillot, D. (2006). Amphidromous fish school migration revealed by combining fixed sonar monitoring (horizontal beaming) with fishing data. *Journal of Experimental Marine Biology and Ecology*, 334(1):139–150.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Caley et al., 1996] Caley, M. J., Carr, M. H., Hixon, M. A., Hughes, T. P., Jones, G. P., and Menge, B. A. (1996). Recruitment and the local dynamics of open marine populations. *Annual Review of Ecology and Systematics*, 27:477–500.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:679–698.

[Carlos and Alex, 2010] Carlos, S. and Alex, F. (2010). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

[Caseiro et al., 2013] Caseiro, R., Martins, P., Henriques, J. F., Leite, F. S., and Batista, J. (2013). Rolling Riemannian Manifolds to solve the multi-class classification problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41–48.

[Cevikalp and Triggs, 2010] Cevikalp, H. and Triggs, B. (2010). Face recognition based on image sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2567–2573.

[Chambah et al., 2003] Chambah, M., Semani, D., Renouf, A., Courtellemont, P., and Rizzi, A. (2003). Underwater color constancy: enhancement of automatic live fish recognition. In *Electronic Imaging*, pages 157–168.

[Chan et al., 1999] Chan, D., Hockaday, S., Tillett, R. D., and Ross, L. G. (1999). A trainable n-tuple pattern classifier and its application for monitoring fish underwater. In *Proceedings of Seventh International Conference on Image Processing And Its Applications*, volume 1, pages 255–259 vol.1.

[Chen et al., 2006] Chen, X.-W., Zeng, X., and van Alphen, D. (2006). Multi-class feature selection for texture classification. *Pattern Recognition Letters*, 27(14):1685–1691.

[Chib, 1995] Chib, S. (1995). Marginal Likelihood from the Gibbs Output. *Journal of the American Statistical Association*, 90(432):1313–1321.

[Chih-Chung and Chih-Jen, 2011] Chih-Chung, C. and Chih-Jen, L. (2011). LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.

[Cline and Edgington, 2010] Cline, D. E. and Edgington, D. R. (2010). A detection, tracking, and classification system for underwater images. *ICPR Workshop on Visual Observation and Analysis of Animal and Insect Behavior (VAIB), Istanbul*.

[Cootes et al., 2001] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[de Zeeuw et al., 2010] de Zeeuw, P., Pauwels, E., Ranguelova, E., Buonantony, D., and Eckert, S. (2010). Computer assisted photo identification of Dermochelys coriacea. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, pages 165–172.

[Deng et al., 2010] Deng, J., Berg, A. C., Li, K., and Fei-Fei, L. (2010). What does classifying more than 10,000 image categories tell us? In *Proceedings of the 11th European Conference on Computer Vision*, pages 71–84. Springer.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 248–255. IEEE.

[Duan and Keerthi, 2005] Duan, K.-B. and Keerthi, S. S. (2005). Which is the best multiclass SVM method? an empirical study. In *Proceedings of the 6th international conference on Multiple Classifier Systems*, MCS'05, pages 278–285. Springer-Verlag.

[Dunbabin et al., 2006] Dunbabin, M., Corke, P., Vasilescu, I., and Rus, D. (2006). Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2091–2098.

[Edgington et al., 2006] Edgington, D. R., Cline, D. E., Davis, D., Kerkez, I., and Mariette, J. (2006). Detecting, tracking and classifying animals in underwater video. In *OCEANS*, page 1–5.

[Everingham et al., 2009] Everingham, M., Sivic, J., and Zisserman, A. (2009). Taking the bite out of automated naming of characters in TV video. *Image and Vision Computing*, 27:545–559.

[Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.

[Figueiredo and Jain, 2002] Figueiredo, M. A. T. and Jain, A. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396.

[Flusser et al., 2009] Flusser, J., Chi, T. D., and Zitov, B. (2009). *Moments and Moment Invariants in Pattern Recognition*. John Wiley & Sons, Ltd, Chichester, UK.

[Fogel and Sagi, 1989] Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113.

[Forman, 2003] Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305.

[Frouzova et al., 2005] Frouzova, J., Kubecka, J., Balk, H., and Frouz, J. (2005). Target strength of some European fish species and its dependence on fish body parameters. *Fisheries Research*, 75(1-3):86–96.

[Furey et al., 2000] Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and vali-

dation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

[Gehler and Nowozin, 2009] Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *Proceedings of the IEEE 12th International Conference on Computer Vision*, pages 221–228.

[Gomes et al., 2003] Gomes, R. M. F., Sousa, J. B., and Pereira, F. L. (2003). Modeling and control of the IES project ROV. In *European Control Conference*, pages 3436–3441, Cambridge, UK.

[Gordon, 1987] Gordon, A. D. (1987). A review of hierarchical classification. *J. Royal Stat. Soc.*, 150(2):119–137.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182.

[Guyon et al., 2002] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.

[Haralick et al., 1973] Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3(6):610–621.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. J. H. (2001). *The elements of statistical learning*, volume 1. Springer New York.

[He and Yung, 2004] He, X.-C. and Yung, N. H. (2004). Curvature scale space corner detector with adaptive threshold and dynamic region of support. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*, volume 2, pages 791–794. IEEE.

[Heithaus and Dill, 2002] Heithaus, M. R. and Dill, L. M. (2002). Food availability and tiger shark predation risk influence bottlenose dolphin habitat use. *Ecology*, 83(2):480–491.

[Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 278–282.

[Hsu and Lin, 2002] Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

[Hu, 1962] Hu, M. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187.

[Hu et al., 2011] Hu, Y., Mian, A. S., and Owens, R. (2011). Sparse approximated nearest points for image set classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 121–128.

[Huang et al., 2012] Huang, P. X., Boom, B. J., and Fisher, R. B. (2012). Underwater live fish recognition using balance-guaranteed optimized tree. In *Proceedings of the 11th Asian Conference on Computer Vision*, volume 7724, pages 422–433.

[Kaewtrakulpong and Bowden, 2001] Kaewtrakulpong, P. and Bowden, R. (2001). An improved adaptive background mixture model for realtime tracking with shadow detection. In *European Workshop on Advanced Video-Based Surveillance Systems*.

[Katselis et al., 2007] Katselis, G., Koukou, K., Dimitriou, E., and Koutsikopoulos, C. (2007). Short-term seaward fish migration in the messolonghi–etoliko lagoons (western greek coast) in relation to climatic variables and the lunar cycle. *Estuarine, Coastal and Shelf Science*, 73(3):571–582.

[Kim et al., 2007] Kim, T.-K., Arandjelović, O., and Cipolla, R. (2007). Boosted manifold principal angles for image set-based recognition. *Pattern Recogn.*, 40(9):2475–2484.

[Larsen et al., 2009] Larsen, R., Ólafsdóttir, H., and Ersbøll, B. (2009). Shape and texture based classification of fish species. In *Proceedings of the Scandinavian Conference on Image Analysis*, page 745–749.

[Lee et al., 2004] Lee, D., Schoenberger, R. B., Shiozawa, D., Xu, X. Q., and Zhan, P. C. (2004). Contour matching for a fish recognition and migration-monitoring system. *Proc. of SPIE*, 5606(1):37–48.

[Lee et al., 2003] Lee, D. J., Redd, S., Schoenberger, R., Xu, X., and Zhan, P. (2003). An automated fish species classification and migration monitoring system. In *Proceedings of the IEEE Industrial Electronics Society.*, volume 2, pages 1080–1085.

[Li et al., 2004] Li, T., Zhang, C., and Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437.

[Liang et al., 2010] Liang, Y., Li, J., and Zhang, B. (2010). Learning vocabulary-based hashing with adaboost. In *Proceedings of the 16th International Conference on Advances in Multimedia Modeling*, pages 545–555. Springer.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from Scale-Invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Ma et al., 2000] Ma, B., Hero, A., Gorman, J., and Michel, O. (2000). Image registration with minimum spanning tree algorithm. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 481–484.

[Maron and Lozano-Pérez, 1998] Maron, O. and Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 570–576.

[Mathis and Breuel, 2002] Mathis, C. and Breuel, T. (2002). Classification using a hierarchical Bayesian approach. In *Proceedings of 16th International Conference on Pattern Recognition*, volume 4, pages 103–106. IEEE.

[Matthews and Baker, 2004] Matthews, I. and Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164.

[McFarlane and Tillett, 1997] McFarlane, N. J. B. and Tillett, R. D. (1997). Fitting 3D point distribution models of fish to stereo images. In *British Machine Vision Conference BMVC*, volume 1, page 330–339.

[Mckenna et al., 1998] Mckenna, S. J., Gong, S., and Raja, Y. (1998). Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12):1883–1892.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

[Miller and Lea, 1976] Miller, D. J. and Lea, R. N. (1976). *Guide to the coastal marine fishes of California*. UCANR Publications.

[Mokhtarian et al., 1997] Mokhtarian, F., Abbasi, S., Kittler, J., et al. (1997). Efficient and robust retrieval by shape content through curvature scale space. *Series on Software Engineering and Knowledge Engineering*, 8:51–58.

[Mokhtarian and Mackworth, 1992] Mokhtarian, F. and Mackworth, A. K. (1992). A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805.

[Mokhtarian and Suomela, 1998] Mokhtarian, F. and Suomela, R. (1998). Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381.

[Morais et al., 2005] Morais, E. F., Campos, M. F. M., Padua, F. L. C., and Carceroni, R. L. (2005). Particle Filter-Based predictive tracking for robust fish counting. In *Proceedings of the 18th Brazilian Symposium on Computer Graphics and Image Processing. SIBGRAPI*, pages 367–374.

[Nadarajan et al., 2009] Nadarajan, G., Chen-Burger, Y. H., and Fisher, R. B. (2009). A knowledge-based planner for processing unconstrained underwater videos. In *IJCAI' Workshop on Learning Structural Knowledge From Observations*.

[Nadarajan et al., 2011] Nadarajan, G., Chen-Burger, Y.-H., Fisher, R. B., and Spampinato, C. (2011). A flexible system for automated composition of intelligent video analysis. In *Proceedings of the 7th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 259–264. IEEE.

[Nery et al., 2005] Nery, M. S., Machado, A., Campos, M. F. M., Padua, F., Carceroni, R., and Queiroz-Neto, J. (2005). Determining the appropriate feature set for fish classification tasks. In *Proceedings of the 18th Brazilian Symposium on Computer Graphics and Image Processing SIBGRAPI*, pages 173–180.

[Nilsback and Zisserman, 2006] Nilsback, M.-E. and Zisserman, A. (2006). A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–C1454.

[Nilsback and Zisserman, 2007] Nilsback, M.-E. and Zisserman, A. (2007). Delving into the whorl of flower segmentation. In *Proceedings of the BMVC*, volume 1, pages 570–579.

[Nilsback and Zisserman, 2008] Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

[Otsu, 1979] Otsu, N. (1979). A threshold selection method from graylevel histograms. *IEEE Trans. Syst., Madn. & Cybern.*, 9:62–66.

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances In Large Margin Classifiers*, pages 61–74. MIT Press.

[Platt et al., 2000] Platt, J. C., Cristianini, N., and Shawe-Taylor, J. (2000). Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems 12*, pages 547–553.

[Rodrigues et al., 2010] Rodrigues, M. T. A., Páanddua, F. L. C., Gomes, R. M., and Soares, G. E. (2010). Automatic fish species classification based on robust feature extraction techniques and artificial immune systems. In *Proceedings of the IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 1518 –1525.

[Ros-Sánchez et al., 2010] Ros-Sánchez, G., García-Mateos, G., Vera, L. M., and Sánchez-Vázquez, F. J. (2010). A new taxonomy and graphical representation for visual fish analysis with a case study. *environment*, 1:10000.

[Rother et al., 2004] Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314.

[Rova et al., 2007] Rova, A., Mori, G., and Dill, L. M. (2007). One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video. In *IAPR Conference on Machine Vision Applications*, pages 404–407.

[Ruff et al., 1995] Ruff, B. P., Marchant, J. A., and Frost, A. R. (1995). Fish sizing and monitoring using a stereo image analysis system applied to fish farming. *Aquacultural engineering*, 14(2):155–173.

[Saeys et al., 2007] Saeys, Y., Inza, I. n., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

[Salam et al., 2004] Salam, R. A., Ee, A. O., and Hitam, M. S. (2004). *Unsupervised Color Correction Using Cast Removal for Underwater Images*. World Scientific and Engineering Academy (WSEAS) Transactions on Information Science and Applications.

[Schettini and Corchs, 2010] Schettini, R. and Corchs, S. (2010). Underwater image processing: state of the art of restoration and image enhancement methods. *EURASIP J. Adv. Signal Process*, 2010:14:1–14:7.

[Shakhnarovich et al., 2002] Shakhnarovich, G., Fisher, J. W., and Darrell, T. (2002). Face recognition from long-term observations. In *Proceedings of the 7th European Conference on Computer Vision*, pages 851–865. Springer.

[Shental et al., 2003] Shental, N., Bar-hillel, A., Hertz, T., and Weinshall, D. (2003). Computing gaussian mixture models with EM using equivalence constraints. In *Advances in Neural Information Processing Systems 16*. MIT Press.

[Shieh and Yang, 2008] Shieh, M.-D. and Yang, C.-C. (2008). Multiclass SVM-RFE for product form feature selection. *Expert Systems with Applications*, 35(1-2):531–541.

[Soh and Tsatsoulis, 1999] Soh, L.-K. and Tsatsoulis, C. (1999). Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2):780–795.

[Spampinato et al., 2008] Spampinato, C., Chen-Burger, Y., Nadarajan, G., and Fisher, R. B. (2008). Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *Proceedings of the 3rd International Conference on Computer Vision Theory and Applications (VISAPP'08)*, page 514–519.

[Spampinato et al., 2010] Spampinato, C., Giordano, D., Salvo, R. D., Chen-Burger, Y. H., Fisher, R. B., and Nadarajan, G. (2010). Automatic fish classification for underwater species behavior understanding. In *Proceedings of the first ACM international workshop on analysis and retrieval of tracked events and motion in imagery streams*, pages 45–50, New York, NY, USA.

[Strachan, 1993a] Strachan, N. J. C. (1993a). Length measurement of fish by computer vision. *Computers and electronics in agriculture*, 8(2):93–104.

[Strachan, 1993b] Strachan, N. J. C. (1993b). Recognition of fish species by colour and shape. *Image and Vision Computing*, 11:2–10.

[Strachan et al., 1990] Strachan, N. J. C., Nesvadba, P., and Allen, A. R. (1990). Fish species recognition by shape analysis of images. *Pattern Recognition*, 23(5):539–544.

[Suykens and Vandewalle, 1999] Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[Toh et al., 2009] Toh, Y. H., Ng, T. M., and Liew, B. K. (2009). Automated fish counting using image processing. In *International Conference on Computational Intelligence and Software Engineering*, pages 1–5.

[Tong and Koller, 2002] Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66.

[Torres et al., 2004] Torres, R. S., Falcâo, A. X., and da F. Costa, L. (2004). A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174.

[Torres-Mendez and Dudek, 2005] Torres-Mendez, L. A. and Dudek, G. (2005). A statistical learning-based method for color correction of underwater images. *Research on Computer Science*, 17(10).

[Varma and Ray, 2007] Varma, M. and Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *Proceeding of 11th International Conference on Computer Vision*, pages 1–8. IEEE.

[Walther et al., 2004] Walther, D., Edgington, D. R., and Koch, C. (2004). Detection and tracking of objects in underwater video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 544–549.

[Wang et al., 2012] Wang, R., Guo, H., Davis, L. S., and Dai, Q. (2012). Covariance discriminative learning: A natural and efficient approach to image set classification.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2496–2503.

[Wang et al., 2008] Wang, R., Shan, S., Chen, X., and Gao, W. (2008). Manifold-Manifold distance with application to face recognition based on image set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Wang and Casasent, 2009] Wang, Y.-C. F. and Casasent, D. (2009). A support vector hierarchical method for multi-class classification and rejection. In *Proceedings of the International Joint Conference on Neural Networks IJCNN*, pages 3281–3288.

[Wolf and Shashua, 2003] Wolf, L. and Shashua, A. (2003). Learning over sets using kernel principal angles. *The Journal of Machine Learning Research*, 4:913–931.

[Yamaguchi et al., 1998] Yamaguchi, O., Fukui, K., and Maeda, K.-i. (1998). Face recognition using temporal image sequence. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 318–323.

[Yang et al., 2005] Yang, J., Yan, R., and Hauptmann, A. G. (2005). Multiple instance learning for labeling faces in broadcasting news video. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 31–40.

[Zahn and Roskies, 1972] Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21(3):269–281.

[Zhang and Goldman, 2001] Zhang, Q. and Goldman, S. A. (2001). EM-DD: an improved multiple-instance learning technique. In *Advances in neural information processing systems*, pages 1073–1080.

[Zhao et al., 2003] Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35:399–458.

[Zion et al., 1999] Zion, B., Shklyar, A., and Karplus, I. (1999). Sorting fish by computer vision. *Computers and electronics in agriculture*, 23(3):175–187.

[Zion et al., 2000] Zion, B., Shklyar, A., and Karplus, I. (2000). In-vivo fish sorting by computer vision. *Aquacultural Engineering*, 22(3):165–179.

[Zompola et al., 2008] Zompola, S., Katselis, G., Koutsikopoulos, C., and Cladas, Y. (2008). Temporal patterns of glass eel migration (*Anguilla anguilla* l. 1758) in relation to environmental factors in the western greek inland waters. *Estuarine, Coastal and Shelf Science*, 80(3):330–338.

[Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.