Chapter 11
Hierarchical Classification System with Reject Option for Live Fish Recognition

Phoenix X. Huang

Abstract This chapter presents a Balance-Guaranteed Optimized Tree with Reject option (BGOTR) for live fish recognition in a non-constrained environment. It recognizes the top 15 common species of fish and detects new species in an unrestricted natural environment recorded by underwater cameras. This system can assist ecological surveillance research, e.g., obtaining fish population statistics from the open sea. BGOTR is automatically constructed based on inter-class similarities. We apply a Gaussian Mixture Model (GMM) and Bayes rule as a reject option after hierarchical classification—we estimate the posterior probability of being a certain species and then filter out less confident decisions. The proposed BGOTR-based hierarchical classification method achieves significant improvements compared to state-of-the-art techniques on a live fish image dataset of 24,150 manually labeled images from the south Taiwan sea.

11.1 Introduction

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.

P.X. Huang (✉)
Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
e-mail: forestrocket@gmail.com

R.B. Fisher et al. (eds.), *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*, Intelligent Systems Reference Library 104,
DOI 10.1007/978-3-319-30208-9_11

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 classification 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. After forward sequential feature selection and training each Support vector machine (SVM), Individual feature selection based SVM (IFS-SVM) classifies each test sample by counting votes that are optimized for every pair of specific classes. Tested on a manually labeled fish dataset of 24,150 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. A figure of the whole recognition system is shown in Fig. 11.1.

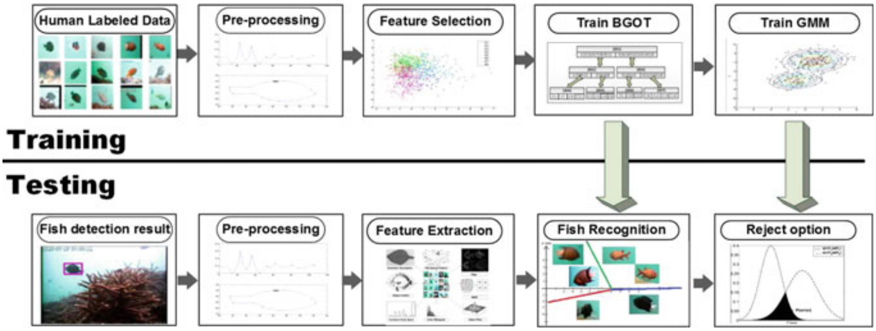

Fig. 11.1 The fish recognition system, an overview framework

11.2 Related Work

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 environment, 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. 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). This equipment has produced large amounts of data, and it requires informatics technology like computer vision and pattern recognition to analyze 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). Unlike the simple and constrained environments found in the majority of previous work (e.g., fish tanks (Lee et al. 2004; Ruff et al. 1995), conveyor belts (Strachan 1993), dead fish (Larsen et al. 2009)), we investigate the recognition task of more fish species in a more complex and fundamentally challenging natural environment. We use underwater camera 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). Recently, Duan et al. (2012) used fine-grained method to closely related categories like classify animal species by choosing relevant local attributes. However, the fine-grained method requires high standard about the quality of input images, which is not always met in our dataset. Instead, we designed some species-specific features for fish recognition (e.g., white tail for *Chromis margaritifer*, color stripe for *Amphiprion clarkii*).

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 could 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). This method, called hierarchical classification, 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). However, one problem of the hierarchical classification method is error accumulation. Each level of the hierarchical tree has some classification errors and these compounds as one goes deeper down the tree. As a result, realistic applications usually require rejection to eliminate the accumulated errors from hierarchical classification (Wang and Casasent 2009). 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, false detections (e.g., non-fish objects, blurred images) and fish from an unknown species are also input to the recognition process. We introduce rejection into hierarchical classification by calculating the Bayesian posterior density. A GMM model is applied at the leaves of the hierarchical tree as the reject option. It evaluates the posterior probability of the test samples and rejects low probability samples. Using a reject option produces a lower false positive rate, but at the price of a slightly lower true positive rate due to incorrect rejections.

11.3 Feature Extraction

We observe fish images from underwater telerecording streams. These fish images record the illumination values (RGB) of pixels over the observing range. Unfortunately, the appearance of the fish are not constant due to the various conditions of, e.g., illuminations, reflections, shadows, etc. However, computers can only distinguish the fish from digital numeral data of extracted features. For example, in fish recognition, some species of fish have specific colors, fin shapes, stripes or texture. Computer vision techniques exploit these similarities, and present them by similar feature density distributions.

This section 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 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 color, shape and texture properties in different parts of the fish such as tail, head, top and bottom are extracted.

11.3.1 Image Pre-processing

The pre-processing is undertaken to improve the quality of features. Firstly, the detection and tracking software described in Spampinato et al. (2014b) 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-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 neighbors. This method can overcome normal image distortion, such as additional noise and water reflection, which triggers segmentation errors in other algorithms. 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 Fig. 11.2, where most parts of the key feature (white tail) are preserved by the segmentation algorithm.

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 that a fish's head part is smoother than its tail part because it needs a more frictional tail (caudal fin) to swim and keep its body balanced. As a result, the centroid of the curvature value on the fish contour is located on the tail part.

More specifically, the curvature value of each boundary pixel is defined as follows (Mokhtarian and Suomela 1998; He and Yung 2004):

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{\frac{3}{2}}} \tag{11.1}$$

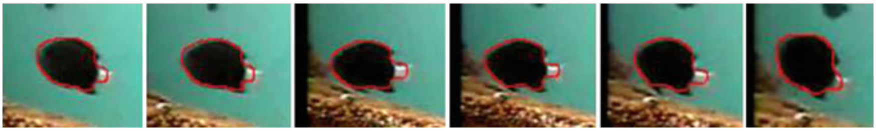

Fig. 11.2 An example of fish detection from a whole trajectory of *Chromis margaritifer*. This species of fish has a noteworthy white tail. This feature is essential for discriminating it from other species of fish, especially *Dascyllus reticulatus*. These images have successfully maintained most parts of the white tails

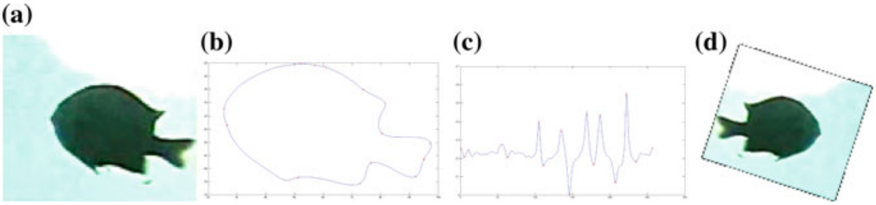

Fig. 11.3 Fish orientation demonstration: **a** input image of *Dascyllus reticulatus* fish; **b** fish boundary after Gaussian smoothing, with small spines eliminated since we are only interested in substantial fluctuations; **c** curvature levels along fish boundary, where the x-axis is the index of 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. This method helps to divide fish in a constant way and extracts specific features (e.g., the white tail of *Chromis margaritifer*)

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. A typical fish orientation procedure is illustrated in Fig. 11.3. Considering the first image (Fig. 11.3a) as input, we first smooth the contour image with a Gaussian filter to eliminate spines, which generate pulses in curvature and should be excluded since we only care about substantial components (Fig. 11.3b). The degrees of curvature of fish contour are illustrated in Fig. 11.3c, 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 11.3d shows the final result, where the *Dascyllus reticulatus* is rotated horizontally and faces right. The fish orientation method achieves 95 % correct fish orientation ±15° using 1000 manually labeled fish images.

11.3.2 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 classification step. Feature engineering work aims at obtaining discriminative characteristics of input data. In this section, we propose a set of effective low level visual descriptors for fish images. We treat this as an incremental process, where new features are designed to improve on the accuracy achieved by appropriate combinations of existing features. More specifically, we put all existing features into a pool for selection, and the algorithm chooses the candidate features which maximize the averaged classification accuracy over all species. Sixty nine types of feature are extracted. These features are a combination of color, shape and texture properties in different parts of the fish such as tail/head/top/bottom, as well as the whole fish. We use normalized color histogram in the Red&Green channel and the Hue component in HSV color space. These color features are normalized to minimize the effect of illumination changes. In order to equalize the color 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. We also introduce a set of new features which help distinguish fish species that tend to be misclassified, including projected color density, tail/head and tail/body area ratios. These features are designed to integrate computer vision techniques with marine knowledge. 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 color density, which describes the color variations of fish body changes in both horizontal and vertical directions and generates a density histogram by calculating the mean value of color along the axis. This feature is useful for describing the significant surface marks such as the colorful tail, stripes, and spots of fish. The mean and standard deviation of the projected density are stored as idiosyncratic fish features.

In order to describe the fish texture, we calculate the Gray-Level Co-occurrence Matrix (GLCM), Fourier descriptor and Gabor filter. The GLCM describes the co-occurrence frequency of two gray scale pixels at a given distance d. The frequency is calculated for four angles ϕ: 0°, 45°, 90°, and 135°. The offset distance ranges from 1 to 10. We computed the GLCM for 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.

Histogram of oriented gradients and Moment Invariants, as well as Affine Moment Invariants, are employed as the shape features. Furthermore, some specific features like tail/head area ratio, tail/body area ratio, etc. are also included.

These descriptors are found to be effective. 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 analyzing the experimental results. As discussed before, we propose 69 groups of features (2626 dimensions) to recognize fish. Example and more details are included in Huang (2014). These features are a combination of the color, shape, 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).

11.4 Fish Recognition

The Balance Guaranteed Optimized Tree with reject option (BGOTR) 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. BGOTR 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 section, which compares the normal flat classifier approach to the hierarchical classification method. The latter method uses a divide and conquer strategy, and organizes candidate classes into multiple levels. In a greatly imbalanced dataset, the less common 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 BGOTR 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. 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.

11.4.1 The Balance Guaranteed Optimized Tree Method

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, $\tilde{\text{F}}_t \subset \{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$ 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 Fig. 11.4, 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.

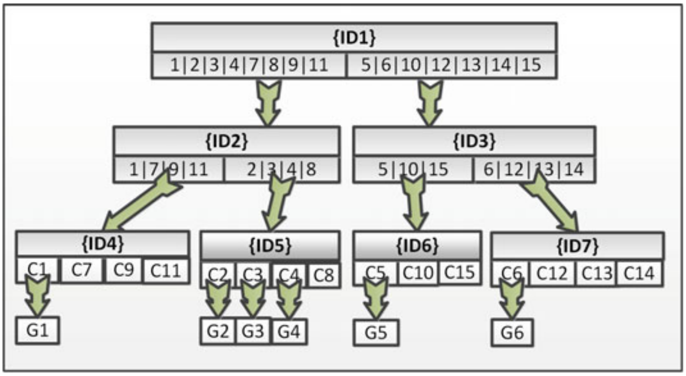

Fig. 11.4 GMM for rejection post-processing for classes C1, . . ., C6 in hierarchical classification, integrated with a BGOTR method

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 the given split, using forward sequential feature selection based on grouped subset of features. 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. 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.

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 (Table 11.1), when the full dataset of 241,500 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

Table 11.1 Fish recognition exploration for choosing the most effective classifier

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

Average Recall (AR), Average Precision (AP), Accuracy by Count (AC) are introduced in the experimental Sect. 11.5

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.

11.4.2 Trajectory Voting Method

In the view of a 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. Using the fish recognition results 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 in a non-constrained environment. 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 matching. Shakhnarovich et al. (2002) consider the features of multiple observations as a whole, and propose a classification based on their distributions. 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 (Maron and Lozano-Pérez 1998; Zhang and Goldman 2001; Yang et al. 2005).

In our live fish recognition system, we have applied the majority voting algorithm to make use of the temporal information embedded in fish trajectories, and to minimize the environmental influence. 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.

11.4.3 Gaussian Mixture Model For Reject Option

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 colors, 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 assigning the training class), the GMM captures the prominent density distributions and is not constrained by the label information. 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 and Jain (2002) present an unsupervised learning algorithm to learn a proper mixture model from multivariate data. It can automatically select the finite mixture model by using the minimum message length (MML) with advantages compared to other deterministic criteria, e.g., Bayesian Inference criterion (BIC), Minimum Description length (MDL): in particular, it is less sensitive to the initialization, and avoids the boundary of the parameter space.

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 integrating the result score along the path of the hierarchy, here a GMM model is applied after the BGOTR classification to implement the reject option (Fig. 11.4). The GMM model is trained by a subset of features by using the forward sequential selection method. For each BGOTR 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:

$$p(C_i \mid X) = \frac{p(C_i)p(X \mid C_i)}{p(X)} = \frac{p(C_i)p(X \mid C_i)}{\sum_j p(C_j)p(X \mid C_j)} \tag{11.2}$$

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 BGOTR but a different subset was selected (using the same feature selection criteria). 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 like-

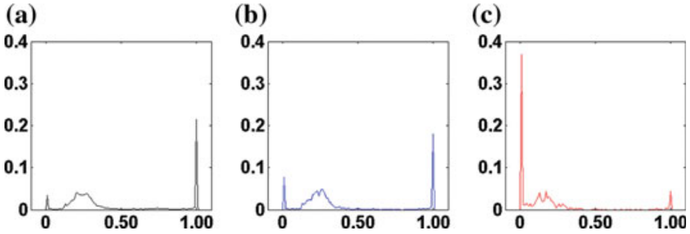

Fig. 11.5 **a** Distribution of posterior probability of the training samples of species *Chromis chrysura*. **b** Distribution of posterior probability of test sample True Positives. **c** Distribution of posterior probability of test sample False Positives. See text for details

lihood 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 BGOTR prediction C_i, we calculate its posterior probability $P(C_i \mid X)$ from Eq. 11.2 and set a small threshold (i.e., 0.01) to reject all samples whose posterior probabilities are below the threshold. Figure 11.5 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, Fig. 11.5b) or misclassified (False Positives, Fig. 11.5c). The distribution of the posterior probability of False Positives (as shown in Fig. 11.5c) has a peak distribution (about 38 %) around the value of zero while most of the True Positives have higher posterior probability (Fig. 11.5b). The difference 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).

11.5 Fish Recognition Experiments

Our data is acquired from a live fish dataset of the 15 different species shown in Fig. 11.6. 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.

All fish are manually labeled by following instructions from marine biologists (Boom et al. 2012). The labeling work was supported by a clustering method. Then, three users checked and cleared the clustering results. The final annotation work was confirmed by two marine biologists. 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 section.

11.5.1 Fish Recognition Experiments Using Ground Truth Data

We use the BGOTR 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

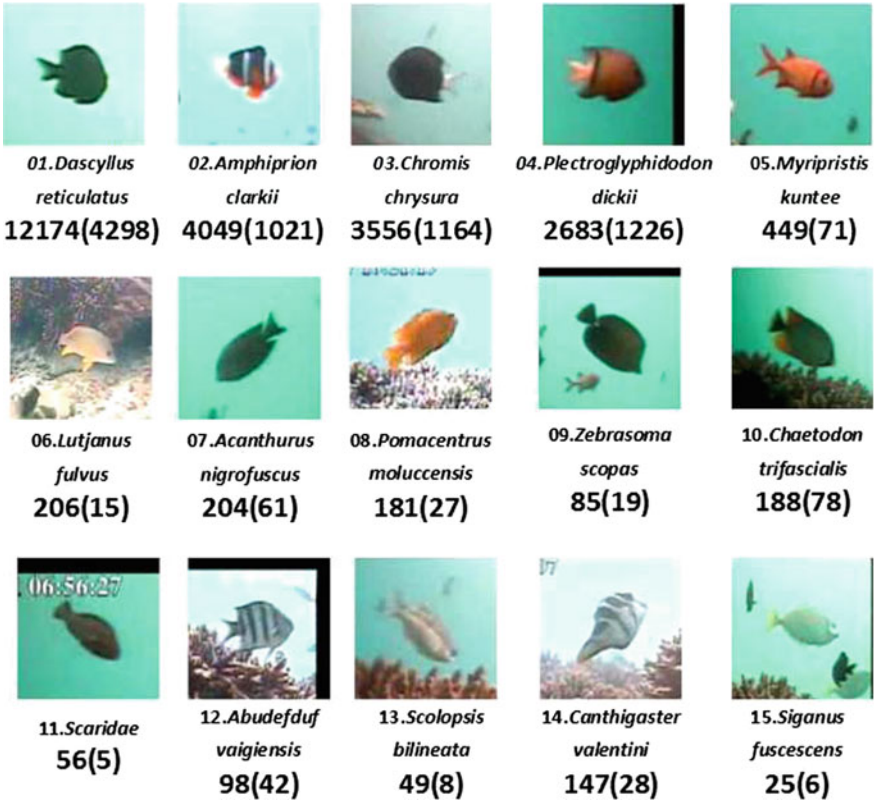

Fig. 11.6 Top 15 species of fish in underwater videos, with the number of observations and trajectories in the ground-truth. All in all, there are 24,150 observations and 8069 trajectories

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.
6. An automatically generated tree (BGOTR) 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 BGOTR 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 less common classes. Feature selection typically selects about 300 of the features at each node.

The experiment is based on 24,150 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 11.2 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. 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. 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.

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. 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 seventh 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

Table 11.2 Fish recognition results

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
BGOTR	**84.8*** ± 3.9	**91.4** ± 2.8	**97.5*** ± 0.6

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

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 BGOTR 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 %. These experimental results demonstrate that reject option has significantly improved the fish recognition performance where comparing to other state-of-the-art techniques, more details are included in Huang et al. (2014).

11.5.2 BGOTR Application to New Real Fish Videos

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 Fig. 11.7. In this section we experimentally evaluate how many false detections our BGOTR system can reject while preserving the valid ones. We chose 3 underwater videos and have labeled 1000 detections from each video.

The recognition results are shown in Tables 11.3 and 11.4. We use BGOTR 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

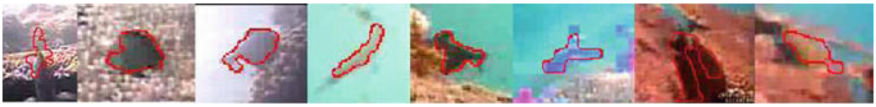

Fig. 11.7 Invalid fish images, chosen from 3 underwater videos. In a normal classifier without a reject option, these images would be classified and cause unexpected results. Our rejection algorithm aims at eliminating them while preserving most valid fish images

Table 11.3 Experiment result for real videos

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

In each video we select the first 1000 detections and manually label all samples

Table 11.4 Experiment of rejection result in real videos

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	68
Average	323	677	501	465	36

TR = True Rejection, FR = False Rejection

demonstrates that the BGOTR method recognizes about 78 % of the real, untrained valid fish images correctly. The test images include many invalid detections (692, 892, 487, respectively). The BGOTR 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 %).

11.6 Conclusion

Live fish recognition in the open sea is fundamentally challenging because of a 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. None of these methods works because of the unconstrained environment and imbalanced dataset. In this chapter, we presented a novel Balance-Guaranteed Optimized Tree (BGOTR) classifier for live fish recognition in a non-constrained environment. Although hierarchical classification is widely applied in machine vision applications, BGOTR improves the normal hierarchical method by two heuristics for how to organize a single classifier and construct a hierarchical tree with higher accuracy. After constructing the tree architecture, a novel trajectory voting method is used to eliminate accumulated errors during hierarchical classification and achieves better performance. The novel rejection system enhances 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 BGOTR hierarchical method. It evaluates the posterior probability of the testing samples and reduces the false positive rate, since some misclassification errors in the BGOTR classifier can be overcome at the price of a slightly lower true positive rate due to incorrect rejections. 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 11.2). More detailed information is included in Huang et al. (2012, 2014), Huang (2014).

References

Boom, B., P. Huang, J. He, and R.B. Fisher. 2012. Supporting ground-truth annotation of image datasets using clustering. In *Proceedings of 21st international conference on pattern recognition (ICPR)*, 1542–1545. IEEE.

Brehmer, P., T.D. Chi, and D. Mouillot. 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, L. 2001. Random forests. *Machine learning* 45(1): 5–32.

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

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

Chib, S. 1995. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association* 90(432): 1313–1321.

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

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

Deng, J., A.C. Berg, K. Li, and L. Fei-Fei. 2010. What does classifying more than 10,000 image categories tell us? In *Proceedings of the 11th european conference on computer vision*, 71–84. Berlin: Springer.

Duan, K., D. Parikh, D. Crandall, and K. Grauman. 2012. Discovering localized attributes for fine-grained recognition. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, 3474–3481. IEEE.

Edgington, D.R., D.E. Cline, D. Davis, I. Kerkez, and J. Mariette. 2006. Detecting, tracking and classifying animals in underwater video. In *OCEANS*, 1–5.

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

Gordon, A.D. 1987. A review of hierarchical classification. *Journal of the Royal Statistical Society* 150(2): 119–137.

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

Hastie, T., R. Tibshirani, and J.J.H. Friedman. 2001. *The elements of statistical learning*, vol. 1. New York: Springer.

He, X.-C., and N.H. Yung. 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*, vol. 2, 791–794. IEEE.

Ho, T.K. 1995. Random decision forests. In *Proceedings of the third international conference on document analysis and recognition*, 278–282.

Huang, X.P. 2014. Balance-Guaranteed Optimized Tree with Reject option for live fish recognition. PhD thesis, University of Edinburgh.

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

Huang, P.X., B.J. Boom, and R.B. Fisher. 2014. GMM improves the reject option in hierarchical classification for fish recognition. In *Proceedings of Workshop on Applications of Computer Vision 2014*. 371–376.

Larsen, R., H. Ólafsdóttir, and B. Ersbøll. 2009. Shape and texture based classification of fish species. In *Proceedings of the scandinavian conference on image analysis*, 745–749.

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

Liang, Y., J. Li, and B. Zhang. 2010. Learning vocabulary-based hashing with adaboost. In *Proceedings of the 16th international conference on advances in multimedia modeling*, 545–555. Springer.

Maron, O., and T. Lozano-Pérez. 1998. A framework for multiple-instance learning. In *Proceedings of the conference on advances in neural information processing systems*, 570–576.

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

Nadarajan, G., Y.-H. Chen-Burger, R.B. Fisher, and C. Spampinato. 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)*, 259–264. IEEE.

Rother, C., V. Kolmogorov, and A. Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transaction on Graphics* 23(3): 309–314.

Rova, A., G. Mori, and L.M. Dill. 2007. One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video. In *IAPR conference on machine vision applications*, 404–407.

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

Shakhnarovich, G., J.W. Fisher, and T. Darrell. 2002. Face recognition from long-term observations. In *Proceedings of the 7th European conference on computer vision*, 851–865. Springer.

Shental, N., A. Bar-hillel, T. Hertz, and D. Weinshall. 2003. Computing gaussian mixture models with EM using equivalence constraints. In *Advances in neural information processing systems 16*. MIT Press.

Soh, L.-K., and C. Tsatsoulis. 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, C., S. Palazzo, B. Boom, J. van Ossenbruggen, I. Kavasidis, R. Di Salvo, F.-P. Lin, D. Giordano, L. Hardman, and R. Fisher. 2014b. Understanding fish behavior during typhoon events in real-life underwater environments. *Multimedia Tools and Applications* 70(1): 199–236.

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

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

Toh, Y.H., T.M. Ng, and B.K. Liew. 2009. Automated fish counting using image processing. In *International conference on computational intelligence and software engineering*, 1–5.

Wang, Y.-C.F., and D. Casasent. 2009. A support vector hierarchical method for multi-class classification and rejection. In *Proceedings of the international joint conference on neural networks IJCNN*, 3281–3288.

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

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

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