View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Edinburgh Research Explorer

THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Underwater Live Fish Recognition Using a Balance-Guaranteed Optimized Tree

Citation for published version:
Huang, PX, Boom, BJ & Fisher, RB 2013, Underwater Live Fish Recognition Using a Balance-Guaranteed Optimized Tree. in KM Lee, Y Matsushita, JM Rehg & Z Hu (eds), Computer Vision – ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I. Lecture Notes in Computer Science, vol. 7724, Springer-Verlag GmbH, pp. 422-433. DOI: 10.1007/978-3-642-37331-2_32

Digital Object Identifier (DOI):
10.1007/978-3-642-37331-2_32

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Computer Vision – ACCV 2012

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05. Apr. 2019

Huang, P. X., Boom, B. J., & Fisher, R. B. (2013). Underwater Live Fish Recognition Using a Balance-Guaranteed Optimized Tree. In Lee, K. M., Matsushita, Y., Rehg, J. M., & Hu, Z. (Eds.), Computer Vision ? ACCV 2012. (pp. 422-433). (Lecture Notes in Computer Science). Springer Berlin / Heidelberg. doi: 10.1007/978-3-642-37331-2_32

Underwater Live Fish Recognition using a Balance-Guaranteed Optimized Tree

Phoenix X. Huang, Bastiaan J. Boom, Robert B. Fisher

School of Informatics, University of Edinburgh

Abstract. Live fish recognition in the open sea is a challenging multi-class classification task. We propose a novel method to recognize fish in an unrestricted natural environment recorded by underwater cameras. This method extracts 66 types of features, which are a combination of color, shape and texture properties from different parts of the fish and reduce the feature dimensions with forward sequential feature selection (FSFS) procedure. The selected features of the FSFS are used by an SVM. We present a Balance-Guaranteed Optimized Tree (BGOT) to control the error accumulation in hierarchical classification and, therefore, achieve better performance. A BGOT of 10 fish species is automatically constructed using the inter-class similarities and a heuristic method. The proposed BGOT-based hierarchical classification method achieves about 4% better accuracy compared to state-of-the-art techniques on a live fish image dataset.

1 Introduction

Live fish recognition in the open sea has been investigated by [1–4] for commercial and environmental applications like fish farming and a meteorologic monitoring. The detected fish are in 3D positions and against coral and sand as well as the open sea. Statistics about the 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 [5, 6]. However, the recognition task is fundamentally challenging because fish can move freely and illumination levels change frequently in such environments [7, 8]. As a result, this task remains an outstanding research problem. Prior research is mainly restricted to constrained environments (*e.g.*, fish tanks [1], conveyor belts [9]). Strachan et al. [3] achieves the scores of 73%, 63% and 90%, respectively, on three types of fish. C. Spampinato et al. [10] classifies 360 images of ten different species and achieves an average accuracy of about 92%. R. Larsen et al. [11] classify three fish species and achieve a recognition rate of 76%. In contrast, this paper investigates novel techniques to perform effective live fish recognition in an unrestricted natural environment.

1.1 Related work

SVM method. The fish recognition task is seen as an application of multi-class classification, which has become an important and interesting research area s-

ince the influence of machine learning theory. Over the last decade, SVM [12] has shown impressive accuracy on the multi-class classification task because of its maximum-margin advantages. However, SVM is originally designed for a binary classification task. Therefore, to enable multi-class classification, several mechanisms, such as one-vs-one and one-vs-rest, have been developed. This kind of multi-class classifier could be considered as a flat classifier because it classifies all classes at the same time [13] 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.

Hierarchical classification tree method. 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 [14]. 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 specifical features at each level [15].

Hierarchical classification has several noticeable advantages. Firstly, it divides all classes into certain subsets and leaves similar classes for a later stage. This strategy balances the load of any single node. 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 the similar groupings. This is especially useful with a large number of categories [14]. Hierarchical structures are popular in document and image categorization. Mathis [16] organizes documents hierarchically by making use of the correlations between topical subjects. Deng *et.al.* [17] 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.

Nonetheless, the hierarchical structure has a critical disadvantage called error accumulation. Each level of the hierarchical tree may have some classification errors. These errors are accumulated into deeper layers and reduce the average accuracy of the final result.

1.2 The framework

In this paper, we propose a novel method to recognize fish in an unrestricted natural environment from underwater videos. We use the Balance-Guaranteed Optimized Tree (BGOT) to help resolve the error accumulation issue and make use of the inner-class similarities among fish species. The framework is illustrated in Fig 1.

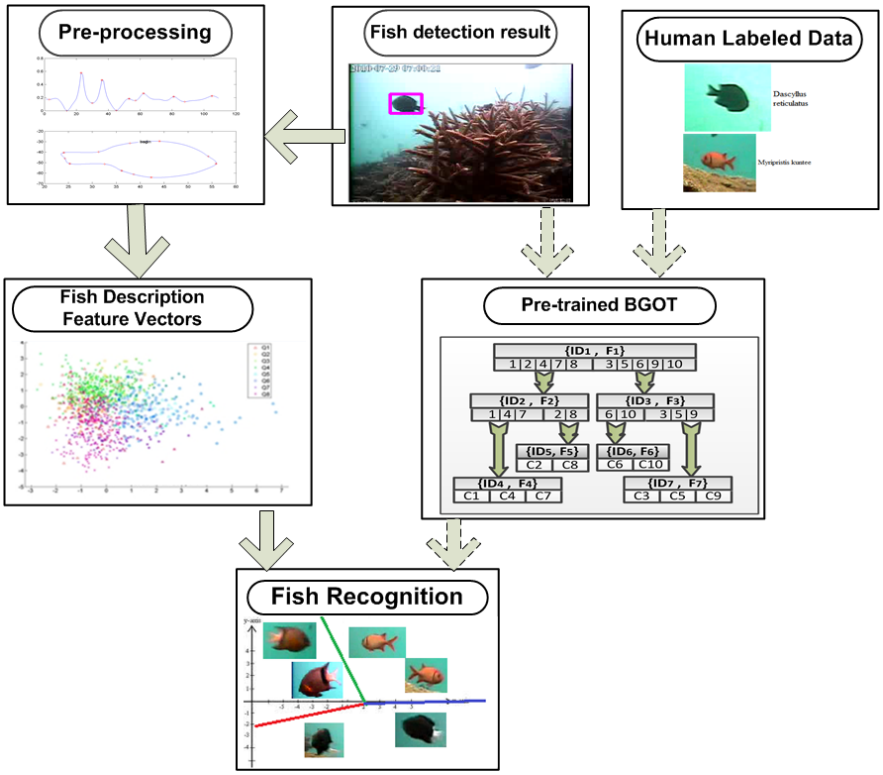


Fig. 1. The framework of our BGOT-based hierarchical classification system. The work flow of dotted arrows shows the training procedure and the solid arrows indicate the recognition procedure.

In this paper we propose a hierarchical classification approach for live fish recognition. Furthermore, we use a heuristic method to construct an automatically generated BGOT and the proposed method is evaluated on a live fish dataset. The algorithm itself is presented in section 2, including the mathematical explanation of hierarchical classification, a set of heuristics which help construct the hierarchical tree. In section 4, we compared the proposed BGOT tree to an Ada-boost [18] method and a flat SVM [12] (section 3) on a fish image set [19].

2 Hierarchical classification approach

Given a set of samples $\{x_i\}_{i=1}^n$, the feature vector $f_i = \{f_{i,1}, ..., f_{i,m}\}$ denotes the m feature values for sample x_i. Let $\{y_i\}_{i=1}^n$ indicate the class label of 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 f_i as input to predict the class label $\tilde{y}_i = h(f_i)$ that maximizes the classification accuracy.

A hierarchical classifier approach 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, $\tilde{\text{F}}_t \subset \{f_1, ..., f_m\}$ 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 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 10 classes is demonstrated in Figure 2, where the ID_t and $\tilde{\text{F}}_t$ are illustrated in each node and $\hat{\text{C}}_t$ is described as local groups.

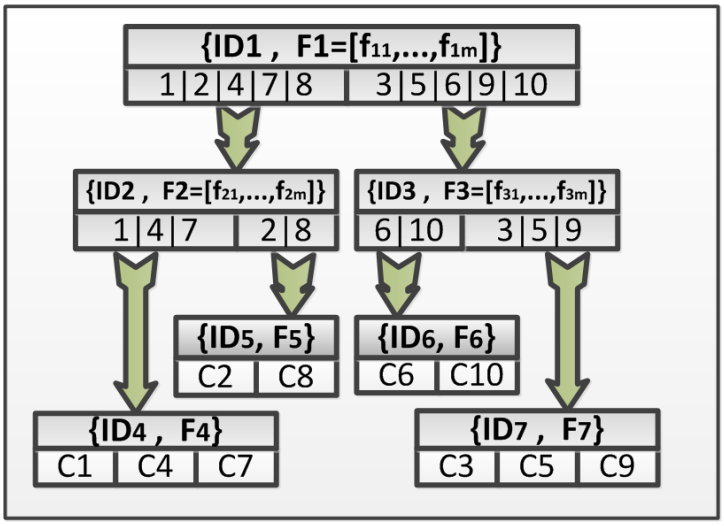


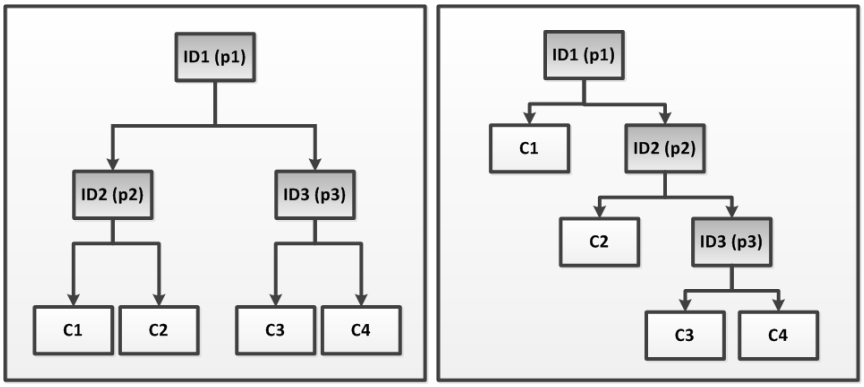
Fig. 2. Automatically generated tree (BGOT), the hierarchical example tree of 10 classes $(C1, ..., C_{10})$.

2.1 Heuristic method

In this paper, 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.

Rule 1 recommends how to assign the single classifiers to a hierarchical tree. We consider the balanced tree T_b in Figure 3(a) with sample number n_i. This



(a) Balanced tree T_b (b) Unbalanced tree T_u

Fig. 3. Examples of hierarchical trees.

tree has 4 classes $\{c_1, c_2, c_3, c_4\}$ and each single classifier has a different accuracy $\{p_1, p_2, p_3\}$. The average accuracy is calculated as $p_1 * \frac{1}{2}(p_2 + p_3)$ assuming all classes have equal magnitude. The best accuracy is achieved by assigning the most accurate classifier to node ID_1. Generally, the result of a balanced hierarchical tree of N nodes has depth $log_2 N$ and average accuracy:

$$P_b = \prod_{i=1}^{log_2 N} \widetilde{P}_i = \prod_{i=1}^{log_2 N} \frac{1}{2^{(i-1)}} \sum_{s=2^{(i-1)}}^{2^i - 1} p_s \tag{1}$$

where p_s is the accuracy of node s and $\widetilde{P}_i$ is the average accuracy of all nodes in layer i. The hierarchical tree achieves better accuracy if we choose the more accurate classifiers at higher layers which equates to assigning these nodes a higher weight. In the future, we will think more about how to construct the hierarchical tree if classes are not at equal size.

Rule 2 is explained by comparing two sample trees: a balanced tree T_b and an unbalanced tree T_u. These examples are shown in Figure 3. Let us assume each class has the same number of samples n_i and each classifier has an equal accuracy p. In T_b, each class is classified with an accuracy p^2, while the average accuracy in T_u is $\frac{1}{4}(p + p^2 + 2p^3)$. We can prove that $P_b > P_u$, for $0.5 < p < 1$. To generalize, a balanced tree of N nodes has average accuracy:

$$P_b = p^{log_2 N} \tag{2}$$

and unbalanced accuracy:

$$P_u = \frac{1}{N}(\sum_{i=1}^{N-1} p^i + p^{N-1}) \tag{3}$$

for $0.5 < p < 1$, $P_b > P_u$. Thus a more balanced hierarchical tree with $log_2 N$ depth suppresses error accumulation, and achieves better accuracy than an unbalanced tree.

2.2 Algorithm of generating BGOT

The BGOT is based on the two heuristics of the last section: keep the hierarchical tree balanced and optimize the performance by putting more accurate nodes at the top layers. In the fish recognition task, some species of fish are more similar than others and the similarity is summarized from the confusion matrix. We illustrate the algorithm of generating BGOT below:

Input: class C1 to Cn
begin $c := [C_1, ..., C_n]$
  $level := 0$
  construct$(c, level)$;
where
proc construct(c, n) $\equiv$
  if $n > MAXDEPTH$ then exit fi;
  comment: find the best binary split of given classes on whole feature set;
  $[cLeft, cRight] := ChooseSplit(c)$;
  comment: The ChooseSplit function splits the class set into equal-size subsets;
  $featureSet = FeatureSelection(cLeft, cRight)$;
  comment: the minimum splitting is set to 3 to limit the max depth;
  if $size(cLeft) > 3$ then
    construct$(cLeft, n + 1)$
  fi;
  if $size(cRight) > 3$ then
    construct$(cRight, n + 1)$
  fi;
end

An example BGOT is shown in Fig 2, where 10 classes are arranged into 3 layers. The first layer splits all classes into two groups: C1, C2, C4, C7, C8 and C3, C5, C6, C9, C10. Then it chooses the feature subset to maximize the average accuracy of these groups. This procedure keeps on until all groups have less than 4 classes.

3 Experiment with fish recognition

Our data is acquired from a live fish dataset with 3179 fish images of the 10 different species shown in Figure 4. This figure shows the fish species name and the numbers of images. As can be seen, the data is very imbalanced where the first two species account for 2564 images. The fish detection and tracking software described in [19] is used to obtain the fish images. The fish species are manually labeled by following instructions from marine biologists.

Figure 3 shows some hard fish examples: blurred, occlusion by other fish or background objects, which include coral, the sea flower and open sea.

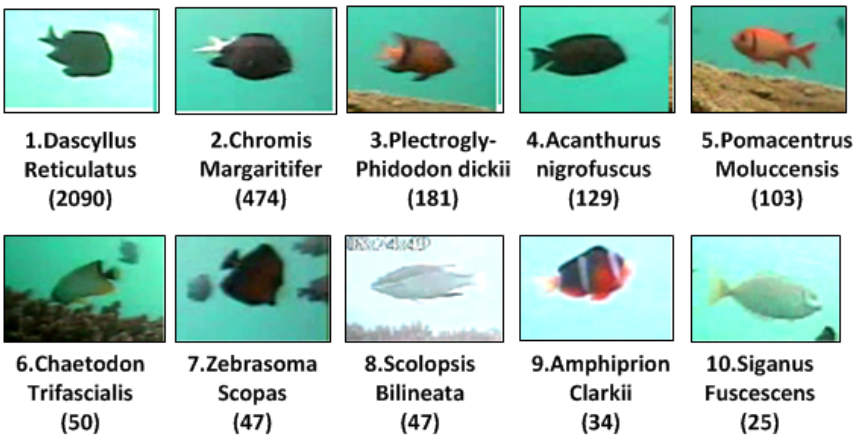


Fig. 4. Top 10 species of fish in underwater videos.

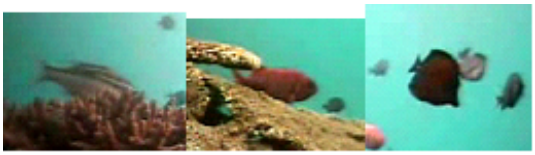


Fig. 5. Hard fish examples.

3.1 Feature extraction

After constructing the fish dataset, some pre-processing procedures are undertaken to improve the recognition rate. Firstly, the Grabcut algorithm [20] is employed to segment fish from the background. Secondly, we propose a streamline hypothesis, which uses the assumption that the head is smoother than the tail. We calculate the fish orientation by weighting each contour pixel with its local curvature scale, and we use this algorithm to align all fish horizontally where the head of the fish is located on the right. we align the fish images to the same direction before further processing. This procedure is carried out based on a streamline assumption, which assumes that most fish have a smoother head than tail because fish need a more frictional tail (caudal fin) to swim and help them keep balance. In order to find the tail side, we smooth the fish boundary with a Gaussian filter to eliminate some noise, and then calculate the curvature of each boundary pixel as following [21, 22]:

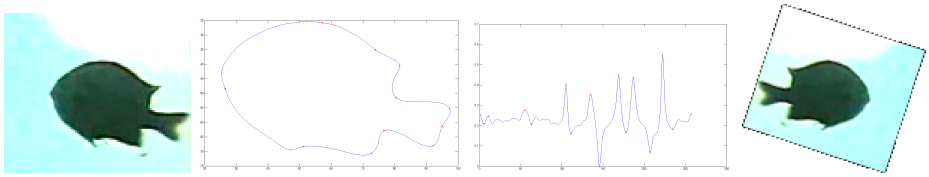
$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2))^{\frac{3}{2}}} \tag{4}$$

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 $y(u)$. However, the pixel curvature is sensitive to local corners and we normalize it using the logarithm function:

$$\kappa_{normalize} = \begin{cases} log(\kappa) & \text{if } \kappa \geq 1 \\ -log(2-\kappa) & \text{if } \kappa < 1 \end{cases} \tag{5}$$

The fish boundary coordinates are weighted by their local curvature and the vector from the center of mask to the curvature weighed center estimates the tail orientation. A typical fish orientation procedure is illustrated in Figure 6. The fish orientation method achieves 95% accuracy using 1000 manually labeled fish images. Finally, every fish image is divided into four parts (head/tail/top/bottom) according to the relative positions from the fish center.



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

Fig. 6. Fish orientation demonstration: (a) original fish image; (b) fish boundary after gaussian filter; (c) curvature along fish boundary; (d) oriented fish image.

After this, 66 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. We recompute the range of every bin according to the average distribution over all samples and map them into a 11-bin histogram to take full advantage of all bins, as shown below:

$$\widetilde{B}_i = \sum_{j=a_i}^{a_{i+1}} B_j \qquad s.t. \qquad a_i = min\{X \in \mathbb{N}^+ \mid \Sigma_{j=1}^{X} \overline{B}_j \geq \frac{i}{11}\} \tag{6}$$

where $B_j, j \in \{1, ..., 50\}$ is the original color histogram bin, $\overline{B}_j, j \in \{1, ..., 50\}$ is the averaged histogram over all samples and $\widetilde{B}_i, i \in \{1, ..., 11\}$ is the recomputed bin.

In order to describe the fish texture, we calculate the co-occurrence matrix, Fourier descriptor and gabor filter. The grey level co-occurrence matrices describe the co-occurrence frequency of two grey scale pixels at a given distance

d [10]:

$$C_{\Delta u,\Delta v}(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1 & \text{if } I(p,q)=i \text{ and } I(p+\Delta u, q+\Delta v)=j \\ 0, & \text{otherwise}\end{cases} \tag{7}$$

The frequency is calculated for several orientations λ. We compute Contrast, Correlation, Energy, Entropy, Homogeneity, Variance, Inverse Difference Moment, Cluster Shade, Cluster Prominence, Max Probability, Auto correlation, Dissimilarity. These 12 features are useful as they are the first selected features by the feature selection procedure.

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. All features are normalized by subtracting the mean and dividing by the standard deviation (z-score normalized).

3.2 Hierarchical classification

As the SVM is firstly designed for binary classification problem, we introduce a one-vs-one strategy with a voting mechanism to convert the binary SVM into a multi-class classifier [12]. Based on the multi-class classifier, we designed two classifiers (see Figure 2):

1. A flat SVM classifier, which classifies all 10 classes simultaneously, is implemented as a baseline classifier.
2. An automatically generated tree (BGOT) is designed by recursively choosing a binary split which has the best accuracy in given classes. We choose binary splitting to keep the tree balanced.

An Ada-boost method [18], which boosts on individual features, is also implemented as a comparison method.

3.3 Results and analysis

The experiment is based on 3179 fish images with a 6-fold cross validation procedure. The training and testing sets are isolated so fish images from the same trajectory sequence are not used during both training and testing. Sequential forward feature selection is applied at each node. We then train a customized classifier at each node for specific classes. Results are listed in Table 1 where the AP and AR results are averaged over all classes rather than over all fish. This is because of the greatly unbalanced class sizes.

The accuracy of a classification system is evaluated as Average Recall (AR), Average Precision (AP) and Accuracy over Count (AC). Generally, given True Positive / False Positive / False Negative, the AR is defined as:

$$AR = \frac{1}{c}\sum_{j=1}^{c}\left(\frac{TruePositive_j}{TruePositive_j + FalseNegative_j}\right) \tag{8}$$

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

$$AP = \frac{1}{c}\sum_{j=1}^{c}(\frac{TruePositive_j}{TruePositive_j + FalsePositive_j}) \tag{9}$$

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

$$AC = \frac{\sum_{j=1}^{c} TruePositive_j}{\sum_{j=1}^{c}(TruePositive_j + FalsePositive_j)} \tag{10}$$

We compare the hierarchical classification against the Ada-boost method (75.3% AR) and flat SVM classifier (86.3% AR). The automatically generated hierarchical tree (BGOT), which chooses the best splitting by exhaustively searching all possible combinations while remaining balanced, achieves an AR of (90.0%). The search procedure takes several hours and a possible improvement is to integrate the hierarchical method with domain knowledge like taxonomy, which helps organize similar species for later processing, instead of exhaustive searching. In the average precision (AP) score, the proposed BGOT method is about 6% better than the baseline SVM method, which are 85.8% and 91.7%, respectively. The Ada-boost method is 76.9% in AP. The AC score of BGOT is tested in a t-test with 95% confidence of significant improvement than the SVM method and Ada-boost method. We calculate the average AC rates at each level in the hierarchy (BGOT): 0.977 (Level 1), 0.9725 (Level 2), 0.950 (Level 3).

Algorithm	AR	AP	AC
Ada-boost	0.753 ± 0.091	0.769 ± 0.092	0.923
Flat SVM	0.863 ± 0.052	0.858 ± 0.061	0.934
BGOT method	0.900 ± 0.042	0.917 ± 0.045	0.950*

Table 1. Fish recognition result. * means significant improvement with 95% confidence.

The individual class recall/precision is shown in Figure 7 and 8. The hierarchical approaches achieve a better accuracy than the flat SVM classifier because they arrange the similar species (1,4,7) into the same group and add fish-tail features to distinguish these species. The flat SVM method misclassified some fish from species 4 to species 8, which achieves a better precision in species 4.

4 Conclusion

In this paper we propose a hierarchical classification approach for live fish recognition. Furthermore, we propose a set of heuristics which are helpful to construct

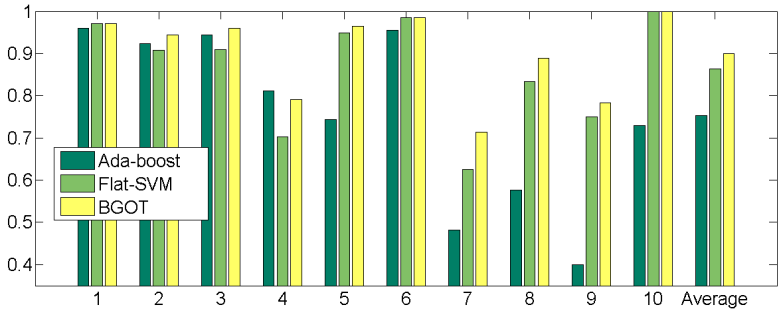


Fig. 7. Recall of 10 species. The BGOT method is better than the baseline method.

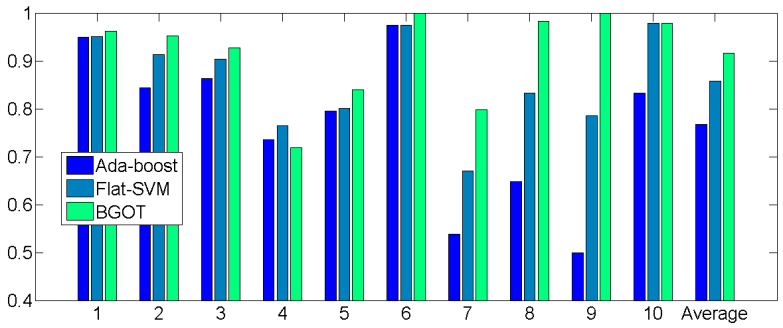


Fig. 8. Precision of 10 species. The BGOT method is better than the baseline method (class 4 is an exception, and is discussed in the result section).

a hierarchical tree. The proposed method is evaluated on a live fish dataset and the automatically generated hierarchical tree (BGOT) achieves *c.* 4% improvement of the average recall (AR) compared to the flat SVM classifier.

Acknowledgements

This work is supported by the Fish4Knowledge project, which is funded by the European Union Seventh Framework Programme [FP7/2007-2013].

References

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

2. Ruff, B.P., Marchant, J.A., Frost, A.R.: Fish sizing and monitoring using a stereo image analysis system applied to fish farming. Aquacultural engineering **14** (1995) 155–173
3. Strachan, N.J.C., Nesvadba, P., Allen, A.R.: Fish species recognition by shape analysis of images. Pattern Recognition **23** (1990) 539–544
4. Okamoto, M., Morita, S., Sato, T.: Fundamental study to estimate fish biomass around coral reef using 3-dimensional underwater video system. In: OCEANS 2000 MTS/IEEE Conference and Exhibition. Volume 2. (2000) 1389–1392
5. Rova, A., Mori, G., Dill, L.M.: One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video. In: IAPR Conference on Machine Vision Applications. (2007) 404–407
6. Zion, B., Shklyar, A., Karplus, I.: In-vivo fish sorting by computer vision. Aquacultural Engineering **22** (2000) 165–179
7. Strachan, N.J.C.: Length measurement of fish by computer vision. Computers and electronics in agriculture **8** (1993) 93–104
8. Toh, Y.H., Ng, T.M., Liew, B.K.: Automated fish counting using image processing. In: International Conference on Computational Intelligence and Software Engineering. (2009) 1–5
9. Strachan, N.J.C.: Recognition of fish species by colour and shape. Image and Vision Computing **11** (1993) 2–10
10. Spampinato, C., Giordano, D., Salvo, R.D., Chen-Burger, Y.H., Fisher, R.B., Nadarajan, G.: 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. (2010) 45–50
11. Larsen, R., Ólafsdóttir, H., Ersbøll, B.: Shape and texture based classification of fish species. In: Proceedings of SCIA. (2009) 745–749
12. Chih-Chung, C., Chih-Jen, L.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2** (2011) 1–27:
13. Carlos, S., Alex, F.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22** (2010) 31–72
14. Deng, J., Berg, A., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: ECCV. Volume 6315. (2010) 71–84
15. Gordon, A.D.: A review of hierarchical classification. J. Royal Stat. Soc. **150** (1987) 119–137
16. Mathis, C.: Classification using a hierarchical bayesian approach. Volume 4 of Proc. of ICPR. (2002) 103–106
17. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. CVPR (2009) 248 –255
18. Freund, Y., Schapire, R.E.: A Decision-Theoretic generalization of on-Line learning and an application to boosting. Computational Learning Theory **904** (1995) 23–37
19. Nadarajan, G., Chen-Burger, Y., Fisher, R., Spampinato, C.: A flexible system for automated composition of intelligent video analysis. Proc. of ISPA (2011) 259–264
20. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. on Graphics (TOG) (2004) 309–314
21. He, X.C., Yung, N.H.C.: Curvature scale space corner detector with adaptive threshold and dynamic region of support. In: Pattern Recognition, International Conference on. Volume 2., IEEE Computer Society (2004) 791–794
22. Mokhtarian, F., Suomela, R.: Robust image corner detection through curvature scale space. IEEE Transactions on PAMI **20** (1998) 1376–1381