Postprint: Asymmetric Dual-Branch Interactive Neural Network for Underwater Organism Recognition

Authors: Zhao Li, Song Wei

Date: 2020-09-28T00:00:00+00:00

Abstract

To address the challenges of low visibility, poor lighting conditions, and indistinct inter-species feature differences in underwater environments, we propose a novel asymmetric dual-branch underwater biological classification model based on convolutional neural networks. The interaction branch in the model leverages different intermediate layers of the convolutional neural network to extract local features and employs an interaction module to facilitate feature interaction, thereby enhancing the local feature learning capability of the classification model; the convolutional neural network branch effectively learns the global features of the target, compensating for the global information overlooked by the interaction branch. Experimental results on three datasets—Fish4-Knowledge (F4K), EILAT, and RAMAS—demonstrate accuracies of 98.9%, 98.3%, and 97.9%, respectively, representing significant improvements over prior methods. Visual explanations further validate that the model can effectively capture local features while eliminating background interference. These results collectively demonstrate that the proposed model achieves favorable classification performance in underwater environments.

Full Text

Preamble

Vol. 38 No. 3
Application Research of Computers
Accepted Paper

Asymmetric Two-Branch Interactive Neural Network for Underwater Biological Recognition

Zhao Li, Song Wei
(School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214000, China)

Abstract: To address the challenges of low visibility, poor illumination conditions, and subtle inter-species feature differences in underwater environments, this paper proposes a novel asymmetric two-branch underwater biological classification model based on convolutional neural networks. The interactive branch in the model utilizes different intermediate layers of convolutional neural networks to extract local features and performs interaction on these local features through an interactive module, thereby enhancing the local feature learning capability of the classification model. The convolutional neural network branch effectively learns the global features of the target, compensating for the global information overlooked by the interactive branch. The model achieves accuracies of 98.9%, 98.3%, and 97.9% on the Fish4Knowledge (F4K), EILAT, and RAMAS datasets, respectively, representing significant improvements over previous methods. Visual explanations also verify that the model can effectively capture local features while eliminating background influence. The results demonstrate that the proposed model exhibits excellent classification performance in underwater environments.

Keywords: underwater biological classification; asymmetric branch; interactive branch; interactive module; local features; convolutional neural network branch; global features

0 Introduction

Marine organisms play a crucial role in human life and represent valuable resources. After decades of investigation and research by marine experts and scholars, Chinese jurisdictional waters have recorded over 20,278 marine biological species, encompassing 5 biological kingdoms and 44 phyla, accounting for 10% of the world' s total marine biological species and 50% of the total quantity. Underwater biological recognition has broad applications in research, development, and management of aquatic, biological, and marine environments. Establishing databases for various organisms and employing artificial intelligence methods for automatic recognition not only facilitates the exploitation and utilization of marine biological resources but also plays an important role in marine fishery production, holding significant academic and economic value.

Traditional machine learning approaches for species recognition typically involve: image acquisition, feature extraction, classifier construction, and classification using the extracted features. For instance, Phenoix et al. [1] employed a Bayesian and Gaussian kernel mixture model for hierarchical classification of

fish features. Du Weidong et al. [2] proposed a method extracting wavelet packet coefficient singular values, time-domain centroids, and discrete cosine transform coefficients from multi-directional acoustic scattering data, performing feature fusion before classification using SVM. Although such methods have made significant progress in computer vision-based marine biological classification research, they suffer from obvious shortcomings: classifier performance heavily depends on whether manually designed features are reasonable, yet human feature selection often relies on experience, exhibiting considerable blindness and uncertainty.

In contrast to traditional machine learning methods, deep learning approaches emerging in recent years can automatically learn features from large amounts of data through convolution operations, effectively solving the problem of manual feature selection and becoming the preferred solution for many computer vision problems. For example, Abdelouahid et al. [3] and Gu Zhengping et al. [4] both proposed fish recognition methods using deep convolutional neural network models. Although these methods achieve good performance, they still have significant limitations. First, feature information loss occurs during transmission in convolutional neural networks, and these models focus on classifying the output of a single convolutional layer, thereby losing some important classification information. Second, in underwater environments with insufficient illumination, convolutional neural networks are easily affected by background interference. Both information loss and background influence lead to degraded classification performance, necessitating large amounts of additional annotation information during training to achieve satisfactory results. However, data annotation is time-consuming and expensive, making it difficult to satisfy in practical applications and imposing considerable limitations.

To address these problems, this paper proposes a novel asymmetric two-branch interactive neural network based on CNN with the following structures and characteristics: (a) Interactive branch: employs intermediate layers of convolutional neural networks to extract image features, then integrates local features learned by different intermediate layers through an interactive module to enhance the interactive branch' s ability to capture and learn local features, effectively compensating for feature information loss during transmission. (b) Convolutional neural network branch: can effectively capture the global information of targets (such as shape and appearance), making up for the interactive branch' s neglect of global information. (c) The two branches are combined through a fusion layer, enabling robust capture and learning of both local and global feature information of targets while distinguishing targets from background and eliminating background influence, even in underwater environments with poor illumination. This significantly improves classification performance. The proposed model effectively overcomes the deficiencies of existing traditional machine learning and deep learning models, demonstrating superior classification performance on all three datasets compared to other models.

1.1 Local Feature Learning

Feature learning constitutes a crucial component of the image classification process. Compared to differences between basic categories (e.g., cats and dogs), differences between different subcategories within the same basic category (e.g., different coral species [18]) are extremely subtle and exist only in local features of target images (such as coral crowns, blades, fish fins, tails, abdomens, etc.). These subtle feature differences are difficult for ordinary fully connected neural layers to parse, often limiting the performance of conventional neural network models in recognition tasks [5]. To address this issue, Zhang et al. [32] proposed an algorithm that selects discriminative local features from convolutional features, using Selective Search to generate candidate local regions, then employing MMP (Multi-max pooling) to directly generate local features from candidate regions, clustering these features and calculating the importance of each cluster to select important clusters as the final image local feature representation. Perronnin et al. [33] used FV (Fisher Vector) encoding to represent all candidate local features of a target image as a single vector, employing a Gaussian mixture model (GMM) to cluster candidate local features and calculating the mutual information values of each class to select important local features for network learning. Simon et al. [34] utilized keypoints generated from convolutional network features to extract local feature information based on these keypoints. Although these methods can effectively extract local features, they all employ the Selective Search method to generate candidate local regions and require calculating the importance among clusters, facing enormous computational costs.

Lin et al. [6] proposed a method fusing output features from two independent CNNs by taking the outer product of the two CNN output feature vectors to generate high-dimensional features for classification through fully connected layers. Kong et al. [7] further reduced computational complexity by applying low-rank approximation to the covariance matrix. Maji et al. [8] proposed matrix square root normalization, further improving classification performance. Wei et al. [9] argued that commonly used 1$×$1 convolution kernels for dimensionality reduction would decrease the diversity of reduced features, thus employing P singular vectors for dimensionality reduction. Gao et al. [29] used Tensor Sketch to unify second-order information and reduce feature dimensions. Cui et al. [20] subsequently used Tensor Sketch to aggregate higher-order information. Gou et al. [10] obtained features containing both first-order and second-order information through feature matrix augmentation and performed fusion operations using Tensor Sketch. However, these methods only consider processing output features from a single convolutional layer, while experiments reveal that different convolutional layers in CNNs learn different features, and significant information loss occurs when these features pass through different layers. Consequently, output feature maps from a single convolutional layer cannot adequately represent subtle differences between local features.

The proposed method utilizes multiple convolutional layers to extract image

features and integrates local information captured by each convolutional layer through interaction, thereby enhancing the model' s ability to capture and learn subtle local features.

1.2 Convolutional Neural Networks

Due to the success of deep learning in various fields in recent years, CNNs (Convolutional Neural Networks) have become universal feature extractors for various visual recognition tasks. Chatfield et al. [11] evaluated CNN performance based on VGGnet for image classification and compared it with previous feature encoding methods, demonstrating that deeper CNNs outperform shallower CNN models trained on augmented data. Despite influences from lighting, color, angle, and other factors, CNNs have proven their advantages in image classification [12–14]. However, these methods have obvious defects: when extracting target features, CNNs are often susceptible to background influence, mistakenly treating background noise as the target for information extraction. Therefore, training requires incorporating manually crafted features such as shape, color, and texture to enhance the model' s ability to distinguish targets from background. This dependence on handcrafted features makes these methods difficult to apply to large-scale datasets and imposes certain limitations. In contrast, the classification method proposed in this paper can effectively eliminate background influence without relying on any manual feature information and can be applied to any underwater image dataset.

2 Asymmetric Two-Branch Network

This chapter proposes an asymmetric two-branch network for underwater species classification based on convolutional neural networks. Without relying on any manual feature information, the network can eliminate background influence and capture subtle local features, making it suitable for biological recognition data in various underwater scenarios.

2.1 Asymmetric Two-Branch Structure

Unlike ordinary basic category recognition, different subcategories within the same basic category typically have similar appearances, with more subtle inter-category differences. Subcategory recognition can only be distinguished through minor local feature differences. Therefore, how to extract and effectively learn local feature information becomes the key to successful subcategory recognition algorithms. However, most convolutional neural network models focus solely on feature learning using single convolutional layers while completely ignoring information loss that occurs when feature information is transmitted between different layers. Consequently, the feature information learned by each convolutional layer is incomplete. To capture more local features, this method incorporates an interactive non-branch based on convolutional neural networks. As shown in [Figure 1: see original paper], the model consists of an interactive branch and

a CNN branch. The interactive branch uses three feature extractors primarily composed of convolutional layers to extract features from images, then inputs features extracted by different extractors into interactive modules to enhance information interaction between different features. Compared to basic category recognition, subcategory recognition places greater emphasis on local feature learning, with lower signal-to-noise ratios in image information, making it more susceptible to lighting, pose, background, and other factors. The CNN branch can effectively extract global information of target images (such as target shape and appearance), enhancing the model' s ability to locate targets in images and eliminating influences from lighting and background factors, thereby compensating for the interactive branch' s focus on local information while neglecting global information. The outputs of the two branches are ultimately integrated through a fusion layer with weighted summation.

2.2 Interactive Branch

The interactive branch consists primarily of feature extractors and interactive modules, with the purpose of capturing subtle local features in target images. These small local features are highly representative in classification and can therefore effectively improve model classification performance.

2.2.1 Interactive Branch Factorization Kim et al. [16] proposed factorization using the Hadamard product for effective attention mechanisms in multimodal learning. This subsection briefly introduces the basic formulas of factorization. Assuming an image I is filtered by a convolutional layer-based feature extractor, the extractor' s output has height, width, and channels; the spatial dimension descriptor in the output is represented as a feature map $x = [x_1, x_2, ..., x_c]^T$. The interaction model can be defined with weight matrices, producing model output. According to the matrix factorization proposed by Rendel [17], the equation can be decomposed into two one-dimensional vectors. Assuming the interactive branch output is $Z = [z_1, z_2, ..., z_o]$ of dimension o, then U and V are weight matrices of different interactive modules, $\odot$ denotes the Hadamard product, and d is a definable scale parameter determining interactive layer performance and computational complexity.

2.2.2 Interactive Module The interactive module aims to enhance interaction between feature maps extracted by different feature extractors. First, features from different extractors are expanded into a high-dimensional space through independent nonlinear mappings to facilitate convolutional layers in capturing features of different target parts. Then, element-wise integration is performed using the Hadamard product to enable interaction between different local features. Finally, summation is executed to compress high-dimensional features into compact features.

In a single interactive module, for different features at spatial dimension i, in-

teraction using the Hadamard product can be defined as:

$$z_i = U_i^T x \odot V_i^T y$$

where x and y are features extracted from different extractors, and U_i and V_i are mapping matrices. Finally, summation is performed over the feature matrix across the entire space to compress high-dimensional features into a compact feature vector. Assuming the spatial dimension is o, this can be written as:

$$Z = \sum_{i=1}^{o} z_i$$

Multiple interactive modules are added to the branch to integrate multiple features, further enhancing the expressive power of feature information in classification. Assuming features from different extractors are x, y, z, for a branch with multiple interactive modules, the interactive branch output, i.e., the extracted local features, is computed with d as the neural layer size parameter in the interactive module that determines performance.

2.2.3 Feature Extractors As shown in [Figure 2: see original paper], since convolutional layers inherently possess feature extraction capabilities, intermediate convolutional layers from the convolutional neural network are extracted, with nonlinear functions (ReLU) and normalization (batch normalization) added to serve as feature extractors in this model.

2.3 Convolutional Neural Network Branch

The interactive branch described above focuses primarily on local feature learning, making it prone to neglecting global information (such as target shape and appearance), resulting in weak target localization capability and susceptibility to background and lighting factors during recognition. Therefore, global information also plays a vital role in subcategory recognition. Although ordinary convolutional neural networks have relatively weak ability to extract and learn subtle local features, they can effectively capture global information of target images. Consequently, complete convolutional and pooling layers are retained in the other branch to extract global information and compensate for the interactive branch' s neglect of global information. Different weights are assigned to the outputs of the two branches in the fusion layer for integration:

$$Z_{output} = w_1 Z_{interact} + w_2 Z_{object}$$

where Z_{object} represents global information extracted by the convolutional neural network, and w_1 and w_2 are weights corresponding to local and global information, respectively, summing to 1 to control the proportion of local and global information in the final classification.

3 Experiments

In the experiments, three most commonly used datasets were employed to evaluate model performance, with comparisons provided against previous methods. Subsequently, each component of the proposed model was evaluated individually, and finally, visual explanations were used to intuitively interpret the model.

3.1 Datasets

Three most commonly used datasets in the underwater biological domain were adopted:

EILAT Dataset [18]: This dataset comprises image patches extracted from full-size images captured by the same camera, containing 1,123 images. All are 64$×$64 pixel full-size images captured by cameras during coral reef surveys near EILAT Island in the Red Sea, labeled by experts into 8 categories. This dataset employs 10-fold cross-validation, where 90% of images constitute the training set and the remaining 10% serve as the test set.

RAMAS Dataset [18]: This dataset was collected by the Rosenstiel School of Marine and Atmospheric Science during coral reef surveys, containing 766 images labeled by experts into 14 categories, with each image sized 256$×$256. This dataset uses the same cross-validation method as the EILAT dataset.

Fish4Knowledge (F4K) Dataset [19]: This dataset comprises video data collected by Taiwan Power Company, Taiwan Ocean Research Institute, and Kenting National Park at underwater observation platforms in Nanwan, Lanyu, and Hubihu, Taiwan, between October 1, 2010, and September 30, 2013. It contains 27,370 underwater fish images of 23 fish species, with sizes ranging from 20$×$20*to*200$×$200. For this dataset, 80% constitutes the training set and the remaining 20% serves as the test set.

3.2 Experimental Setup

VGG-16 pretrained on the ImageNet classification dataset was selected as the convolutional neural network branch in the model (the asymmetric two-branch model can also use other CNN architectures such as Inception, ResNet, etc.). The last three fully connected layers were removed, and the proposed interactive branch was added. Input images were uniformly resized to 224$×$224. To demonstrate that model performance does not rely on sophisticated preprocessing, only the simplest data augmentation methods were applied, such as random horizontal flipping and random translation. During training, the entire model was optimized using stochastic gradient descent (SGD) with a batch size of 16, momentum set to 0.9, weight decay of 5×10^{-3}, and learning rate of 10^{-3}, which was reduced by a factor of 10 when learning stagnated. In the fusion layer, the weights w_1 and w_2 controlling the proportion of local and global information were both initialized to 0.5 to ensure balanced proportions, then adjusted during

experiments. All experiments were implemented using the Google deep learning framework TensorFlow.

3.3 Intermediate Layer Selection

The CNN intermediate layers serving as feature extractors determine whether captured local features are representative for classification, making layer selection crucial. Taking VGG16 and VGG19 as examples, each neural layer before a pooling layer was considered a complete convolutional module (each module contains 2–3 convolutional layers). The output of each convolutional module was separately input into a softmax layer for classification, with accuracies on the three datasets shown in [Figure 3: see original paper]. The results demonstrate that as network depth increases in CNNs, the feature level extracted by convolutional layers becomes higher and more representative for classification. Therefore, deeper layers extract features more beneficial for classification than shallow layers, consistent with conclusions from previous research [20]. Consequently, intermediate layers from the last convolutional module were selected as feature extractors. Under the condition of using identical standard split methods for all datasets, the outputs of three feature extractors (three intermediate layers extracted from the CNN) and the convolutional neural network branch output were each input into fully connected layers for classification testing and compared with the complete model (fusion layer).

To verify that the proposed model can use different categories of CNNs, two completely different CNNs were used in experiments. Comparison results are shown in . Notably, the third extractor performed significantly worse than the first two, leading to the inference that although deeper convolutional layers can extract more abstract features beneficial for classification, obvious feature information loss occurs during information transmission between layers as depth increases, resulting in decreased classification performance. This inference was also confirmed in subsequent visual explanation experiments. Therefore, improving classification performance requires better learning of multiple feature information sources.

3.4 Interactive Module Mapping Size Parameter and Quantitative Analysis

Equation (6) mentions the neural layer size parameter d in the interactive module that determines interactive performance. To select an appropriate d, experiments were conducted on the EILAT dataset using different module combinations, with results shown in [Figure 4: see original paper]. Here, I(1,2)+CNN denotes fusing the outputs of extractors 1 and 2 through an interactive layer and then combining with the CNN branch output for classification; I(1,3) and I(2,3) follow the same notation; F+CNN represents I(1,2)+I(1,3)+I(2,3)+CNN.

[Figure 4: see original paper] reveals several key findings. First, regardless of the d value, the F+CNN combination consistently outperforms other combinations,

indicating that multiple interactive modules can enhance classification performance. Second, as size d increases from 64 to 512, performance improves for all combinations, but decreases when d reaches 1024. Considering that excessively large d values produce higher computational complexity while overly small d values degrade module performance, $d = 512$ was selected as the optimal size for all subsequent experiments.

For quantitative analysis of interactive modules, combinations containing 1–3 interactive modules were evaluated on the RAMAS and EILAT datasets, with results shown in . Comparing the first six items with the last item demonstrates that increasing the number of interactive modules significantly improves classification performance. Additionally, comparing the last two items shows that performance improves after fusing the interactive branch output with the CNN branch output, proving that the independent interactive branch overemphasizes local information while neglecting global information, and that the global information provided by the CNN branch plays an important role in classification.

3.5 Intermediate Layer Performance Analysis

To demonstrate that interaction and integration of local feature information can improve classification performance, the outputs of three feature extractors (intermediate layers from the CNN) and the convolutional neural network branch output were each input into fully connected layers for classification testing and compared with the complete model (fusion layer) under identical standard dataset split conditions.

3.6 Fusion Layer Weight Analysis

Equation (7) mentions that weights w_1 and w_2 in the fusion layer control the proportion of local and global information in the final classification, with their sum always equal to 1. To select appropriate values, experiments were conducted on the three datasets using different w_1 values, with results shown in [Figure 5: see original paper].

[Figure 5: see original paper] reveals two key observations. First, when the local information weight w_1 ranges from 0 to 0.7, classification accuracy improves as w_1 increases, indicating that local information is crucial in the classification process and that incorporating local information significantly enhances model performance. Second, when w_1 ranges from 0.7 to 1, classification accuracy decreases as the global information weight w_2 decreases (since $w_2 = 1 - w_1$), indicating that neglecting global information causes the model to be affected by background and lighting factors during classification, leading to performance degradation. Therefore, incorporating an appropriate proportion of global information can eliminate these influences and further improve classification performance. Considering that excessively small w_1 reduces the model' s ability to learn local information while excessively large w_1 causes the model to ignore global information, the optimal values were selected as $w_1 = 0.7$ and $w_2 = 0.3$.

3.7 Classification Results Comparison

This section compares the performance of the interactive two-branch model with previous methods on each dataset. It should be noted that different methods and models have different learning capabilities, and thus learn different visual features. Whether a model can achieve good classification performance depends on its ability to learn key visual features. shows the results of the interactive two-branch network and the current highest accuracy results on the RAMAS and EILAT datasets. VGG16 and ResNet-50 were both pretrained on ImageNet and then trained on the datasets. Notably, the method in item 8 also relied on handcrafted features during training, while item 9 represents the current state-of-the-art, achieving the highest accuracy but focusing only on local features while neglecting global features of target images. Consequently, classification performance is affected by background in underwater environments with poor lighting conditions. The last item in shows that the interactive two-branch network, which does not rely on any handcrafted features, significantly outperforms other methods and achieves the highest classification accuracy.

shows the interactive two-branch network results and current highest accuracy results on the F4K dataset. VGG16 and ResNet-50 were both pretrained on ImageNet and then trained on the dataset. Wei et al. [27] used fish names from F4K as keywords to download clearer images from Google search engine, manually removing and cropping erroneous images and boundary regions to construct a high-quality dataset, achieving 97.3% accuracy on the high-quality dataset but only 17.14% on the original F4K dataset. Zhang Junlong et al. [31] also divided fish images for each species into three sub-datasets (high, medium, and low quality) based on clarity and background influence, achieving 97.0% accuracy on the high-quality dataset and 94% and 90% on medium- and low-quality datasets, respectively. Gu Zhengping et al. [4] adopted a transfer learning CNN+SVM method, achieving 98.6% accuracy—the highest currently reported on this dataset. Compared with previous methods, the interactive two-branch network shows significant accuracy improvements and does not rely on any manually extracted or crafted features.

For better model interpretation, the Grad-CAM [28] algorithm was used to provide visual explanations for outputs from different layers, as shown in [Figure 6: see original paper]. The heatmap regions represent areas of interest to the model during classification, i.e., the model' s attention regions. Deeper heatmap colors indicate greater contribution and proportion in classification. [Figure 6: see original paper] demonstrates that in underwater scenes with poor lighting conditions, ordinary convolutional layers are easily affected by target background during feature extraction due to weak color differences, mistakenly treating background as target for feature extraction. Additionally, it shows that convolutional layers have certain local feature capture ability during feature extraction and learning, but different convolutional layers focus on significantly different regions, and obvious feature information loss occurs during information transmission. These lost local features (such as fish heads, tails, abdomens,

coral crowns, blades, etc.) play crucial roles in classification. This also proves the inference in Section 3.5 that feature information loss leads to degraded classification performance. The proposed model significantly improves feature information utilization by promoting interaction between different features and integrating them, effectively eliminating background influence caused by weak underwater lighting.

4 Conclusion

This paper investigated underwater biological classification algorithms based on deep learning and proposed an asymmetric interactive two-branch network with interaction and integration modules based on convolutional neural networks. Through experiments, the highest accuracy was achieved on the three most commonly used underwater biological datasets, fully demonstrating that the model does not rely on any manual methods or require domain knowledge about marine organisms to achieve good classification performance. Future work will extend this research to explore how to more effectively learn and integrate multi-layer features to achieve even better classification performance in scenarios with harsher environmental conditions and lower image quality.

References

[1] Huang P X, Bastiaan B, Fisher R. Hierarchical classification with reject option for live fish recognition. Machine Vision and Applications 2015, 26 (1): 89-102.
[2] Du Weidong, Li Haifeng, Wei Yukuo. Multi-azimuth acoustic scattering data cooperative fusion using SVM for fish classification and identification [J]. Transactions of The Chinese Society of Agricultural Machinery, 2015, 61 (3): 39-43.
[3] Tamou A B, Benzinou A, Nasreddine K, et al. Underwater Live Fish Recognition by Deep Learning [C]. International Conference on Image and Signal Processing. Springer, Cham, 2018, 171 (6): 275-283.
[4] Gu Zhengping, Zhu Min. Fish classification algorithm based on deep learning [J]. Computer Application and Software, 2018, 35 (1): 200-205.
[5] Yandex A B, Lempitsky V. Aggregating local deep features for image retrieval [J]. IEEE International Conference on Computer Vision, Computer Science. 2015: 1269–1277.
[6] Lin Tsungyun, Roychowdhury A, Maji S. Bilinear CNN models for fine-grained visual [C]// IEEE International International Conference on Computer Vision, 2015: 1449-1457.
[7] Kong shu, Charless F. Low-rank bilinear pooling for fine-grained classification [C]// IEEE Conference on Computer Vision and Pattern Recognition, 2017: 7025–7034.
[8] Lin Tsungyun, Maji S. Improved bilinear pooling with CNNs [C]// The British Machine Vision Conference, 2017. 2005, 13 (22): 8766-8771.
[9] Wei Xing, Zhang Yue, Gong Yihong, et al. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification [C]// Euro-

pean Conference on Computer Vision, Computer Vision. 2018: 365-380.
[10] Gou Mengran, Xiong Fei, Octavia I C, et al. MoNet: Moments embedding network [C]// Conference on Computer Vision and Pattern Recognition, 2018: 3175-3183.
[11] Ken C, Karen S, Andrea V, et al. Return of the devil in the details: Delving deep into convolutional nets [C]. /The British Machine Vision Conference, Computer Science, 2014.
[12] Khan H, Munawar H, Mohammed B, et al. Cost Sensitive Learning of Deep Feature Representations from Imbalanced Data. IEEE Transactions on Neural Networks and Learning Systems, 2017.
[13] Mahmood A, Bennamoun M, An Senjian, et al. Deep image representations for coral image classification [J]. IEEE Journal of Oceanic Engineering, 2018: 121-131.
[14] Uzair N, Mohammed B, Ferdous S, et al. Deep Fusion Net for Coral Classification in Fluorescence and Reflectance Images [J]. Digital Image Computing: Techniques and Applications, 2019.
[15] Rathi D, Indu S, Jain S, et al. Underwater fish species classification using convolutional neural network and deep learning [J]. International Conference of Advances in Pattern Recognition, 2017.
[16] Kim J H, On K W, Kim J, et al. Hadamard product for low-rank bilinear pooling [J]. International Conference on Learning Representations. 2017.
[17] Rendle S. Factorization machines [J]. Interational Conference on Data Ming, 2010: 559-1000.
[18] Shihavuddin A, Gracias N, Garcia R, et al. Image-based coral reef classification and thematic mapping [J]. Remote Sens, 2013, 5: 1809-1841.
[19] Boom B J, Huang P X, He Jiyin, et al. Supporting ground-truth annotation of image datasets using clustering [C]// International Conference on Pattern Recognition. IEEE, 2012: 1542-1545.
[20] Cui Yin, Zhou Feng, Wang Jiang, et al. Kernel Pooling for Convolutional Neural Networks [C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition. [S. I.] : IEEE, 2017.
[21] Oscar B, Peter J E, David I K, et al. Automated annotation of coral reef survey images [C]// IEEE Conference on Computer Vision and Pattern Recognition, [S. I.] : IEEE, 2012: 1170-1177.
[22] Marcos S, Saloma C A, Soriano M, et al. Classification of coral reef images from underwater video using neural networks [J]. Opt Express, 2005, 13 (22): 8766-8771.
[23] Stokes M D, Deane G B. Automated prosessing of coral reef benthic images [J]. Limnol Oceanogr Meth. 2009. 7 (2): 157-168.
[24] Oscar P, Paul R, Johnson-Roberson M, et al. Colquhoun Towards image-based marine habitat classification [C]// OCEANS 2008, IEEE, 2008: 1-8.
[25] Mary N A B, Dharma D. Coral reef image classification employing Improved LDP feature extraction [J]. Journal of Visual Communication & Image Representation, 2017, 49 (nov.): 225-242.
[26] Qin Hongwei, Li Xiu, Liang Jian, et al. DeepFish: Accurate underwater live fish recognition with a deep architecture [J]. Neurocomputing, 2016, 187: 49-58.

[27] Wei Guanqun, Wei Zhiqiang, Huang Lei, et al. Robust Underwater Fish Classification Based on Data Augmentation by Adding Noises in Random Local Regions. [C]// Pacific Rim Conference on Multimedia 2018: 509-518.
[28] Ramprasaath R. Selvaraju, Michael C, et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization [C]// International Conference on Computer Vision, 2017: 1-24.
[29] Gao Yang, Beijbom O, Zhang Ning, et al. Compact bilinear pooling [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 317-326.
[30] Ammar M, Mohammed B, An Senjian, et al. ResFeats: Residual network based features for underwater image classification [J]. Image and Vision Computing, 2020. 1: Article ID 103811.
[31] Zhang Junlong, Zeng Guosun, Qin Rufu. Fish recognition method for submarine observation video based on deep learning [J]. Journal of Computer Applications, 2019, 39 (2) 72-77.
[32] Zhang Yu, Wei Xiushen, Wu Jianxin, et al. Weakly supervised fine-grained categorization with part-based image representation [J]. IEEE Transactions on Image Processing, 2016, 25 (4): 1713−1725.
[33] Perronnin F, Dance C. Fisher kernels on visual vocabularies for image categorization [C]// Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition. Minneapolis, USA: IEEE, 2007. 1−8.
[34] Simon M, Rodner E. Neural activation constellations: unsupervised part model discovery with convolutional networks [C]// Proceedings of the 15th IEEE International Conference on Computer Vision. Santiago, Chile: IEEE, 2015. 1143−1151.

Note: Figure translations are in progress. See original paper for figures.