ICES Journal of Marine Science, 2023, **80**, 1256–1266
DOI: 10.1093/icesjms/fsad041
Advance access publication date: 3 April 2023
Original Article

Improving the classification accuracy of fishes and invertebrates using residual convolutional neural networks

Z. Zhou [1,*], X. Yang [1], H. Ji [1], and Z. Zhu [2]

[1]School of Computer Science and Technology, Zhejiang Sci-Tech University, 840 Xuelin Street, Jianggan District, Hangzhou, Zhejiang, China
[2]School of Mechanical Engineering, Hangzhou Dianzi University, 188 Xuelin Street, Jianggan District, Hangzhou, Zhejiang, China
[*] *Corresponding author. tel: +86-057186843323;* e-mail: zhouzhiyu1993@zstu.edu.cn.

The visibility of fishes and invertebrates is highly impacted by the complexity of the environment. Images acquired in underwater environments suffer from blurriness and low contrast. This results in a low classification accuracy. To address this problem, this study uses a pre-trained Resnet50 neural network as the feature extractor, which avoids over-fitting and accuracy saturation while realizing improved feature extraction capabilities. It also proposes an enhancement of the error-minimized random vector functional link (EEMRVFL) neural network, which is used as the classifier in the convolutional neural network (CNN) model instead of the original softmax classifier. EEMRVFL reduces the maximum residual error in each incremental process. The selected hidden nodes are added to the network, which improves the compactness of its structure. The proposed residual CNNs model exhibits improved classification accuracy for underwater image classification compared to existing methods. This is demonstrated experimentally on available datasets such as URPC, LifeCLEF 2015, and Fish4Knowledge with accuracy rates reaching 99.68%, 97.34%, and 99.77%, respectively.

Keywords: classification, random vector functional link, Resnet50, underwater image.

Introduction

Advances in science and technology have driven the development of underwater robots, which have strongly contributed to accelerated advancements in marine life research and exploitation of marine resources (Beyan and Browman, 2020; Malde *et al.*, 2020; Chandran *et al.*, 2021; Qiao *et al.*, 2021). In the presence of risk factors and uncertain complexities associated with underwater operations, the use of underwater robots has become increasingly important for various applications in underwater safety, search and rescue, equipment inspection and maintenance, biological research, and investigation of marine environments (Xie *et al.*, 2018; Duan *et al.*, 2021; Zhuang *et al.*, 2021; Chen *et al.*, 2022). Underwater imaging constitutes the main source of information collected by robots for further investigations. The classification of fishes and invertebrates is helpful to obtain and study the quantity and distribution of various aquatic animals and plants and to monitor the changes of underwater creatures, which are often closely related to environmental climate and so on. For example, the acidification of seawater caused by global warming directly affects the population of Antarctic krill and offshore shellfish. Nevertheless, the quality of underwater-collected images is highly degraded owing to the complexity of the underwater environment, which makes image classification highly challenging. Conventionally, after image pre-processing and segmentation, features are extracted by grayscale or texture feature-based methods and principal component analysis (Li *et al.*, 2017); this is followed by a final feature transformation step to classify the target image.

Convolutional neural networks (CNNs; Simonyan and Zisserman, 2015; Szegedy *et al.*, 2016; Krizhevsky *et al.*, 2017; Zhang *et al.*, 2020) have promising potential for application in underwater operations (Lu *et al.*, 2020; Salman *et al.*, 2020; Muhammad *et al.*, 2021). Sung *et al.* (2017) proposed a YOLO detection method based on a CNN capable of prompt and accurate fish detection, outperforming conventional HOG and support vector machine (SVM) methods. Xu *et al.* (2017) and Jin *et al.* (2017) used relatively limited datasets for learning and testing CNNs pre-trained on ImageNet to classify images through transfer learning. To increase the amount of training data, Allken *et al.* (2019) developed a deep vision system for data augmentation, achieving a 94% classification accuracy using pre-trained Inception 3. Banan *et al.* (2020) used a pre-trained VGG network to achieve a 100% classification accuracy for four types of carp. Although their work showed that the model could effectively extract the visual features of fish, the dataset used in experiments was small, and the background was simple. Siddiqui *et al.* (2018) and Mahmood *et al.* (2019) proposed an approach based on fusing features of different layers of CNNs, which improved the classification accuracy but at the expense of high computational resources and costs. Yeh *et al.* (2020) applied an attention mechanism in a CNN to enhance the features obtained by the convolutional layer at the end of the model. This improved the classification accuracy by increasing the amount of obtained information for classification. Deep and Dash (2019) used a hybrid CNN model with traditional classifiers, which demonstrated a better accuracy compared to conventional deep learning methods. Nonetheless, the performance can be improved further through image pre-processing and other techniques. Rathi *et al.* (2017) proposed an algorithm model integrating deep learning and image processing. Pre-processed and original images were jointly trained through the CNN, and the experiment was carried out on the Fish4Knowledge dataset,

Received: 2 January 2023; **Revised:** 14 February 2023; **Accepted:** 1 March 2023

achieving a classification accuracy of 96.29%. Qin *et al.* (2016) used a method for segmenting the foreground and background of fish images, extracting features through deep structures via principal component analysis and using spatial pyramids to process the features. Although a satisfactory accuracy was realized on the Fish4Knowledge dataset, this method resulted in low accuracy and long calculation times when applied to high-dimensional images.

A CNN uses the classical back-propagation learning algorithm to minimize the cross-entropy loss. It usually has good classification results on the majority class when datasets with highly imbalanced training data are considered; however, the minority class is often misclassified. It has been recently shown that using a CNN as a feature extractor and selecting traditional classifiers for classification tasks can yield better classification results (Geng *et al.*, 2016; Meng *et al.*, 2019). However, traditional classifiers, such as SVMs, suffer from several problems associated with a large number of adjustable parameters and low learning speeds.

The random vector functional link (RVFL) network is utilized to classify image features for realizing higher classification accuracy. In addition, as proposed by Pao *et al.* (1994), RVFL randomly generates the parameters of the neural network and solves the output weights by the least-squares method. This significantly improves training speeds compared to conventional feedforward neural networks. Although RVFL does not require all the parameters to be adjusted, several studies have shown that the number of nodes in the hidden layer has a significant impact on the network performance (Qiu *et al.*, 2018; Zhou *et al.*, 2020). An exceedingly small number of nodes leads to inappropriate network training, whereas an exceedingly large number of nodes causes over-fitting problems and significantly increases the training time. However, methods relying on a manual approach for adjusting the number of nodes in the hidden layer are often selected based on experience. These methods usually provide poor results and are highly time-consuming. Huang *et al.* (2006) proposed an incremental extreme learning machine (IELM) algorithm, which does not require a manual adjustment of the number of nodes in the hidden layer. It updates the output weights and determines the network structure by randomly generating hidden layer nodes and adding them one by one. IELM has been proven as a universal approximator. To address the problem of low speeds caused by sequentially adding hidden layer nodes in IELM, Feng *et al.* (2009) proposed an error-minimization extreme learning machine (EMELM). EMELM adds random hidden layer nodes to the network group by group, with a fixed or variable group size. As the network grows, its output weight is constantly updated, which significantly reduces the computational complexity. However, randomly generated hidden layer nodes by EMELM may have a negligible effect on the network output while unnecessarily increasing computational complexity.

We make the following contributions in this paper: Aiming at improving the performance in this context, we propose a classification algorithm model combining Resnet50 and an enhancement of the error-minimization random vector functional link (EEMRVFL). Compared with some basic neural network models and some state-of-the-art deep learning methods, experiments were conducted on URPC and LifeCLEF 2015 data sets. The accuracy improvement of the algorithm proposed in this paper is shown in the form of tables and confusion matrices. In addition, the box diagram is drawn and the data is tested to verify the performance of the algorithm.

Method

Resnet50

CNNs are widely used in various feature extraction tasks. In theory, the network performance is supposed to increase with network depth. However, in practical applications, when the depth of the CNN reaches a certain level, the network performance and the convergence speed start decreasing with the deepening of layers (He and Sun, 2015; Srivastava *et al.*, 2015). He *et al.* (2016) found that residual networks can solve this problem. The main innovation of residual learning is the residual block, which can be defined by Equation (1).

$$H\ (x) = F\,(x) + x, \tag{1}$$

where $H(x)$ represents the output, $F(x)$ the residual part, and x the sample.

Resnet effectively extracts features from the input data by stacking multiple residual blocks, thus achieving high accuracy in many classification applications. Therefore, in this study, Resnet50 was chosen as the network model for feature extraction.

RVFL network

The RVFL network is a random parameter network, which calculates output weights by the least-squares method. Compared to ELM, there is a direct link in the RVFL network connecting the input and output layers.

Suppose there are N arbitrary samples (x_i, y_i), where $x_i = [x_{i1}, x_{i2}, \ldots\ldots, x_{in}]^T \in R^n$, $y_i = [y_{i1}, y_{i2}, \ldots\ldots, y_{im}]^T \in R^m$, n and m represent the dimensions of the vector. An RVFL network can be represented as

$$\sum_{j=1}^{L} \beta_j g\left(w_j X_i + b_j\right) + \sum_{j=L+1}^{L+d} \beta_j X_{ij} = o_i,\ i = 1, 2, \ldots\ldots, N, \tag{2}$$

where L is the number of neurons, $g(x)$ is the activation function, $w_j = [w_{j1}, w_{j2}, \ldots\ldots, w_{jn}]^T$ is the input weight, $b_j = [b_{j1}, b_{j2}, \ldots\ldots, b_{jn}]^T$ is the bias of the jth neuron, d is the dimension of the input data, and $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots\ldots, \beta_{jm}]^T$ is the output weight. W_j and b_j are randomly generated parameters.

To minimize the output error, $E = \sum_{i=1}^{N} (o_i - y_i)^2$, the following condition is expected to be satisfied:

$$\sum_{i\,=\,1}^{N} \|o_i - y_i\| = 0. \tag{3}$$

Using Equation (3), Equation (2) can be directly expressed as

$$\sum_{j=1}^{L} \beta_j g\left(w_j X_i + b_j\right) + \sum_{j=L+1}^{L+d} \beta_j X_{ij} = y_i,\ i = 1, 2, \ldots\ldots, N. \tag{4}$$

The matrix of Equation (4) is expressed as

$$H\beta = Y, \tag{5}$$

where H is the output of the hidden layer, β is the weight, and Y is the desired output. Their specific representations are as follows:

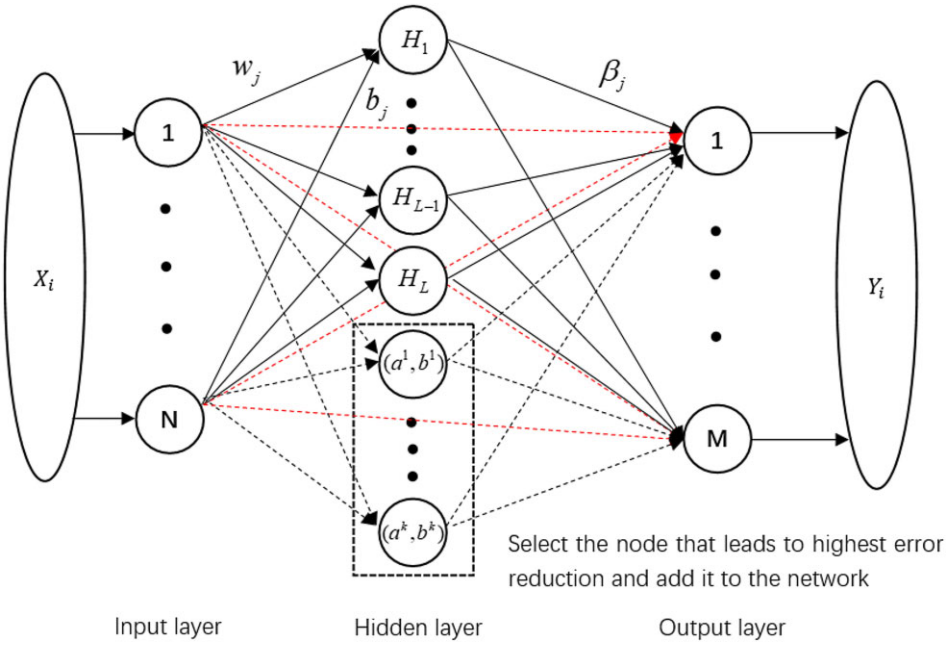

Figure 1. Network structure of enhanced error-minimized RVFL network.

$$H = \begin{bmatrix} g(w_1 X_1 + b_1) & \cdots & g(w_L X_1 + b_L) & x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g(w_1 X_N + b_1) & \cdots & g(w_L X_N + b_L) & x_{N1} & \cdots & x_{Nn} \end{bmatrix}_{N\times(L+n)}, \quad (6)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \\ \beta_{L+1}^T \\ \vdots \\ \beta_{L+n}^T \end{bmatrix}_{(L+n)\times m}, \quad (7)$$

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N\times m}. \quad (8)$$

The output weight could be calculated by $\hat{\beta} = H^{\dagger} Y$, where $H^{\dagger}$ is the Moore–Penrose inverse matrix. In the solution process, we add the orthogonal method to avoid singularity, and the calculation formula is as follows:

$$\beta = H^{\dagger} Y = \begin{cases} H^T (I/\alpha + HH^T)^{-1} T & N < L \\ (I/\alpha + H^T H)^{-1} H^T T & N \geq L \end{cases}, \quad (9)$$

where I/α is an extremely small positive value.

Enhanced error-minimized RVFL network

Aiming at the problem of some hidden layer nodes (added in the incremental process of the incremental network) exhibiting negligible effect on the network structure while increasing complexity, this paper proposes EEMRVFL. Its network structure algorithm is shown in Figure 1.

As can be seen from Figure 1, EEMRVFL has three layers. This study mainly improves the node increment process in the hidden layer. In the increment process, the EEMRVFL randomly generates multiple nodes in each incremental step and selects the node that can minimize the error. The output error is reduced while simultaneously reducing the complexity of the network. Algorithm 1 is the pseudocode of the EEM-RVFL algorithm for selecting nodes.

Algorithm 1. EEMRVFL

Initialization :
1. Given a set of initial training samples $\{(x_i, t_i)\}_{i=1}^{N}$ and the maximum number of neurons L_{max}, the expected error $\varepsilon = 1e-3$.
2. Calculate initial training error $E(H_1) = \min|H_1 \cdot \beta_1 - T_1|$, when $E(H_1) > \varepsilon$, and randomly generate k neurons for selective use.

Growing :
while $E(H_n) > \varepsilon$ and $L_n = L_{n-1} + \delta L_{n-1} < L_{max}$ (δ is a group of hidden nodes, and this paper considers the case of one node in each group)
Let $L_{n+1} = L_n + \delta L_n$
for $i = 1{:}k$
Generate a random hidden node (a^i, b^i) and add it to the existing network, where a^i and b^i are the generated parameters for the ith hidden node selection. The number of hidden nodes is L_{n+1}, and the corresponding hidden layer output matrix is

$$H_{n+1}^{\dagger} = \begin{bmatrix} X_n \\ Y_n \end{bmatrix} = \begin{bmatrix} ((I - H_n H_n^{\dagger})\delta H_n)^{\dagger} \\ (H_n^{\dagger} - H_n^{\dagger}\delta H_n^T X_n) \end{bmatrix}.$$

Update the output weight $\beta_{n+1} = H_{n+1}^{\dagger} \cdot T_{n+1} = \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \cdot T_{n+1}$
Calculate the corresponding output error
$E_i(H_{n+1}^i) = \left\| H_{n+1}^i \beta_i^{(n+1)} - T_{n+1} \right\|$
endfor
Let $i^* = \{i | min_{1\leq i\leq k} \|E_i\|\}$; hidden node (a^{i^*}, b^{i^*}) leads to the largest reduction added to the network $E = E_{i^*}$, $a_{L_j+1} = a^{i^*}$, $b_{L_j+1} = b^{i^*}$, and $\beta^{(j+1)} = \beta_{i^*}$
$n = n + 1$
endwhile

Algorithm flow

The transfer learning consists first in training the network on a large data set to obtain a model with a certain performance. Then, initial weight parameters are used to apply the network to the target data set (Liu *et al.*, 2019; Yuan *et al.*, 2020), which

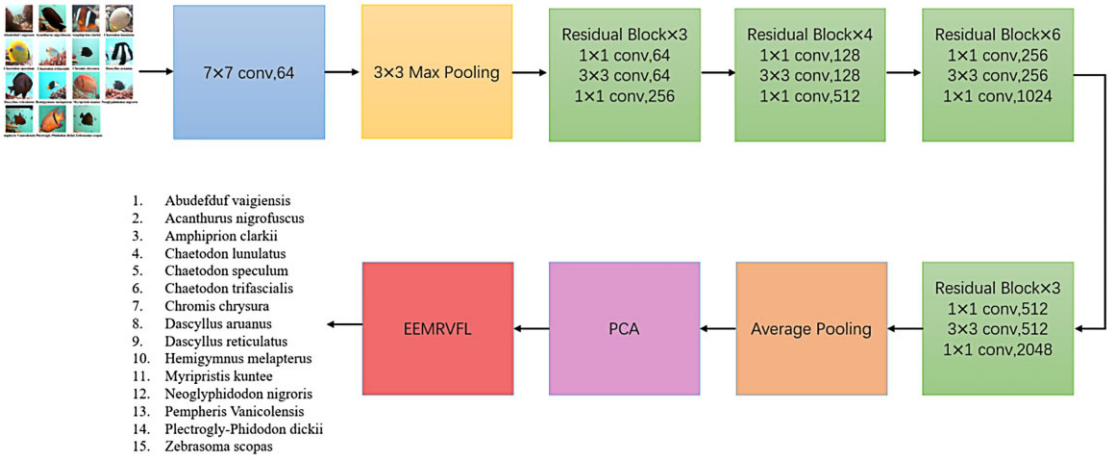

Figure 2. Proposed residual CNNs flowchart.

solves the problem of insufficient training data. Each residual block in Resnet50 contains 49 convolutional layers (i.e. a total of $1 + 3 \times (3 + 4 + 6 + 3) = 49$ convolutional layers). Learned features are passed through the fully connected layer to obtain the classification probability through the softmax layer, and the activation functions used are all ReLU functions.

In this study, the Resnet50 model pre-trained on the ImageNet dataset was used to extract features. Subsequently, the obtained features are linearly transformed by principal component analysis, which reduces the dimensionality to remove redundant information and reduce the amount of calculation. Finally, the EEMRVFL classifier is used to classify the features. The proposed residual CNNs Resnet50-EEMRVFL flowchart is shown in Figure 2. The common name of the species is shown in Supplementary Table S1.

The specific process of the EEMRVFL-Resnet50 model is as follows:

1. Change the image size to 224 × 224 × 3, and perform convolution through 64 7 × 7 convolution kernels. The size of the obtained feature map is 112 × 112 × 64.
2. For max pooling, after 3 × 3 polling units, the obtained feature map size is 56 × 56 × 64.
3. For the three residual blocks, each one of them contains 64 1 × 1, 64 3 × 3, and 256 1 × 1 convolution kernels for three convolutions + batch normalization (BN) unit + rectified linear unit (ReLU). The size after convolution is 56 × 56 × 256.
4. For the four residual blocks, each one contains 128 1 × 1, 128 3 × 3, and 512 1 × 1 convolution kernels for three convolutions + BN unit + ReLU. The size after convolution is 28 × 28 × 512.
5. For the six residual blocks, each one contains 256 1 × 1, 256 3 × 3, and 1024 1 × 1 convolution kernels for three convolutions + BN unit + ReLU. The size after convolution is 14 × 14 × 1024.
6. For the three residual blocks, each one contains 512 1 × 1, 512 3 × 3, and 2048 1 × 1 convolution kernels for three convolutions + BN unit + ReLU. The size after convolution is 7 × 7 × 2048.
7. For average pooling, the size is 1 × 1 × 2048, and the activation function is used to obtain the feature matrix output. Each row in the feature matrix represents a sample. Each sample has 2048 feature dimensions, which is reduced to 512-dimensional feature through principal component analysis.
8. The EEMRVFL classifies data based on the resulting 512-dimensional features.

Results

Six algorithms and some state-of-the-art methods were employed for comparison (i.e. Resnet50, Vgg16, Alexnet, Googlenet, Resnet50-RVFL, and Resnet50-EMRVFL) to evaluate the performance of the proposed Resnet50-EEMRVFL algorithm. An Intel 3106 CPU (1.70 GHz) with 64 GB RAM and Windows 10 operating system were used for experiments performed using the MATLAB R2021a software. In this study, the data set is randomly divided to 4:1 every time the experiment is run to obtain a reliable and stable model.

Dataset pre-processing

This study uses two datasets: LifeCLEF 2015 and URPC. The LifeCLEF 2015 dataset contains data for 15 fish species. The number of images for different species varies significantly. Therefore, in the experiment, images were rotated and flipped to increase the amount of data for fish species with fewer than 650 images. The training and test sets in the dataset were selected in a ratio of 4:1. The random selection method was adopted. The number of species after augmentation was distributed, as shown in Table 1.

When extracting features from LifeCLEF 2015 dataset, the ResNet50 network has good training accuracy and loss when applied to underwater image data after adjusting the weight and deviation learning rate of the last fully connected layer through transfer learning method. The accuracy of the final validation set is 92.22%.

The URPC dataset contains 2901 training images and 800 test images, covering four target categories, including echinus, holothurian, scallop, and starfish (Zhang *et al.*, 2020). In this study, by training the Faster R-CNN target detection model, we obtained images of four biologicals as a dataset for classification tasks. We first extracted the target image from the training set according to the officially provided bounding box and converted it to an image of 227 × 227 pixels. The training set was used to train the Faster R-CNN target detection model for conducting experiments on the test set and expand-

Table 1. Image number of each species in the LifeCLEF 2015 dataset.

Species common name	Training set	Test set
Neoglyphidodon nigroris	516	129
Chaetodon speculum	518	130
Pempheris vanicolensis	548	138
Hemigymnus melapterus	588	147
Chaetodon trifascialis	600	150
Zebrasoma scopas	650	163
Dascyllus aruanus	723	181
Abudefduf vaigiensis	732	183
Plectroglyphidodon dickii	1964	492
Chaetodon lunulatus	1995	499
Acanthurus nigrofuscus	2008	503
Amphiprion clarkii	2388	597
Myripristis kuntee	2403	601
Dascyllus reticulatus	2556	640
Chromis chrysura	2874	719

Table 2. Dataset distribution in URPC dataset.

Species	Training set	Test set
Echinus	14 768	3692
Holothurian	4170	1043
Scallop	5294	1323
Starfish	4635	1154

ing the dataset required in this study. Bad data generated by the test were deleted by manual filtering, and the segmentation of the dataset for the four species is shown in Table 2. Zhang *et al.* (2020) achieved a detection precision of 65.99% on the URPC dataset and studied the relationship between image enhancement and detection accuracy based on deep learning. According to their research conclusion, no enhancement for underwater images was been made.

Activation function

To realize enhanced classification effect, it is necessary to choose an appropriate activation function for RVFL. This study compares the classification accuracy of the proposed Resnet50-EEMRVFL on the complete URPC data set under five activation functions: sigmoid, sine, hardlim, tribas, and radbas. After ten tests on each activation function, the final sigmoid shows the most stable effect, and the accuracy is above 95%, followed by hardlim, while the classification effect of the other three functions is not ideal.

Number of neurons by step

To determine the appropriate number of incremental nodes, we set the number of randomly generated nodes in each incremental step as an array $k = \{1, 5, 10, 20\}$ for comparison across four cases. Each case went through ten experiments to compare the impact of the nodes number in the hidden layer generated in EEMRVFL. When number of neurons by step is taken as 10 or 20, the 10 test results begin to show stable characteristics. Finally, the number is selected as 20 because of the higher precision peak. The 20 alternatives give diversity to the current hidden node.

Algorithm parameter setting

Table 3 shows the training parameters of all relevant algorithms involved in this study. Except DenseNet201-FCMFDA-ELM, which has 150 iterations, the training epoch or iterations of other algorithms are set to 100. The learning rate of all CNNs is set to 4e−3, and the BatchSize is adjusted according to the size of the network model to improve the training efficiency. The input image size is set to [224, 224] except CrossVit-S as [240, 224]. Params represents the size of the network model. For the training method of networks based on optimization algorithms (Resnet50-EEMRVFL, Resnet50-RVFL, Resnet50-EMRVFL, and DenseNet201-FCMFDA-ELM), the prefix CNN model is used to extract the image features of the dataset, and then the suffix optimization classifier is used to perform label classification. In addition, for the Resnet50-EEMRVFL proposed in this paper, several hyperparameters are explained as follows: InputNeurons, fixing the classifier input feature size as 512; MaxNumberOfHiddenNeurons, set the maximum hidden layer node as 100, which is adaptively promoted by incremental operation to obtain the optimal model; and NumberOfNeuronsByStep, each incremental node has 20 options.

Table 3. Algorithm parameter setting.

Method	Hyperparameters	Params
Resnet50	Epochs = 100, BatchSize = 12, LearningRate = 4e−3	27.5
Alexnet	Epochs = 100, BatchSize = 32, LearningRate = 4e−3	0.9
Vgg16	Epochs = 100, BatchSize = 6, LearningRate = 4e−3	134.3
Googlenet	Epochs = 100, BatchSize = 32, LearningRate = 4e−3	6.3
Resnet50-EEMRVFL	Iterations = 100, InputNeurons = 512, MaxNumberOfHiddenNeurons = 100, NumberOfNeuronsByStep = 20	27.5
Resnet50-RVFL	Iterations = 100, InputNeurons = 512, HiddenNeurons = 100	27.5
Resnet50-EMRVFL	Iterations = 100, InputNeurons = 512, MaxNumberOfHiddenNeurons = 100	27.5
DCAE (Banerjee *et al.*, 2022)	Epochs = 100, BatchSize = 12, LearningRate = 4e−3	7.9
Resnet50-MPNCOV (Du *et al.*, 2020)	Epochs = 100, BatchSize = 12, LearningRate = 4e−3, $\alpha = 0.5$	24.2
CrossVit-S (Chen *et al.*, 2021)	Epochs = 100, BatchSize = 12, LearningRate = 4e−3, ImageSize = [240 224], $K = 3$, $N = 1$, $L = 1$, $M = 4$, $r = 4$	26.7
PVT2-B2 (*Wang et al.*, 2022)	Epochs = 100, BatchSize = 12, LearningRate = 4e−3,	24.9
DenseNet201-FCMFDA-ELM (Yang *et al.*, 2022)	Iterations = 150, Additional population = 3, HiddenNeurons = 100, $\beta = 1$, Particle search range = [−1,1]	27.5

Performance evaluation index

This paper measures the classification performance of the model through several aspects: accuracy, precision, recall, and *F*1-score (Marre *et al.*, 2020; Durden *et al.*, 2021). For multi-classification problems, mean average precision (MAP) can be used to represent the classification performance of the model

Table 4. Experimental results of each algorithm on the URPC dataset (optimal values are shown in bold).

Method	Accuracy (%)	Precision (%)	Recall (%)	*F*1-score (%)	MAP (%)
Resnet50	91.38	89.56	87.70	88.37	95.94
Alexnet	90.16	88.05	85.82	86.61	95.73
Vgg16	90.59	88.61	86.44	87.30	95.81
Googlenet	92.34	92.50	88.61	89.86	95.00
Resnet50-EEMRVFL	**99.68**	**99.59**	**99.59**	**99.59**	**99.32**
Resnet50-RVFL	97.78	97.11	97.70	97.40	94.62
Resnet50-EMRVFL	98.06	97.52	97.75	97.73	95.27
DCAE	97.02	96.79	95.75	96.24	98.32
Resnet50-MPNCOV	96.31	95.96	94.53	95.18	98.24
CrossVit-S	94.47	93.22	92.14	92.65	97.18
PVT2-B2	93.54	92.38	90.49	91.30	96.83
DenseNet201-FCMFDA-ELM	96.92	96.67	95.51	96.06	98.28

Table 5. Experimental results of each algorithm on the LifeCLEF 2015 dataset (optimal values are shown in bold).

Method	Accuracy (%)	Precision (%)	Recall (%)	*F*1-score (%)	MAP (%)
Resnet50	92.22	92.94	87.98	88.59	93.21
Alexnet	89.84	90.17	87.00	86.79	87.40
Vgg16	89.79	89.69	84.58	84.40	89.52
Googlenet	90.28	88.97	86.25	85.73	92.31
Resnet50-EEMRVFL	**97.34**	**96.86**	**96.21**	**96.25**	97.42
Resnet50-RVFL	93.65	93.38	90.45	90.64	93.14
Resnet50-EMRVFL	94.08	94.13	91.52	92.04	94.98
DCAE	93.57	92.25	88.37	88.58	95.67
Resnet50-MPNCOV	96.98	96.13	93.36	93.68	97.05
CrossVit-S	94.40	93.78	91.14	91.71	97.14
PVT2-B2	95.41	94.35	91.83	92.24	98.25
DenseNet201-FCMFDA-ELM	96.43	96.50	92.54	93.02	**98.81**

for all classes. The specific formulas for these evaluation indicators are as follows:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}, \tag{10}$$

$$Precision = \frac{TP}{TP + FP}, \tag{11}$$

$$Recall = \frac{TP}{TP + FN}, \tag{12}$$

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{13}$$

$$MAP = \frac{1}{n} \sum_{1}^{N} P_i. \tag{14}$$

FN implies that it is misclassified as a negative sample. *FP* implies that it is misclassified as a positive sample. *TN* implies that it is correctly classified as a negative sample, and *TP* implies that it is correctly classified as a positive sample.

Discussion

To show the improvement in the performance of the CNN by the EEMRVFL classifier and the superiority of the method proposed here, the LifeCLEF 2015 and URPC datasets were extracted through Resnet50. After dimensionality reduction, different classifiers (softmax, EEMRVFL, EMRVFL, and RVFL) were used. Experimental results with evaluation indicators are presented in Tables 4 and 5. Each algorithm was run ten times with a random 4:1 partition of the dataset on each run. Optimal results of the algorithms are listed in the table. The Resnet50-EEMRVFL achieves 99.68% and 97.34% accuracy on the URPC and LifeCLEF 2015 datasets, respectively. This performance is better than that of the other algorithms considered for comparison. The Resnet50-EEMRVFL is better than other models in terms of average accuracy, precision, recall, and *F*1-score that are 99.68%, 99.59%, 99.59%, and 99.59% on the URPC dataset, respectively, and 97.34%, 96.86%, 96.21%, and 96.25% on the LifeCLEF 2015 dataset, respectively. In addition, the Resnet50-RVFL and Resnet50-EMRVFL classification models exhibit better classification accuracy compared to the original softmax classifier.

To better represent the improvement achieved using the Resnet50-EEMRVFL, this study compares the confusion matrix of each algorithm model in Tables 4 and 5 and calculates the corresponding evaluation index to evaluate the algorithm. Figures 3 and 4 show confusion matrices for our method on the classification results of the URPC and LifeCLEF 2015 datasets. Supplementary Figures S1 and S2 show the confusion matrix of other algorithms involved in Table 4. Rows in Figures 3 and 4 represent the predicted categories, whereas columns represent true categories. Table 6 shows the growth of network training time when the classifier of the algorithm in this paper is updated from RVFL to EMRVFL and then to EEMRVFL. In addition, the training time of state-of-the-art's deep learning algorithm involved in Tables 4 and 5 is compared with our method. The training batches of the depth learning methods involved in the comparison are all 100 iterations and the algorithm converges stably. It can be seen from Supplementary Table S2 that the Resnet50-EEMRVFL underwater image classification algorithm proposed in this paper has steadily improved the accuracy and accuracy of the algorithm at the cost of increasing part of the time cost.

To further demonstrate the advantages of Resnet50-EEMRVFL in classification accuracy, this study verifies the four latest CNNs of Densenet201, Efficienetb0, Resnet101, and Darknet53 on the URPC and LifeCLEF 2015 datasets. The results in Supplementary Table S3 show that the Resnet50-EEMRVFL achieves the best classification accuracy.

To show performances improvement of the Resnet50-EEMRVFL compared with the conventional algorithms in underwater image classification tasks, we validated the proposed algorithm model on the Fish4Knowledge dataset. We selected fish images with fewer than 300 data points in the dataset. These images were randomly rotated between -10° and 10° and added to the original dataset. This process was repeated five times. The classification accuracy of each method is listed in Supplementary Table S4, among which the Resnet50-EEMRVFL model, realizing an accuracy of 99.77%, has obvious advantages compared with the remaining existing models. As shown in Figure 5, Resnet50-EEMRVFL also demonstrates excellent classification accuracy for a small number of species on the Fish4 dataset with an extremely uneven distribution of species.

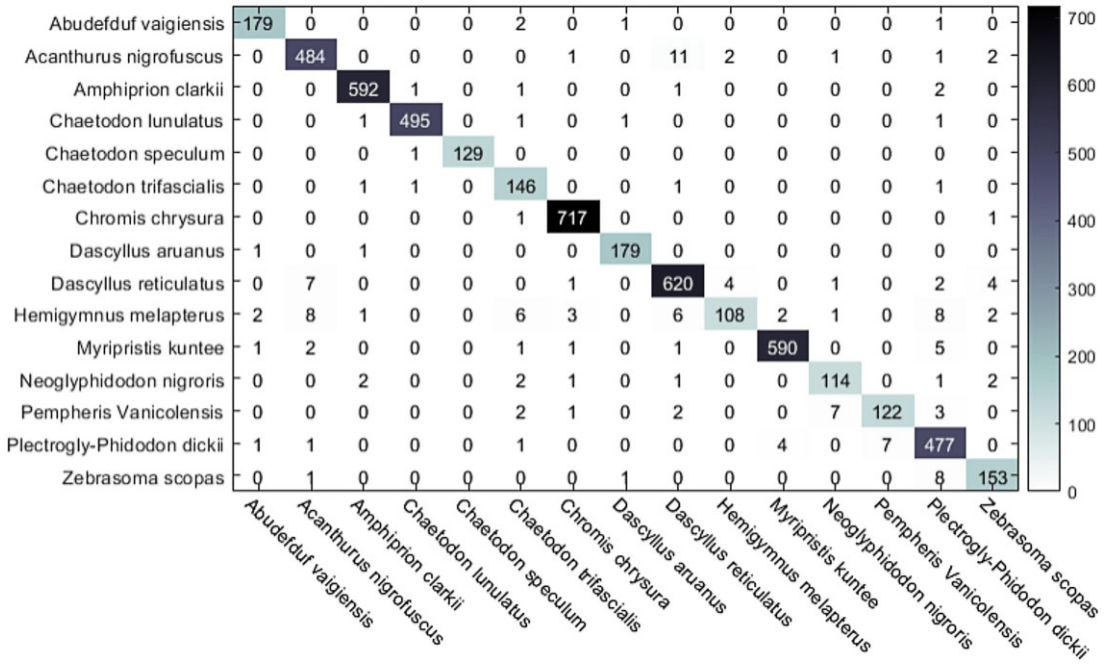

Figure 3. Confusion matrix for LifeCLEF 2015 dataset with proposed residual CNNs.

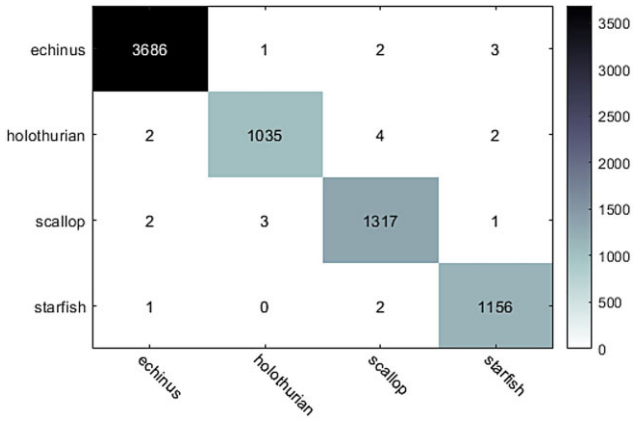

Figure 4. Confusion matrix for URPC dataset with proposed residual CNNs.

Table 6. Training time cost of each algorithm on URPC and LifeCLEF 2015 dataset.

Time cost (s)	URPC	LifeCLEF 2015
Resnet50-RVFL	12 531	9994
Resnet50-EMRVFL	15 587	11 526
Resnet50-EEMRVFL	46 675	39 124
DCAE	30 423	15 942
Resnet50-MPNCOV	31 190	22 359
CrossVit-S	26 908	19 590
PVT2-B2	31 332	21 720
DenseNet201-FCMFDA-ELM	22 117	18 842

We compare the classification accuracy of the four methods Resnet101, Resnet152, Resnet101-EEMRVFL, and Resnet152-EEMRVFL on the LifeCLEF 2015 and URPC datasets to verify the generality of improving the CNN model by replacing the softmax classifier with the EEMRVFL classifier. From results in Supplementary Table S5, it can be concluded that the CNN using the EEMRVFL classifier has better classification accuracy compared to the original pre-trained CNN, which shows that our method has excellent generalization. The classification accuracy of the Resnet101 and Resnet152 models is lower than that of the Resnet50 model on the two datasets used in this study, which may be due to the small number of underwater image datasets and the imbalanced distribution.

In this study, we selected a total of 42391 images from 33 underwater image-related categories on the ImageNet dataset to verify the performance of the Resnet50-EEMRVFL classification model. These included anemone fish, axolotl, box turtle, brain coral, chambered nautilus, coral reef, dugong, electric ray, flatworm, gar, goldfish, great white shark, grey whale, hammerhead, jellyfish, killer whale, leatherback turtle, lionfish, loggerhead, mud turtle, platypus, puffer, rock beauty, sea anemone, sea cucumber, sea slug, sea urchin, spiny lobster, starfish, stingray, sturgeon, terrapin, and tiger shark. Experimental results show that the accuracy of the proposed residual CNNs Resnet50-EEMRVFL model and Resnet50 model on ImageNet dataset are 91.47% and 89.48% respectively.

Stability analysis

In this study, experimental results were obtained by running each algorithm ten times to generate a box plot, which shows the stability of algorithms, as shown in Figure 6 and Supplementary Figure S4. The red line on the box is the median of the classification results. The width and height of the box represent the stability and accuracy of the algorithm, respectively. The smaller the box, the more concentrated the classification accuracy results, and the more stable the classification performance of the algorithm. The higher the position of the box, the higher the overall classification accuracy of the algorithm and the better the classification performance. Analyzing the accuracy distribution in Figure 6 and Supplementary Fig-

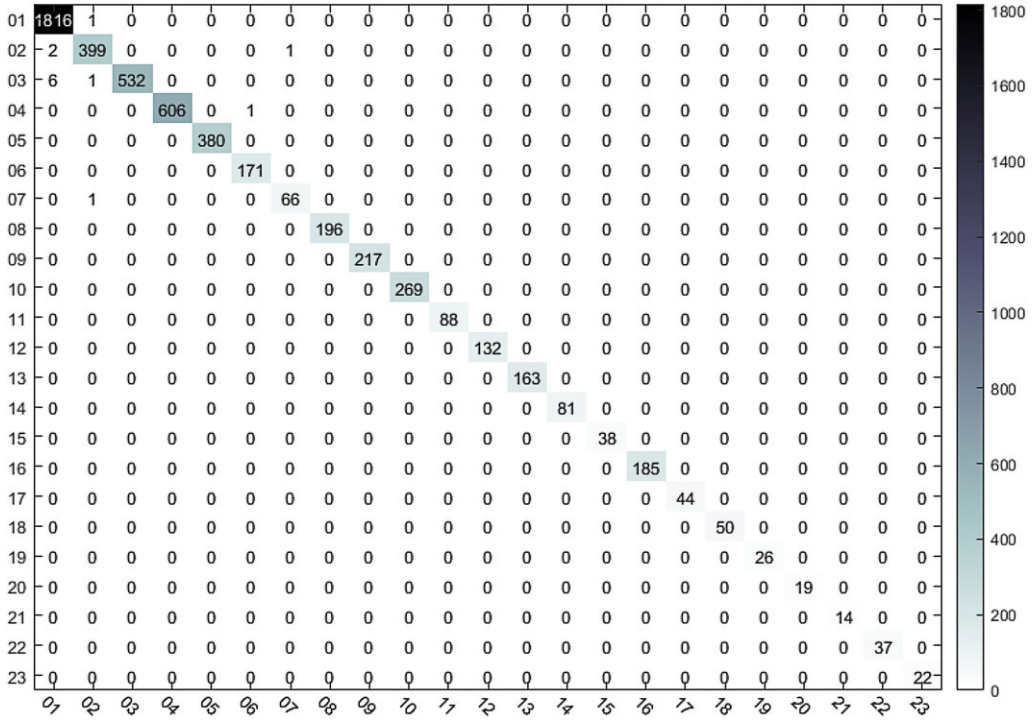

Figure 5. Confusion matrix for Fish4Knowledge dataset with proposed residual CNNs.

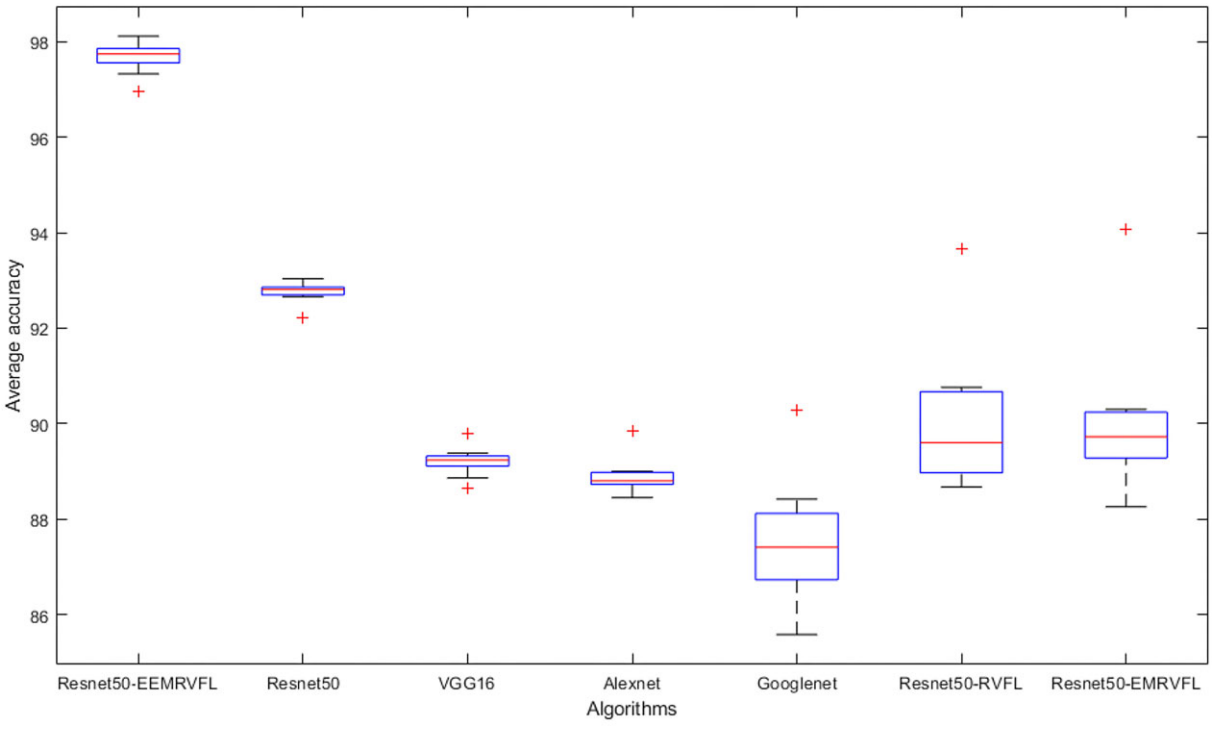

Figure 6. Box plot of accuracy on LifeCLEF 2015 dataset with each algorithm.

ure S3 shows that in the LifeCLEF 2015 and URPC datasets, the Resnet50-EEMRVFL model has the highest experimental accuracy. The algorithm in this study is superior to most algorithms in terms of stability and does not have exceedingly many outliers.

Significance analysis

To illustrate whether there is a significant difference in classification results between the Resnet50-EEMRVFL and other comparison methods, a two-sample analysis of variance was performed on the experimental average and overall results be-

tween algorithms, and the significance level was set to 0.05. The formula for calculating *F*-test is as follows:

$$S^2 = \frac{\sum \left(\overline{X_1} - \overline{X_2}\right)^2}{n-1}, \tag{15}$$

$$F = \frac{S_1^2}{S_2^2}. \tag{16}$$

The calculation formula of the *T*-test is as follows:

$$T = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{\delta_{x_1}^2 + \delta_{x_2}^2 - 2\gamma\delta_{x_1}\delta_{x_2}}{n-1}}}, \tag{17}$$

where $\overline{X_1}$ and $\overline{X_2}$ and $\delta_{x_1}^2$ and $\delta_{x_2}^2$ are the averages and variances of the two groups of algorithms, respectively. γ is the correlation coefficient of the algorithm result, and n is the size of the data.

First, an *F*-test was performed; $p > 0.05$ indicates that the overall variance of the two groups of algorithm results is equal, and the *T*-test was performed. $p > 0.05$ indicates the heteroscedastic two-sample *T*-test was performed followed. As shown in Supplementary Table S6, Resnet50-EEMRVFL and Resnet50, Alexnet, Resnet50-EMRVFL, DeepFish-SVM-aug-scale (Qin *et al.*, 2016), and DeepCNN-KNN (Deep and Dash 2019) are in line with the homogeneity of variances. The *T*-test with equal variance hypothesis is employed. Resnet50-EEMRVFL, Vgg16, Resnet50-RVFL, Googlenet, Median filtering with CNN (Jin *et al.*, 2017), and DeepCNN-KNN (Deep and Dash 2019) employ the heteroscedastic two-sample *T*-test. The results listed in Supplementary Tables S7 and S8 are both <0.05, which indicates that results of the Resnet50-EEMRVFL model are significantly different from other algorithms. This shows that the algorithm proposed in this study is significantly better than other algorithms.

Conclusion

The key points of this study are summarized as follows:

1) EEMRVFL is proposed, which randomly generates multiple hidden nodes in each incremental learning step and selects the hidden nodes that can minimize the network error to join the network.
2) The EEMRVFL is used as the classifier of underwater images for Resnet50. The proposed algorithm takes advantage of the CNN to extract image features and simultaneously obtains a better image classification effect through the EEMRVFL classifier.
3) Resnet50 was used to extract underwater image features and the EEMRVFL classifier to classify underwater targets. To verify the performance of Resnet50-EEMRVFL, the stability and significance of the algorithm were tested using box plots and variance analysis. In addition, this study analyzes the influence of various algorithm parameters on experimental results.
4) The proposed residual CNNs Resnet50-EEMRVFL algorithm is verified on the LifeCLEF 2015 and URPC datasets, achieving 97.34% and 99.68% accuracies, respectively. Its performance is evidently better than that of conventional CNN. The accuracy, recall, F1-score, and MAP of Resnet50-EEMRVFL on the LifeCLEF 2015 dataset are 96.86%, 96.21%, 96.25%, and 97.42%, respectively. To compare performances of the proposed algorithm with similar existing ones, experiments have been conducted on the Fish4Knowledge dataset, and the accuracy is found to be 99.77%, which is superior to existing methods. In addition, the stability and significance of the Resnet50-EEMRVFL algorithm are tested using box plots and through variance analysis. In future studies, further improvements to the CNN and increase in the training speed of the EEMRVFL classifier will be considered.

Acknowledgements

We appreciate the support of the National Key R&D Program of China (No. 2022YFC2803903) and the Key R&D Program of Zhejiang Province (No. 2021C03013).

Supplementary data

Supplementary material is available at the *ICESJMS* online version of the manuscript.

Conflict of interest

The authors have no conflicts of interest to declare.

CRediT authorship contribution statement

Zhiyu Zhou: Writing—review & editing, Methodology, Supervision. **Xingfan Yang:** Writing—review, Comparative experiment. **Haodong Ji:** Methodology, Data curation, Investigation, Writing—original draft, Software. **Zefei Zhu:** Supervision, Funding acquisition, Writing—review.

Data availability

The datasets were derived from sources in the public domain: http://en.cnurpc.org/and https://www.imageclef.org/lifeclef/2015/fish

References

Allken, V., Handegard, N. O., Rosen, S., Schreyeck, T., Mahiout, T., and Malde, K. 2019. Fish species identification using a convolutional neural network trained on synthetic data. ICES Journal of Marine Science, 76: 342–349.

Banan, A., Nasiri, A., and Taheri-Garavand, A. 2020. Deep learning-based appearance features extraction for automated carp species identification. Aquacultural Engineering, 89: 102053.

Banerjee, A., Das, A., Behra, S., Bhattacharjee, D. *et al.* 2022. Carp-DCAE: deep convolutional autoencoder for carp fish classification. Computers and Electronics in Agriculture, 196: 106810.

Beyan, C., and Browman, H. I 2020. Setting the stage for the machine intelligence era in marine science. ICES Journal of Marine Science, 77: 1267–1273.

Chandran, C. S., Kamal, S., Mujeeb, A., and Supriya, M. H 2021. Generative adversarial learning for improved data efficiency in underwater target classification. Engineering Science and Technology, an International Journal, 30:101043, https://doi.org/10.1016/j.jestch.2021.07.006.

Chen, C-Fu (R), Fan, Q., and Panda, R. 2021. CrossViT: cross-attention multi-scale vision transformer for image classification. Computer Vision and Pattern Recognition, arXiv:2103.14899.

Chen, Y., Zhu, J., Wan, L., Fang, X., Tong, F., and Xu, X. 2022. Routing failure prediction and repairing for AUV-assisted underwater acoustic sensor networks in uncertain ocean environments. Applied Acoustics, 186: 108479.

Deep, B. V., and Dash, R. 2019. Underwater fish species recognition using deep learning techniques. 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 665–669pp.

Du, A., Gu, Z., Yu, Z., Zheng, H., and Zheng, B. 2020. Plankton image classification using deep convolutional neural networks with second-order features. 2020 Global Oceans, Singapore, U.S. Gulf Coast.

Duan, S., Lin, Y., Zhang, C., Li, Y., Zhu, D., Wu, J., and Lei, W. 2021. Machine-learned, waterproof MXene fiber-based glove platform for underwater interactivities. Nano Energy, 91:106650.

Durden, J. M., Hosking, B., Bett, B. J., Cline, D., and Ruhl, H. A 2021. Automated classification of fauna in seabed photographs: the impact of training and validation dataset size, with considerations for the class imbalance. Progress in Oceanography, 196: 102612.

Feng, G., Huang, G., Lin, Q., and Gay, R. 2009. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Transactions on Neural Networks, 20: 1352–1357.

Geng, M., Wang, Y., Tian, Y., and Huang, T. 2016. CNUSVM: hybrid CNN-uneven SVM model for imbalanced visual learning. 2016 IEEE Second International Conference on Multimedia Big Data (BigMM), 186–193pp.

He, K., and Sun, J. 2015. Convolutional neural networks at constrained time cost. 2015 IEEE Conference on Computer Vision and Pattern Recognition, 5353–5360pp.

He, K., Zhang, X., Ren, S., and Sun, J. 2016. Deep residual learning for image recognition. Computer Vision and Pattern Recognition, 770–778pp.

Huang, G., Chen, L., and Siew, C. K 2006. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks, 17: 879–892.

Jin, L., and Liang, H. 2017. Deep learning for underwater image recognition in small sample size situations. Conference Oceans. 1–4pp.

Krizhevsky, A., Sutskever, I., and Hinton, G. E 2017. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60: 84–90.

Li, Y., Liu, S., Zhu, P., Yu, J., and Li, S. 2017. Extraction of visual texture features of seabed sediments using an SVDD approach. Ocean Engineering, 142: 501–506.

Liu, X., Jia, Z., Hou, X., Fu, M., Ma, L., and Sun, Q. 2019. Real-time marine animal images classification by embedded system based on mobilenet and transfer learning. OCEANS 2019, Marseille, 1–5pp.

Lu, Y., Tung, C., and Kuo, Y. 2020. Identifying the species of harvested tuna and billfish using deep convolutional neural networks. ICES Journal of Marine Science, 77: 1318–1329.

Mahmood, A., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. 2020. ResFeats: residual network based features for underwater image classification. Image and Vision Computing, 93: 103811.

Malde, K., Handegard, N. O., Eikvil, L., and Salberg, A. B 2020. Machine intelligence and the data-driven future of marine science. ICES Journal of Marine Science, 77: 1274–1285.

Marre, G., Deter, J., Holon, F., Boissery, P., and Luque, S. 2020. Fine-scale automatic mapping of living Posidonia oceanica seagrass beds with underwater photogrammetry. Marine Ecology Progress Series, 643: 63–74.

Meng, X., Zhang, S., and Zang, S. 2019. Lake wetland classification based on an SVM-CNN composite classifier and high-resolution images using wudalianchi as an example. Journal of Coastal Research, 93: 153–162.

Muhammad, I., Zheng, J., Muhammad, O., Zafar, M., Muhammad, H. A., and Syed, R. U. H 2021. Brain inspired lifelong learning model based on neural based learning classifier system for underwater data classification. Expert Systems with Applications, 186: 115798.

Pao, Y. H., Park, G. H., and Sobajic, D. J 1994. Learning and generalization characteristics of the random vector functional-link net. Neurocomputing, 6: 163–180.

Qiao, W., Khishe, M., and Ravakhah, S. 2021. Underwater targets classification using local wavelet acoustic pattern and multi-layer perceptron neural network optimized by modified Whale Optimization Algorithm. Ocean Engineering, 219: 108415.

Qin, H., Li, X., Liang, J., Peng, Y., and Zhang, C. 2016. DeepFish: accurate underwater live fish recognition with a deep architecture. Neurocomputing, 187: 49–58.

Qiu, X., Suganthan, P. N., and Amaratunga, G. A. J 2018. Ensemble incremental learning random vector functional link network for short-term electric load forecasting. Knowledge-Based Systems, 145: 182–196.

Rathi, D., Jain, S., and Indu, S. 2017. Underwater fish species classification using convolutional neural network and deep learning. 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), 1–6pp.

Salman, A., Siddiqui, S. A., Shafait, F., Mian, A., Shortis, M. R., Khurshid, K., Ulges, A. *et al.* 2020. Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES Journal of Marine Science, 77: 1295–1307.

Siddiqui, S. A., Salman, A., Malik, M. I., Shafait, F., Mian, A., Shortis, M. R., and Harvey, E. S., 2018. Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data. ICES Journal of Marine Science, 75: 374–389.

Simonyan, K., and Zisserman, A. 2015. Very deep convolution networks for large-scale image recognition. CoRR, arXiv: abs/1409.1556.

Srivastava, R. K., Greff, K., and Schmidhuber, J. 2015. Highway networks. arXiv:1505.00387.

Sung, M., Yu, S., and Girdhar, Y. 2017. Vision based real-time fish detection using convolutional neural network. OCEANS, 2017: 1–6.

Szegedy, C., Vanhoucke, V., lofffe, V., Shlens, J., and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. Computer vision and pattern recognition, 2818–2826pp.

Wang, W., Xie, E., Xiang, L. *et al.* 2022. PVT v2: improved baselines with Pyramid vision transformer. Computer Vision and Pattern Recognition, arXiv:2102.12122.

Xie, K., Pan, W., and Xu, S. 2018. An underwater image enhancement algorithm for environment recognition and robot navigation. Robotics, 7: 14.

Xu, Y., Zhang, Y., Wang, H., and Liu, X. 2017. Underwater image classification using deep convolutional neural networks and data augmentation. 2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 1–5pp.

Yang, J., Cai, M., Yang, X. *et al.* 2022. Underwater image classification algorithm based on convolutional neural network and optimized extreme learning machine. Journal of Marine Science and Engineering, 10: 1841.

Yeh, C., Lin, M., Chang, P., and Kang, L. 2020. Enhanced visual attention-guided deep neural networks for image classification. IEEE Access, 8: 163447–163457.

Yuan, H., Zhang, S., Chen, G., and Yang, Y. 2020. Underwater image fish recognition technology based on transfer learning and image enhancement. Journal of Coastal Research, 105: 124–128.

Zhang, J., Zhu, L., Xu, L., and Xie, Q. 2020. Research on the correlation between image enhancement and underwater object detection. 2020 Chinese Automation Congress (CAC), 5928–5933pp.

Zhou, P., Li, W., Wang, H., Li, M., and Chai, T. 2020. Robust online sequential rvflns for data modeling of dynamic time-varying systems with application of an ironmaking blast furnace. IEEE Transactions on Cybernetics, 50: 4783–4795.

Zhuang, S., Zhang, X., Tu, D., Ji, Y., and Yao, Q. 2021. A dense stereo matching method based on optimized direction-information images for the real underwater measurement environment. Measurement, 186: 110142.

Handling Editor: Cigdem Beyan

Received: 2 January 2023; **Revised:** 14 February 2023; **Accepted:** 1 March 2023