



Article

Live Fish Species Classification in Underwater Images by Using Convolutional Neural Networks Based on Incremental Learning with Knowledge Distillation Loss

Abdelouahid Ben Tamou^{1,2} , Abdesslam Benzinou^{1,*} and Kamal Nasreddine¹ ¹ ENIB, UMR CNRS 6285 LabSTICC, 29238 Brest, France² Univ Brest, UMR CNRS 6285 LabSTICC, 29238 Brest, France

* Correspondence: benzinou@enib.fr

Abstract: Nowadays, underwater video systems are largely used by marine ecologists to study the biodiversity in underwater environments. These systems are non-destructive, do not perturb the environment and generate a large amount of visual data usable at any time. However, automatic video analysis requires efficient techniques of image processing due to the poor quality of underwater images and the challenging underwater environment. In this paper, we address live reef fish species classification in an unconstrained underwater environment. We propose using a deep Convolutional Neural Network (CNN) and training this network by using a new strategy based on incremental learning. This training strategy consists of training the CNN progressively by focusing at first on learning the difficult species well and then gradually learning the new species incrementally using knowledge distillation loss while keeping the high performances of the old species already learned. The proposed approach yields an accuracy of 81.83% on the LifeClef 2015 Fish benchmark dataset.



Citation: Ben Tamou, A.; Benzinou, A.; Nasreddine, K. Live Fish Species Classification in Underwater Images by Using Convolutional Neural Networks Based on Incremental Learning with Knowledge Distillation Loss. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 753–767. <https://doi.org/10.3390/make4030036>

Academic Editor: Weiping Ding

Received: 8 July 2022

Accepted: 18 August 2022

Published: 22 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: underwater image; fish recognition; deep learning; convolutional neural network; incremental learning; knowledge distillation loss

1. Introduction

The underwater environment, particularly coral reefs, contains diverse ecosystems rich in biodiversity. These reefs are composed of assemblages of corals, algae and sponges. This complex structure offers an ideal habitat for many species, especially for protection and feeding [1]. The underwater environment is ecologically and economically important, but is threatened by pollution [2], over-fishing [3], and climate change [4]. These factors are destroying the ecosystem and accelerating the loss of coral and fish species living there [5]. It is now necessary to monitor the evolution of these ecosystems in order to identify and even anticipate any threatening degradation of the ecosystem [6]. This monitoring is achieved by observing and then estimating the diversity and abundance of fish species to understand the structure and dynamics of the coral reef community [7]. Traditional techniques used to observe ecosystems and monitor biodiversity, such as techniques (fishing [8], anesthesia [9]) and underwater visual census (UVC) [10], are destructive and/or do not ensure continuous monitoring of underwater biodiversity. It is important to adopt more advanced techniques that are non-destructive and provide continuity in ecosystem monitoring.

In recent years, underwater video techniques have been increasingly used to observe macrofauna and habitat in marine ecosystems. Advances in video camera technology, battery life and information storage make these techniques accessible to the majority of users. Underwater video has some notable advantages. Inexpensive in terms of cost and time, it allows for a large number of observations that can be reused at any time. It also makes it possible to monitor the aquatic communities of the ecosystem without disturbing its functioning. This is a major advantage over UVCs, which require the presence of a diver in the environment. In addition, this technique is preferable for monitoring large areas

such as marine protected areas (MPAs), marine parks or World Heritage sites. It is also easy to implement, and the recorded videos can be used by non-specialists.

However, current underwater video observation techniques require human experts to analyze the rapidly increasing amount of data collected. Automatic processing of underwater videos is the key to efficiently and objectively analyzing these large amounts of data. In fact, this technique is not yet provided due to the difficulties presented by the underwater environment, which poses great challenges for computer vision (Figure 1). The luminosity changes frequently due to ocean current, visibility is limited, and the complex coral background sometimes changes rapidly due to moving aquatic plants. Object recognition in underwater video images is an open challenge in pattern recognition, especially when dealing with fish species recognition. Fish move freely in all directions, and they can also hide behind rocks and algae. In addition, the problems of fish overlap, and similarities in shape and pattern between fish of different species pose significant challenges in our application.

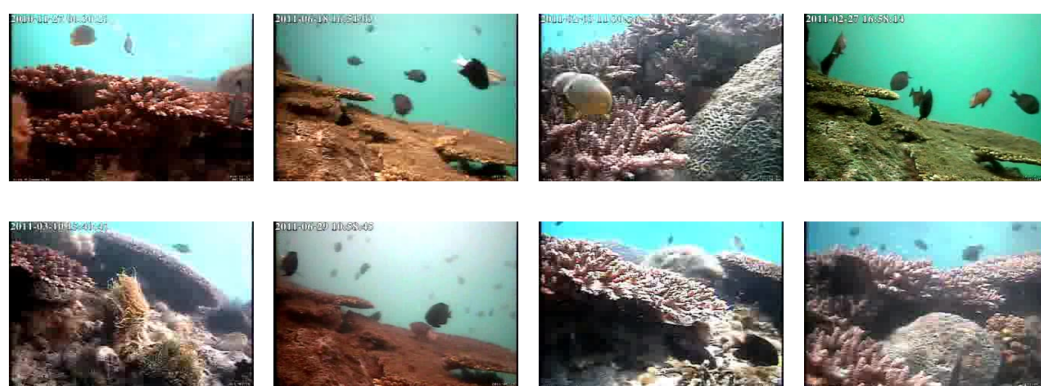


Figure 1. Examples of underwater images from different videos of the LifeClef 2015 Fish dataset. These examples illustrate the high variation in unconstrained underwater environments, such as complex, crowded and dynamic backgrounds and luminosity variation.

We can divide automatic fish recognition into two steps: (1) fish detection, which aims to localize every single fish in the underwater image, (2) and fish classification, which aims to identify the species of each detected fish. In our previous work [11], we focused on fish detection in unconstrained underwater videos. In this paper, we address the fish species classification in underwater images.

Last decade, several works developed methods for live fish species recognition in open sea. The early works mainly used hand-crafted techniques such as forward sequential feature selection (FSFS) [12], discriminant analysis approach [13], histogram of oriented gradients (HOG) [14], and SURF [15].

Recently, convolutional neural networks (CNNs) have shown high performance for underwater vision enhancement [16,17], and for fish detection [11,18–20]. For fish species classification, the first works based on CNNs applied well-known pretrained networks such as AlexNet [21–23], VGGNet [24], GoogleNet [25,26], and ResNet [27–29]. Other works attempted to develop new architectures with few convolutional layers [30–32]. Others proposed a hybrid of deep architecture with hand-crafted techniques [33,34]. Some other works attempted to address fish detection and species classification within the same framework [18,19,35].

All of these CNN-based works used classical multi-class CNN, which is trained at once on all classes in the dataset. With this structure, the CNN treats all classes the same. However, some classes are naturally more likely to be misclassified than others, especially for classes that have fewer samples or are difficult to classify. In a live fish recognition task, the species in the training set can be grouped into categories. We can group species according to their degree of difficulty. Difficult species require special treatment when learning the CNN model. On the other hand, incremental learning allows a model to

integrate new examples without having to perform a complete re-training and without destroying the knowledge acquired from old data. We propose to build a CNN classifier starting with the difficult species. Initially, the model focuses on learning the difficult species well and then gradually learns the other species in an incremental way. We aim to keep the model stable when introducing new species while maintaining high performances on the old species already learned.

The main contributions of this paper are summarized as follows:

- We propose a novel approach of using knowledge distillation for the training of the CNN architecture for live coral reef fish species classification task in unconstrained underwater images.
- We propose to train the pre-trained ResNet50 [36] progressively by focusing at the beginning on hard fish species and then integrating more easy species.
- Extensive experiments and comparisons of results with other methods are presented. The proposed approach outperforms state-of-the-art fish identification approaches on the LifeClef 2015 Fish (www.imageclef.org/lifeclef/2015/fish accessed on 7 July 2022) benchmark dataset.

The rest of this paper is organized as follows. Section 2 outlines related work. Section 3 presents the proposed approach for underwater live fish species classification. Section 4 describes the benchmark dataset used in this work, provides the experimental results and performs a comparative study. Finally, the conclusion and perspectives are discussed in Section 5.

2. Related Works

In this section, we present a state-of-the-art computer vision approach for fish species classification (Section 2.1). Then, we briefly expose related works for incremental learning approaches (Section 2.2).

2.1. Fish Species Classification

Early work used hand-crafted features to recognize fish species in the open sea. Spampinato et al. [13] combined texture and shape features. An affine transformation is also applied to the acquired images to represent the fish in 3D. Cabrera-Gómez et al. [14] computed different local descriptors such as histogram of oriented gradients (HOG), local binary patterns (LBP), uniform local binary patterns (ULBP) and local gradient patterns (LGP). To improve classification results, they adopted a score-level fusion approach where the first layer is composed of a set of classifiers designed to each chosen descriptor, while the second layer classifier takes as input the scores of the first layer. SVM-based techniques can be considered flat classifiers because they classify all classes at the same time using the same features for all classes. Sometimes it may be useful to choose specific features for different classes; hierarchical classification tree techniques take this into account. The idea is to gradually separate the set of images into sub-classes, with each node in the tree having its own set of features. The main disadvantage of this structure is the accumulation of errors because if an error is made at a node, it will necessarily lead to new errors in the child nodes. Huang et al. [12] extracted 66 types of features: color, shape, and texture of different parts of the fish. They then proposed a hierarchical classification method called “Balance-Guaranteed Optimized Tree (BGOT)” that is supposed to minimize the error accumulation problem. Szűcs et al. [15] used speeded up robust features (SURFs) to classify fish species [37]. Dhar and Guha [38] extracted the robust gist feature and gray level co-occurrence matrix feature from a fish image. Then, they combined these features to feed an XgBoost classifier.

With the advent of deep learning, in particular convolutional neural networks (CNNs), many studies have focused on investigating the contribution of CNNs to the resolution of different tasks in computer vision. Villon et al. [25] presented two methods for recognizing fish in coral in HD underwater videos. The first method is based on a traditional two-step approach: extraction of HOG features and use of an SVM classifier. The second method

is based on deep learning using the GoogleNet architecture [39]. They compared the two methods and found that deep learning is more efficient than HOG+SVM. Salman et al. [31] proposed a CNN of three convolution layers to extract features and feed standard classifiers such as SVM and K nearest neighbors (KNN). Qin et al. [30] proposed a CNN with three convolutional layers trained from scratch on the Fish Recognition Ground-Truth dataset. They also proposed in [33] a hybrid deep architecture with traditional methods to extract features from fish images. In this architecture, principal component analysis (PCA) is used in two convolutional layers, followed by binary hashing in the non-linear layer and block-wise histogram in the pooling layer. Then, spatial pyramid pooling (SPP) [40] is used. Finally, classification is performed with a linear SVM. Compared to their first work [30], the proposed hybrid deep architecture improved the accuracy by only 0.07%. Sun et al. [34] applied two deep architectures, PCANet [41] and NIN [42], to extract features from underwater images. A linear SVM classifier is used for classification. Jäger et al. [21] used features extracted from the activations of the seventh hidden layer of the pre-trained AlexNet model [43], and fed a multi-class SVM classifier. Sun et al. [22] proposed to extract the features of fish from a pre-trained deep CNN using transfer learning [44]. They re-trained AlexNet with artificial data augmentation using an SVM classifier. Mathur et al. [28] fine-tuned a ResNet50 model by only re-training the last fully connected layers without any data augmentation. They used Adamax as an optimizer. Zhang et al. [29] proposed AdvFish, which addresses the noisy background problem. They fine-tuned the ResNet50 model by adding a new term in the loss function. This term encourages the network to automatically differentiate the fish regions from the noisy background and pay more attention to the fish regions. Pang et al. [45] used the teacher-student model to reduce the impact of interference on fish species classification. They distilled interference information by reducing the discrepancy of two distance matrices generated separately from a processed fish image and a raw fish image. KL-divergence is used to further reduce noise in the raw data distribution.

Other works tried to modify the original pre-trained models. Ju et al. [23] proposed an improved AlexNet model. This model has less structural complexity, it consists of four convolutional layers instead of five and an added item-based soft attention layer. Iqbal et al. [24] used a reduced version of AlexNet consisting of four convolutional layers and two fully-connected layers. Cheng et al. [46] proposed a deep residual shrinkage network. This network is an improved attention mechanism algorithm based on a deep residual network, which embeds a soft threshold as a shrinkage layer into a residual module to automatically set thresholds for each feature channel.

Other work attempted to address fish detection and species classification in the same framework. Li et al. [18] applied fast R-CNN convolutional networks [47] to detect and recognize fish species. Knausgard et al. [48] used the You Only Look Once (YOLO) object detection technique to detect fish in underwater images [49]. Then, they adopted a CNN with the squeeze-and-excitation (SE) architecture [50] to classify each fish in the image with a transfer learning framework. Jalal et al. [19] proposed a hybrid approach combining spatial and temporal information based on YOLOv3 to detect and classify fish images. An interesting recent survey cites all works on fish species recognition [51,52].

The training of our approach is different from classical multi-class CNN training that trains CNN at once in all classes. In our application, we group species according to their degree of difficulty. First, we start training our CNN with the difficult species, and then it gradually learns the other species in an incremental way.

2.2. Incremental Learning

Incremental learning [53,54] refers to learning from streaming data, which arrives over time. It allows a model to receive and integrate new examples without having to perform a complete re-training and without destroying the knowledge acquired from old data. An incremental learning algorithm is defined in [55] as one that meets the following criteria:

- It should be able to learn additional knowledge from new data;

- It should not require access to the original data (i.e., the data that were used to learn the current classifier);
- It should preserve previously acquired knowledge;
- It should be able to learn new classes that may be introduced with new data.

These four points apply to any general incremental learning problem.

In our application, we want to use the principle of incremental learning in a classical transfer learning context. In classical transfer learning, a pre-trained model is re-trained once on a new dataset with a predefined number of classes. Our approach based on incremental learning trains a model progressively while adding new classes at each transfer of knowledge. The common point between this learning and classical incremental learning is that the model learns new data without destroying the knowledge acquired from the old data. On the other hand, the difference between them is that our approach requires re-training the system on both old and new data.

We can distinguish mainly three types of incremental learning algorithms:

- Architectural strategy [56]: this algorithm modifies the architecture of the model in order to mitigate forgetting, e.g., adding layers, fixing weights. . .
- Regularization strategy [57,58]: loss terms are added to the loss function that promotes the selection of important weights to keep the knowledge gained. This type also includes basic regularization techniques such as stalling and early stopping.
- Repetition strategy [59]: old data are periodically replayed in the model to strengthen the connections associated with the learned knowledge. A simple approach is to store some of the previous training data and interleave it with new data for future training.

In our approach, we use the regularization strategy by modifying the loss function of the system.

For validation purposes, we carried out experiments on the LifeClef 2015 Fish (LCF-15) dataset. This underwater benchmark dataset is captured by underwater cameras in the open sea.

3. Proposed Approach

Here we propose a CNN approach for efficient fish species identification based on transfer learning and incremental learning. This approach trains the CNN progressively while adding a new species. For learning the new species, we use the Learning Without Forgetting approach (Section 3.2), which modifies the loss function by adding a term of loss called knowledge distillation loss (Section 3.3).

3.1. Architecture of the Approach

The architecture of our proposed approach is illustrated in Figure 2. We consider first a CNN with a set of shared parameters θ_s (the convolutional layers). In the output layer, we consider specific parameters tuned using old species θ_o (the weights of neurons in the output layer corresponding to the old species). Finally, we have specific parameters assigned to the new classes, randomly initialized θ_n (the weights of neurons in the output layer corresponding to the new species). Our goal is to learn the specific parameters θ_n and update the parameters θ_s and θ_o in order to ensure that the entire model performs well on both old and new species.

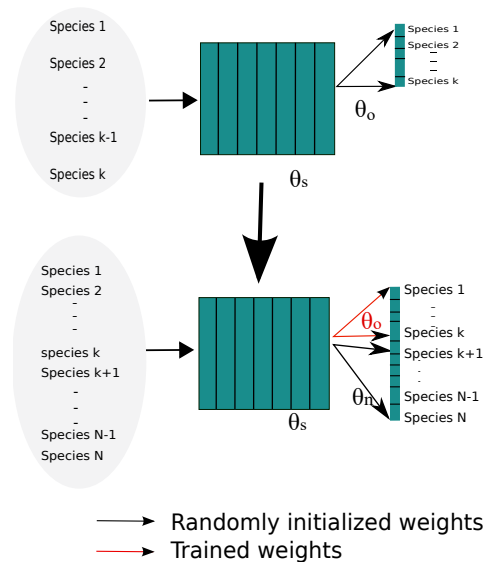


Figure 2. General view of the proposed approach based on incremental learning. The system initializes the weights corresponding to the new species randomly and keeps the trained weights.

3.2. Learning Phase

Let X_o be the set of samples of k difficult species (also called old species), X_n the set of samples of $N-k$ species, where N is the total number of species in dataset and $X = X_o \cup X_n$ is the total training set. We train our model in three steps:

- Step 1: train parameters θ_s and θ_o : First, using classical transfer learning, we train a pre-trained network, here ResNet50, on X_o .
- Step 2: calculate probabilities: At the end of the first step, each image $x_i \in X$ is passed through the trained CNN (of parameters θ_s and θ_o) to generate a vector of probabilities of belonging to the k old species $p_o(i)$. The set $P_o = f(\theta_s, \theta_o, X)$ of probabilities serves as labels corresponding to the training image set X ; $f(\cdot)$ is the output of the CNN using the parameters θ_s and θ_o . The objective is to train the network without moving these predictions much.
- Step 3: train all parameters: In order to incorporate the new species, we add nodes for each new species to the classification layer with randomly initialized weights (parameters θ_n). When training the new model, we jointly train all model parameters θ_s , θ_o and θ_n until convergence. This procedure, called joint-optimize training, encourages the computed output probabilities $\hat{P}_o$ to approximate the recorded probabilities P_o . To achieve this, we modify the network loss function by adding a knowledge distillation term.

3.3. Knowledge Distillation

Knowledge distillation is an approach originally proposed by Hinton et al. [60] with the goal of reducing the size of a network. It uses two networks: a powerful but complex and expensive one called the master, and a smaller one called the student. First, the master network is trained, and then the student network is trained to imitate it. Practically, it predicts the outputs of the master by imitating the probabilities assigned to each class. In the end, we will have two networks of different sizes that produce the same outputs. This approach results in a lighter student model and improves performance.

In our approach, we propose to use knowledge distillation to train the network when adding the new species without forgetting the old knowledge. The knowledge distillation allows the network to reconcile its outputs after the integration of new classes to the outputs of the network before the integration. This can be modeled by a modified cross-entropy loss that increases the weight for smaller probabilities:

$$L_{distillation}(p_o, \hat{p}_o) = - \sum_{i=1}^l p_o'^{(i)} \log(\hat{p}_o'^{(i)}) \quad (1)$$

where l is the number of labels, p_o' and $\hat{p}_o'$ are modified versions of the recorded p_o and calculated $\hat{p}_o$ probabilities:

$$p_o'^{(i)} = \frac{(p_o^{(i)})^{\frac{1}{T}}}{\sum_j (p_o^{(j)})^{\frac{1}{T}}} ; \quad \hat{p}_o'^{(i)} = \frac{(\hat{p}_o^{(i)})^{\frac{1}{T}}}{\sum_j (\hat{p}_o^{(j)})^{\frac{1}{T}}} \quad (2)$$

where $p_o'^{(i)}$ and $\hat{p}_o'^{(i)}$ are softmax outputs and T is a parameter called temperature, which for a standard softmax function is normally set to 1. As T grows, the probability distribution generated by the softmax function becomes softer, providing more information about which classes the master finds more similar to the predicted class.

3.4. Total Loss Function

The total loss function (L_{total}) of the network is the sum of the knowledge distillation ($L_{distillation}$) and the loss function used by the network to learn the classes (L_{loss}).

$$L_{total} = \lambda_o L_{distillation}(P_o, \hat{P}_o) + L_{loss}(Y, \hat{Y}) \quad (3)$$

where λ_o is a loss balance weight between old and new classes. By increasing its value, we favor training the old images over the new images. Y and $\hat{Y}$ are the vectors of truth and calculated labels, respectively, corresponding to the training image set $X = X_o \cup X_n$.

This approach has advantages over classic transfer learning. Indeed, the strategy of feature extraction without fine-tuning generally performs worse on a new dataset because the shared parameters θ_s are related to the old classes, and they have not learned to extract discriminative features related to new classes. On the other hand, fine-tuning degrades performance in the old classes because the shared parameters have been relearned. However, proposed incremental learning allows learning a network without forgetting the old knowledge.

4. Experiments

We used the LCF-15 benchmark underwater dataset to evaluate the effectiveness of the proposed approach for fish species classification. The dataset contains images of fish of different colors, textures, positions, scales, and orientations. It was issued from the European project Fish4Knowledge (F4k) (www.fish4knowledge.eu accessed on 7 July 2022) [61]. During this project of five years, a large dataset of over 700,000 unconstrained underwater videos with more than 3000 fish species was collected in Taiwan, the largest fish biodiversity environment in the world.

4.1. LifeClef 2015 Fish (LCF-15) Benchmark Dataset

The LCF-15 is an underwater live fish dataset. The training set consists of 20 annotated videos and more than 22,000 annotated sample images. In this dataset, we have 15 different fish species. Figure 3 shows examples of the 15 fish species, and Table 1 gives the distribution of the fish species in the dataset. Each video is manually labeled and agreed on by two specialist experts. The dataset is imbalanced in the number of instances of different species; for example, the number of the species 'Dascyllus reticulatus' is about 40 times more than the species 'Chaetodon speculum'. The fish images have various sizes ranging from about 20×20 to about 200×200 pixels.

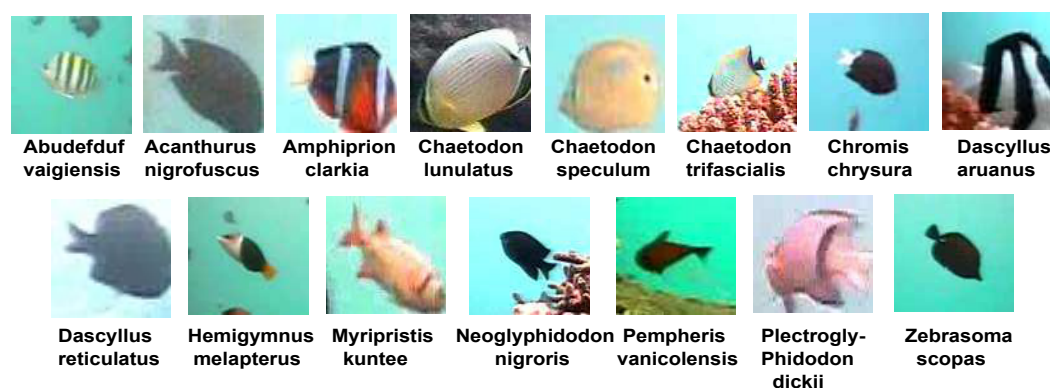


Figure 3. Sample images of 15 fish species in LCF-2015 dataset.

Table 1. The fish species distribution in LCF-15 dataset.

ID	Species	Training Set Size	Test Set Size
AV	Abudefduf vaigiensis	436	94
AN	Acanthurus nigrofusus	2805	129
AC	Amphiprion clarkia	3346	553
CL	Chaetodon lunulatus	3711	1876
CS	Chaetodon speculum	162	0
CT	Chaetodon trifascialis	681	1319
CC	Chromis chrysur	3858	24
DA	Dascyllus aruanus	1777	2013
DR	Dascyllus reticulatus	6333	4898
HM	Hemigymnus melapterus	356	0
MK	Myripristis kuntee	3246	118
NN	Neoglyphidodon nigroris	114	1643
PV	Pempheris Vanicolensis	1048	0
PD	Plectrogly-Phidodon dickii	2944	676
ZS	Zebrasoma scopas	343	187
Total		31,260	13,530

The test set has 73 annotated videos. We note that for three fish species, there are no occurrences in the test set (Table 1). This is to evaluate the method's capability of rejecting false positives.

Compared with other underwater live fish datasets, the LCF-15 dataset provides challenging underwater images and videos marked by more noisy and blurry environments, complex and dynamic backgrounds and poor lighting conditions [31]. We choose this dataset because it contains two categories of species: hard and easy species.

Finally, we note that fish images can be extracted from videos using the available ground truth of available fish bounding boxes.

4.2. Learning Strategy for Live Fish Species Classification

The incremental learning of the model is performed in two steps. First, we train the model on images of difficult fish species. Then, we integrate new images of easy fish species and train the model on all images: old and new ones according to the approach described in Section 3.

- Construction of two groups: in order to separate the species into two subsets, difficult and easy, we train the pre-trained network ResNet50 on all species of the LCF-15 dataset with transfer learning. Figure 4 illustrates the confusion matrix. From this confusion matrix, we can group the species into two main groups: group of species with low precision, difficult species, (AN, AV, CC, CT, MK, NN, PD, ZS) and group of species with high precision, easy species, (AC, CL, CS, DA, DR, HM, PV).

- Step 1 (difficult species): We first train the model on the first group using a pre-trained ResNet50 model. We want the model to focus on this subset. For this reason, we apply a data augmentation technique. To perform data augmentation, we proceed as follows. We flip each fish sample horizontally to simulate a new sample where fish are swimming in the opposite direction; then, we scale each fish image to different scales (tinier and larger). We also crop the images by removing one quarter from each side to eliminate parts of the background. Finally, we rotate fish images with angles -20° , -10° , 10° and 20° for invariant rotation fish recognition issues. At the end of this training, the model generates the shared parameters θ_s and the specific parameters for the first group θ_o .
- Step 2 (all species): Then, we add the species of the second group. In order to integrate these new species, we add a number of neurons equal to the number of species in this group into the classification layer. We randomly initialize the values of the weights of these new neurons (parameters θ_n) and keep the weights corresponding to the old species (θ_s and θ_o). We apply in this second training the new loss function to learn the new species while keeping the knowledge learned in the old training.

AC	76.13	0.72	0.90	0.18	3.80	0.00	0.18	7.05	3.98	5.97	0.00	0.00	0.90	0.00	0.18
AN	0.00	24.03	0.00	0.00	0.00	0.00	0.00	0.00	68.22	2.33	0.00	5.43	0.00	0.00	0.00
AV	3.19	1.06	46.81	0.00	10.64	0.00	2.13	7.45	23.40	4.26	1.06	0.00	0.00	0.00	0.00
CC	0.00	0.00	0.00	16.67	0.00	0.00	0.00	0.00	83.33	0.00	0.00	0.00	0.00	0.00	0.00
CL	0.37	0.00	0.00	0.16	97.23	0.00	0.00	0.53	1.17	0.11	0.37	0.00	0.05	0.00	0.00
CS	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CT	4.09	0.08	2.81	0.00	4.32	0.38	58.83	1.67	24.87	1.21	1.36	0.15	0.00	0.23	0.00
DA	2.09	0.00	0.35	0.00	0.15	0.00	0.35	92.90	3.68	0.35	0.00	0.00	0.00	0.15	0.00
DR	0.12	0.88	0.16	0.33	0.14	0.00	0.02	0.76	93.75	0.73	0.00	3.00	0.00	0.00	0.10
HM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
MK	42.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.63	0.85	47.46	0.00	1.69	0.00	0.00
NN	0.67	6.15	0.06	0.43	0.00	0.00	0.00	0.24	64.82	1.46	0.00	25.20	0.49	0.00	0.49
PD	7.69	2.07	0.00	0.30	3.40	0.00	0.30	0.00	31.21	1.18	0.00	0.30	53.40	0.00	0.15
PV	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
ZS	0.00	3.21	0.00	0.00	0.00	0.00	0.00	0.00	90.37	0.00	0.00	0.53	0.00	0.00	5.88
	AC	AN	AV	CC	CL	CS	CT	DA	DR	HM	MK	NN	PD	PV	ZS

Figure 4. Confusion matrix of classic transfer learning of ResNet50 on the LCF-15 dataset.

4.3. Results

In this section, we present the results of our proposed approach. First, we evaluate the model trained on difficult species (Section 4.3.1). Then, we present the results of the model trained on all species after adding the new ones. For the last model, we analyze the effect of different parameters on its performance (Section 4.3.2). Finally, we compare our approach with state-of-the-art approaches (Section 4.3.3).

4.3.1. Model Trained on Difficult Species

Figure 5 shows the confusion matrix of the model trained on the first group that contains species with low precision. The model identifies the species CT and MK well, followed by the species AV, CC, NN and PD. The species AN and ZS remain difficult to identify. We obtain an accuracy of 85.26%. If we compare the precision of these species with those of Figure 4, we notice that in this model, the precisions are higher. The aim of our new approach is to maintain these high precisions when we add the other species.

AN	43.41	0.00	52.71	3.88	0.00	0.00	0.00	0.00
CC	0.00	87.50	12.50	0.00	0.00	0.00	0.00	0.00
NN	3.59	2.80	85.58	5.72	0.00	1.03	0.00	1.28
ZS	9.63	0.00	21.39	68.98	0.00	0.00	0.00	0.00
AV	0.00	0.00	0.00	0.00	87.50	12.50	0.00	0.00
CT	0.83	0.00	1.59	2.27	1.74	92.04	0.00	1.52
MK	0.00	0.00	5.93	0.00	0.00	0.00	94.07	0.00
PD	0.00	1.33	8.58	4.73	0.89	2.51	0.00	81.95
	AN	CC	NN	ZS	AV	CT	MK	PD

Figure 5. Confusion matrix of the first group of the LCF-15 image dataset.

4.3.2. Model Trained on All Species

In order to improve the performance of this model, we evaluate different optimizers and parameters, particularly the loss balance weight λ_o and temperature T.

i. Optimization technique

The choice of the optimizer is a crucial step that greatly influences performance. We first evaluate our approach using different optimizers for loss function (Equation (3)). For this, we set λ_o to 0, and we test different optimization techniques (SGD, Adam, Adamax or RMSprop). Table 2 compares the accuracies with different optimization techniques on the LCF-15 dataset. We can observe that Adam shows the best results for our problem statement. Adam is recommended as the default optimizer for most of the deep learning applications. It inherits the good features of RMSProp and other algorithms. Unlike maintaining a single learning rate through training in SGD, the Adam optimizer updates the learning rate for each network weight individually. We use this optimizer for the rest of the work, and we add the distillation loss term.

Table 2. Comparison of fish recognition accuracies of various optimization techniques on the LCF-15 dataset.

Optimizer	Accuracy
RMSprop	71.79%
Adamax	79.08%
SGD	79.16%
Adam	80.06%

ii. Effect of parameter λ_o

λ_o is a loss balance weight between old and new classes. In order to explore the effect of this balance weight on performance, we evaluate our system using different values of λ_o . Figure 6 illustrates the impact of λ_o on accuracies. It can be seen that the performance gradually decreases with the increase in λ_o . This is because when we increase its value, we favor training the old images over the new images, which makes the system very related to old images. The accuracy is relatively good when $\lambda_o = 0.5$ or 1. We set λ_o to 0.5 for the next experiment.

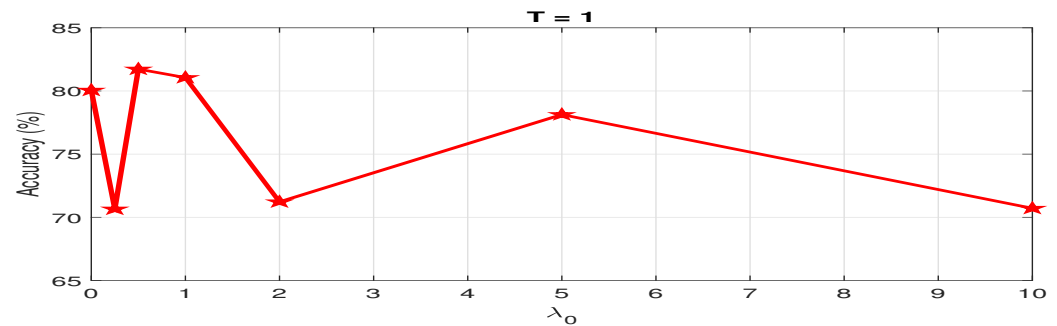


Figure 6. Comparison of accuracies of various values of λ_o for temperature parameter $T = 1$ using the Adam optimizer on the LCF-15 dataset.

iii. Effect of temperature parameter T

We also evaluate the effect of temperature parameter T seen in Equation (2). Figure 7 visualizes the effect of this parameter on the performance of our approach. We can observe that the accuracy is better when $T \geq 1$. Hinton et al. [60] suggest to set $T > 1$, which increases the weight of smaller values and encourages the network to better learn similarities among classes. We achieve the best accuracy of 81.83% for $T = 2$.

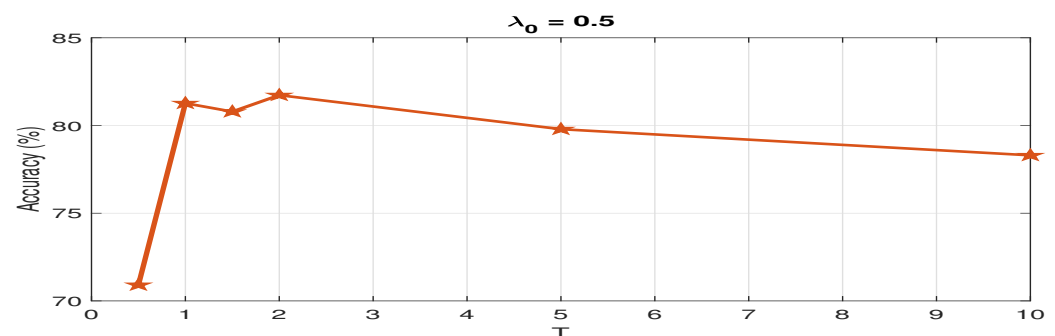


Figure 7. Comparison of accuracies of various values of temperature parameter T for $\lambda_o = 0.5$ using the Adam optimizer on the LCF-15 dataset.

iv. Performance analysis

After analyzing the effects of different parameters, we show results using the following setting: $\lambda_o = 0.5$ and $T = 2$ using the Adam optimizer. We achieve an accuracy of 81.83%, which exceeds that of the non-incremental CNN (76.90%). Figure 8 illustrates the confusion matrix of the model trained on all species. We note that the precision of AN and AV are improved (AN from 43.14% to 44.19% and AV from 87.50% to 100%). The precision of the difficult species is reduced, but they remain higher than those of non-incremental learning (c.f. Figure 4); for example, the precision of the NN species is reduced from 85.58% to 60.99%, which is higher than 25.20% with non-incremental learning. The same for the species CC, the precision is reduced from 87.50% to 62.50%, which is much better than 16.67% with non-incremental learning. Due to the loss function with the knowledge distillation, we instruct the model not to forget too much of the knowledge acquired in the first training. The forgetting of old knowledge is due to the similarities between the old and new species and the fact that the newly added species are more representative. We also note that the precision of some representative fish species is reduced, for example, CL from 97.23% to 87.47% and DR from 93.75% to 82.73%. The network becomes more related to the old species because of the distillation of knowledge loss. We achieve our objective of keeping the model stable when introducing new species while maintaining high performances on the difficult species already learned.

AN	44.19	0.00	10.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	44.96	0.00	0.78
CC	0.00	62.50	16.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20.83	0.00	0.00
NN	3.16	0.85	60.99	0.24	0.00	0.24	0.00	0.85	0.24	0.00	0.00	0.18	32.32	0.43	0.49
ZS	14.44	0.00	13.90	10.16	0.00	3.21	0.00	0.00	0.00	0.00	2.67	0.00	55.61	0.00	0.00
AV	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CT	0.45	0.00	1.06	0.00	3.71	87.79	0.30	0.61	1.52	0.08	0.00	2.35	2.12	0.00	0.00
MK	0.00	0.00	1.69	0.00	0.00	5.93	86.44	0.85	0.00	0.00	0.00	0.00	5.08	0.00	0.00
PD	1.18	0.00	6.80	2.37	0.00	4.44	0.59	75.74	1.33	0.00	0.00	0.15	6.51	0.00	0.89
AC	1.08	0.00	0.00	0.00	0.54	0.72	0.00	4.88	86.44	3.07	0.00	2.35	0.18	0.72	0.00
CL	0.00	0.00	0.05	0.16	0.05	6.45	1.12	1.01	1.01	87.47	0.00	1.01	1.65	0.00	0.00
CS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
DA	0.15	0.00	0.05	0.00	0.84	0.84	0.00	0.10	0.10	0.00	0.00	96.77	0.79	0.00	0.35
DR	0.71	0.45	13.54	0.29	0.04	0.69	0.02	0.06	0.29	0.02	0.00	0.63	82.73	0.00	0.53
HM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
PV	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
	AN	CC	NN	ZS	AV	CT	MK	PD	AC	CL	CS	DA	DR	HM	PV

Figure 8. Confusion matrix of the second group of the LCF-15 image dataset.

4.3.3. Comparative Study

Table 3 shows the comparison performances of our proposed approach with the state-of-the-art methods on the LCF-15 benchmark dataset. We note that we implemented modified AlexNet [24] and FishResNet [28] approaches by using the same provided parameters as in their papers. We also implemented the approach of Yolov3 [19] based on the spatial information using RGB images. From Table 3, we observe that our proposed approach based on incremental learning for live fish species classification outperforms the-state-of-the-art methods. We note that Salman et al. [31] reached an accuracy of 93.65% by testing their model on 7500 fish images issued from the LCF-15 dataset, but these fish images are not from the original test set provided in the dataset. When they tested their method, they did not use the original training and test split provided in the LCF-15 dataset. Furthermore, Jalal et al. [19] did not use the original test set provided in the LCF-15 benchmark dataset. Instead, they merged the training and test sets, and then they took 70% of samples for training and 30% for test. The test set of the LCF-15 benchmark dataset is highly blurry compared with the training set, which explains the high accuracy in their article compared with ours.

Table 3. Comparison of fish recognition accuracies of various methods on the LCF-2015 dataset.

Approach	Accuracy
SURF-SVM [15]	51%
FishResNet [28]	54.24
CNN-SVM [21]	66%
NIN-SVM [34]	69.84%
Modified AlexNet [24]	72.25
Yolov3 [19]	72.63
AdvFish [29]	74.54
ResNet50 (with non incremental learning)	76.90%
PCANET-SVM [34]	77.27%
ResNet50 (with proposed incremental learning)	81.83%

5. Conclusions

In this paper, we proposed a new CNN approach based on incremental learning for a live fish classification task in an unconstrained underwater environment. We proposed to first train ResNet50 on a hard fish species and then add an easy fish species by using incremental learning to reduce the forgotten problem. For this, we modified the loss function by adding knowledge distillation loss. Experiments on the LifeClef 2015 Fish benchmark dataset demonstrated that incremental precisions are higher than non-incremental preci-

sions, and our proposed approach outperforms various state-of-the-art methods for fish species identification.

Our future work will aim to keep the old and new precision of each species higher in order to improve the performance of the system.

Author Contributions: Conceptualization, A.B.T., A.B. and K.N.; methodology, A.B.T., A.B. and K.N.; software, A.B.T., A.B. and K.N.; validation, A.B.T., A.B. and K.N.; formal analysis, A.B.T., A.B. and K.N.; investigation, A.B.T., A.B. and K.N.; resources, A.B.T., A.B. and K.N.; data curation, A.B.T., A.B. and K.N.; writ draft preparation, A.B.T., A.B. and K.N.; writ and editing, A.B.T., A.B. and K.N.; visualization, A.B.T., A.B. and K.N.; supervision, A.B. and K.N.; project administration, A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The LifeClef 2015 Fish dataset (www.imageclef.org/lifeclef/2015/fish accessed on 7 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brandl, S.J.; Goatley, C.H.; Bellwood, D.R.; Tornabene, L. The hidden half: Ecology and evolution of cryptobenthic fishes on coral reefs. *Biol. Rev.* **2018**, *93*, 1846–1873. [[CrossRef](#)] [[PubMed](#)]
2. Johannes, R. Pollution and degradation of coral reef communities. In *Elsevier Oceanography Series*; Elsevier: Amsterdam, The Netherlands, 1975; Volume 12, pp. 13–51.
3. Robinson, J.P.; Williams, I.D.; Edwards, A.M.; McPherson, J.; Yeager, L.; Vigliola, L.; Brainard, R.E.; Baum, J.K. Fishing degrades size structure of coral reef fish communities. *Glob. Chang. Biol.* **2017**, *23*, 1009–1022. [[CrossRef](#)] [[PubMed](#)]
4. Leggat, W.P.; Camp, E.F.; Suggett, D.J.; Heron, S.F.; Fordyce, A.J.; Gardner, S.; Deakin, L.; Turner, M.; Beeching, L.J.; Kuzhiumparambil, U.; et al. Rapid coral decay is associated with marine heatwave mortality events on reefs. *Curr. Biol.* **2019**, *29*, 2723–2730. [[CrossRef](#)]
5. D’agata, S.; Mouillot, D.; Kulbicki, M.; Andréfouët, S.; Bellwood, D.R.; Cinner, J.E.; Cowman, P.F.; Kronen, M.; Pinca, S.; Vigliola, L. Human-mediated loss of phylogenetic and functional diversity in coral reef fishes. *Curr. Biol.* **2014**, *24*, 555–560. [[CrossRef](#)] [[PubMed](#)]
6. Hughes, T.P.; Barnes, M.L.; Bellwood, D.R.; Cinner, J.E.; Cumming, G.S.; Jackson, J.B.; Kleypas, J.; Van De Leemput, I.A.; Lough, J.M.; Morrison, T.H.; et al. Coral reefs in the Anthropocene. *Nature* **2017**, *546*, 82–90. [[CrossRef](#)] [[PubMed](#)]
7. Jackson, J.B.; Kirby, M.X.; Berger, W.H.; Bjorndal, K.A.; Botsford, L.W.; Bourque, B.J.; Bradbury, R.H.; Cooke, R.; Erlandson, J.; Estes, J.A.; et al. Historical overfishing and the recent collapse of coastal ecosystems. *Science* **2001**, *293*, 629–637. [[CrossRef](#)]
8. Jennings, S.; Pinnegar, J.K.; Polunin, N.V.; Warr, K.J. Impacts of trawling disturbance on the trophic structure of benthic invertebrate communities. *Mar. Ecol. Prog. Ser.* **2001**, *213*, 127–142. [[CrossRef](#)]
9. Fernandes, I.; Bastos, Y.; Barreto, D.; Lourenço, L.; Penha, J. The efficacy of clove oil as an anaesthetic and in euthanasia procedure for small-sized tropical fishes. *Braz. J. Biol.* **2016**, *77*, 444–450. [[CrossRef](#)]
10. Thresher, R.E.; Gunn, J.S. Comparative analysis of visual census techniques for highly mobile, reef-associated piscivores (Carangidae). *Environ. Biol. Fishes* **1986**, *17*, 93–116. [[CrossRef](#)]
11. Ben Tamou, A.; Benzinou, A.; Nasreddine, K. Multi-stream fish detection in unconstrained underwater videos by the fusion of two convolutional neural network detectors. *Appl. Intell.* **2021**, *51*, 5809–5821. [[CrossRef](#)]
12. Huang, P.X.; Boom, B.J.; Fisher, R.B. Underwater live fish recognition using a balance-guaranteed optimized tree. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; pp. 422–433.
13. Spampinato, C.; Giordano, D.; Di Salvo, R.; Chen-Burger, Y.H.J.; 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, Firenze, Italy, 29 October 2010; pp. 45–50.
14. Cabrera-Gámez, J.; Castrillón-Santana, M.; Dominguez-Brito, A.; Hernández Sosa, J.D.; Isern-González, J.; Lorenzo-Navarro, J. Exploring the use of local descriptors for fish recognition in lifeclef 2015. In Proceedings of the CEUR Workshop Proceedings, Toledo, Spain, 14 September 2015.
15. Szűcs, G.; Papp, D.; Lovas, D. SVM classification of moving objects tracked by Kalman filter and Hungarian method. In Proceedings of the Working Notes of CLEF 2015 Conference, Toulouse, France, 8–11 September 2015.
16. Hu, K.; Weng, C.; Zhang, Y.; Jin, J.; Xia, Q. An Overview of Underwater Vision Enhancement: From Traditional Methods to Recent Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 241. [[CrossRef](#)]

17. Edge, C.; Islam, M.J.; Morse, C.; Sattar, J. A Generative Approach for Detection-driven Underwater Image Enhancement. *arXiv* **2020**, arXiv:2012.05990.
18. Li, X.; Shang, M.; Qin, H.; Chen, L. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In Proceedings of the OCEANS 2015-MTS/IEEE, Washington, DC, USA, 19–22 October 2015; pp. 1–5.
19. Jalal, A.; Salman, A.; Mian, A.; Shortis, M.; Shafait, F. Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecol. Inform.* **2020**, *57*, 101088. [\[CrossRef\]](#)
20. Zhang, D.; O’Conner, N.E.; Simpson, A.J.; Cao, C.; Little, S.; Wu, B. Coastal fisheries resource monitoring through A deep learning-based underwater video analysis. *Estuar. Coast. Shelf Sci.* **2022**, *269*, 107815. [\[CrossRef\]](#)
21. Jäger, J.; Rodner, E.; Denzler, J.; Wolff, V.; Fricke-Neuderth, K. SeaCLEF 2016: Object Proposal Classification for Fish Detection in Underwater Videos. In Proceedings of the CLEF (Working Notes), Évora, Portugal, 5–8 September 2016; pp. 481–489.
22. Sun, X.; Shi, J.; Liu, L.; Dong, J.; Plant, C.; Wang, X.; Zhou, H. Transferring deep knowledge for object recognition in low-quality underwater videos. *Neurocomputing* **2018**, *275*, 897–908. [\[CrossRef\]](#)
23. Ju, Z.; Xue, Y. Fish species recognition using an improved AlexNet model. *Optik* **2020**, *223*, 165499. [\[CrossRef\]](#)
24. Iqbal, M.A.; Wang, Z.; Ali, Z.A.; Riaz, S. Automatic fish species classification using deep convolutional neural networks. *Wirel. Pers. Commun.* **2021**, *116*, 1043–1053. [\[CrossRef\]](#)
25. Villon, S.; Chaumont, M.; Subsol, G.; Villéger, S.; Claverie, T.; Mouillot, D. Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between Deep Learning and HOG + SVM methods. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Lecce, Italy, 24–27 October 2016; pp. 160–171.
26. Murugaiyan, J.S.; Palaniappan, M.; Durairaj, T.; Muthukumar, V. Fish species recognition using transfer learning techniques. *Int. J. Adv. Intell. Inform.* **2021**, *7*, 188–197. [\[CrossRef\]](#)
27. Mathur, M.; Vasudev, D.; Sahoo, S.; Jain, D.; Goel, N. Crosspooled FishNet: Transfer learning based fish species classification model. *Multimed. Tools Appl.* **2020**, *79*, 31625–31643. [\[CrossRef\]](#)
28. Mathur, M.; Goel, N. FishResNet: Automatic Fish Classification Approach in Underwater Scenario. *SN Comput. Sci.* **2021**, *2*, 273. [\[CrossRef\]](#)
29. Zhang, Z.; Du, X.; Jin, L.; Wang, S.; Wang, L.; Liu, X. Large-scale underwater fish recognition via deep adversarial learning. *Knowl. Inf. Syst.* **2022**, *64*, 353–379. [\[CrossRef\]](#)
30. Qin, H.; Li, X.; Yang, Z.; Shang, M. When underwater imagery analysis meets deep learning: A solution at the age of big visual data. In Proceedings of the OCEANS 2015-MTS/IEEE, Washington, DC, USA, 19–22 October 2015; pp. 1–5.
31. Salman, A.; Jalal, A.; Shafait, F.; Mian, A.; Shortis, M.; Seager, J.; Harvey, E. Fish species classification in unconstrained underwater environments based on deep learning. *Limnol. Oceanogr. Methods* **2016**, *14*, 570–585. [\[CrossRef\]](#)
32. Paraschiv, M.; Padrino, R.; Casari, P.; Bigal, E.; Scheinin, A.; Tchernov, D.; Fernández Anta, A. Classification of Underwater Fish Images and Videos via Very Small Convolutional Neural Networks. *J. Mar. Sci. Eng.* **2022**, *10*, 736. [\[CrossRef\]](#)
33. Qin, H.; Li, X.; Liang, J.; Peng, Y.; Zhang, C. DeepFish: Accurate underwater live fish recognition with a deep architecture. *Neurocomputing* **2016**, *187*, 49–58. [\[CrossRef\]](#)
34. Sun, X.; Shi, J.; Dong, J.; Wang, X. Fish recognition from low-resolution underwater images. In Proceedings of the 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, China, 15–17 October 2016; pp. 471–476.
35. Zhao, Z.; Liu, Y.; Sun, X.; Liu, J.; Yang, X.; Zhou, C. Composited FishNet: Fish detection and species recognition from low-quality underwater videos. *IEEE Trans. Image Process.* **2021**, *30*, 4719–4734. [\[CrossRef\]](#)
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
37. Bay, H.; Tuytelaars, T.; Gool, L.V. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
38. Dhar, P.; Guha, S. Fish Image Classification by XgBoost Based on Gist and GLCM Features. *Int. J. Inf. Technol. Comput. Sci.* **2021**, *4*, 17–23. [\[CrossRef\]](#)
39. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
40. Grauman, K.; Darrell, T. The pyramid match kernel: Discriminative classification with sets of image features. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05), Beijing, China, 17–21 October 2005; Volume 1–2. pp. 1458–1465.
41. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [\[CrossRef\]](#)
42. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
43. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Available online: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf> (accessed on 7 July 2022).
44. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [\[CrossRef\]](#)
45. Pang, J.; Liu, W.; Liu, B.; Tao, D.; Zhang, K.; Lu, X. Interference Distillation for Underwater Fish Recognition. In Proceedings of the Asian Conference on Pattern Recognition, Macau SAR, China, 4–8 December 2022; pp. 62–74.

46. Cheng, L.; He, C. Fish Recognition Based on Deep Residual Shrinkage Network. In Proceedings of the 2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE), Wuhan, China, 4–6 November 2021; pp. 36–39.
47. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
48. Knausgård, K.M.; Wiklund, A.; Sjørdalen, T.K.; Halvorsen, K.T.; Kleiven, A.R.; Jiao, L.; Goodwin, M. Temperate fish detection and classification: A deep learning based approach. *Appl. Intell.* **2022**, *52*, 6988–7001. [[CrossRef](#)]
49. Redmon, J.; Farhadi, A. Yolo3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
50. Olsvik, E.; Trinh, C.; Knausgård, K.M.; Wiklund, A.; Sjørdalen, T.K.; Kleiven, A.R.; Jiao, L.; Goodwin, M. Biometric fish classification of temperate species using convolutional neural network with squeeze-and-excitation. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Graz, Austria, 9–11 July 2019; pp. 89–101.
51. Mittal, S.; Srivastava, S.; Jayanth, J.P. A Survey of Deep Learning Techniques for Underwater Image Classification. Available online: https://www.researchgate.net/profile/Sparsh-Mittal-2/publication/357826927_A_Survey_of_Deep_Learning_Techniques_for_Underwater_Image_Classification/links/61e145aec5e310337591ec08/A-Survey-of-Deep-Learning-Techniques-for-Underwater-Image-Classification.pdf (accessed on 7 July 2022).
52. Saleh, A.; Sheaves, M.; Rahimi Azghadi, M. Computer vision and deep learning for fish classification in underwater habitats: A survey. *Fish Fish.* **2022**, *23*, 977–999. [[CrossRef](#)]
53. Shmelkov, K.; Schmid, C.; Alahari, K. Incremental learning of object detectors without catastrophic forgetting. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3400–3409.
54. Xiao, T.; Zhang, J.; Yang, K.; Peng, Y.; Zhang, Z. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 177–186.
55. Polikar, R.; Upda, L.; Upda, S.S.; Honavar, V. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2001**, *31*, 497–508. [[CrossRef](#)]
56. Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive neural networks. *arXiv* **2016**, arXiv:1606.04671.
57. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [[CrossRef](#)]
58. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2935–2947. [[CrossRef](#)]
59. Hayes, T.L.; Cahill, N.D.; Kanan, C. Memory efficient experience replay for streaming learning. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019; pp. 9769–9776.
60. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
61. Boom, B.J.; Huang, P.X.; He, J.; Fisher, R.B. Supporting ground-truth annotation of image datasets using clustering. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 1542–1545.