fishes

MDPI

Article

Fish Recognition in the Underwater Environment Using an Improved ArcFace Loss for Precision Aquaculture

Liang Liu [1,2,3], Junfeng Wu [1,2,3,*], Tao Zheng [1,2,3], Haiyan Zhao [1,2,3], Han Kong [1,2,3], Boyu Qu [1,2,3] and Hong Yu [1,2,3]

[1] College of Information Engineering, Dalian Ocean University, Dalian 116023, China; liangliu0203@gmail.com (L.L.); ztan0414luck@gmail.com (T.Z.); zhaohaiyan0803@gmail.com (H.Z.); konghan5219@gmail.com (H.K.); quboyu1208@gmail.com (B.Q.); yuhong@dlou.edu.cn (H.Y.)
[2] Dalian Key Laboratory of Smart Fisheries, Dalian 116023, China
[3] Key Laboratory of Environment Controlled Aquaculture, Dalian Ocean University, Ministry of Education, Dalian 116023, China
* Correspondence: wujunfeng@dlou.edu.cn

Abstract: Accurate fish individual recognition is one of the critical technologies for large-scale fishery farming when trying to achieve accurate, green farming and sustainable development. It is an essential link for aquaculture to move toward automation and intelligence. However, existing fish individual data collection methods cannot cope with the interference of light, blur, and pose in the natural underwater environment, which makes the captured fish individual images of poor quality. These low-quality images can cause significant interference with the training of recognition networks. In order to solve the above problems, this paper proposes an underwater fish individual recognition method (FishFace) that combines data quality assessment and loss weighting. First, we introduce the Gem pooing and quality evaluation module, which is based on EfficientNet. This module is an improved fish recognition network that can evaluate the quality of fish images well, and it does not need additional labels; second, we propose a new loss function, FishFace Loss, which will weigh the loss according to the quality of the image so that the model focuses more on recognizable fish images, and less on images that are difficult to recognize. Finally, we collect a dataset for fish individual recognition (WideFish), which contains and annotates 5000 images of 300 fish. The experimental results show that, compared with the state-of-the-art individual recognition methods, Rank1 accuracy is improved by 2.60% and 3.12% on the public dataset DlouFish and the proposed WideFish dataset, respectively.

Keywords: deep learning; convolutional neural network; biometric recognition; fish individual recognition

Key Contribution: We propose a method for the recognition of underwater fish individuals that combines data quality assessment and loss weighting to address the interference that low-quality images can bring to the training of recognition networks.

Citation: Liu, L.; Wu, J.; Zheng, T.; Zhao, H.; Kong, H.; Qu, B.; Yu, H. Fish Recognition in the Underwater Environment Using an Improved ArcFace Loss for Precision Aquaculture. *Fishes* **2023**, *8*, 591. https://doi.org/10.3390/fishes8120591

Academic Editors: Jun Jiang and Dimitrios Moutopoulos

Received: 26 October 2023
Revised: 20 November 2023
Accepted: 28 November 2023
Published: 30 November 2023

1. Introduction

Industrialization is the new trend in aquaculture, and precision aquaculture is at the forefront of this industrial revolution. Disease detection, the accurate estimation of fish length and weight, as well as fish behavior recognition are all vital components of precision aquaculture [1]. At the core of these processes lies the precise recognition of individual fish, which serves as the foundation for achieving optimal results in the industry [2].

Traditional fish recognition primarily relies on radio frequency technology (RFID) tagging for each fish [3]. However, this approach has several drawbacks. The installation of these low-cost tags at the tail of the fish through perforation is time-consuming and labor-intensive for farming organizations. Furthermore, this method can significantly harm the fish's well-being. Additionally, due to the aquatic environment in which fish reside, the

presence of water weakens magnetic field strength, thus resulting in a limited reading and writing range for RFID technology (which is typically used up to a maximum of 10 cm). Consequently, RFID technology is unsuitable for precise individual fish recognition tasks.

Current research on individual fish recognition utilizes face recognition methods for the purpose of fish recognition [4]. The overall process, as illustrated in (Figure 1), consists of two main steps: fish detection and fish recognition. The fish detection is based on finding the position and boundary of the fish in the picture, and the fish recognition is based on judging the individual based on its features. However, the application of face recognition methods to the practice of fish recognition poses challenges. The primary reason for this is that the accuracy of face recognition heavily relies on the quality of face images [5]. Better image quality translates to improved performance in recognition tasks. Unfortunately, most fish images are captured underwater, where they are susceptible to various interferences such as illumination, blur, and pose. Consequently, the collected dataset often contains numerous low-quality images. During the training process of the recognition network, these low-quality images act as noise. If not properly addressed [6], they can degrade the overall performance of fish recognition.

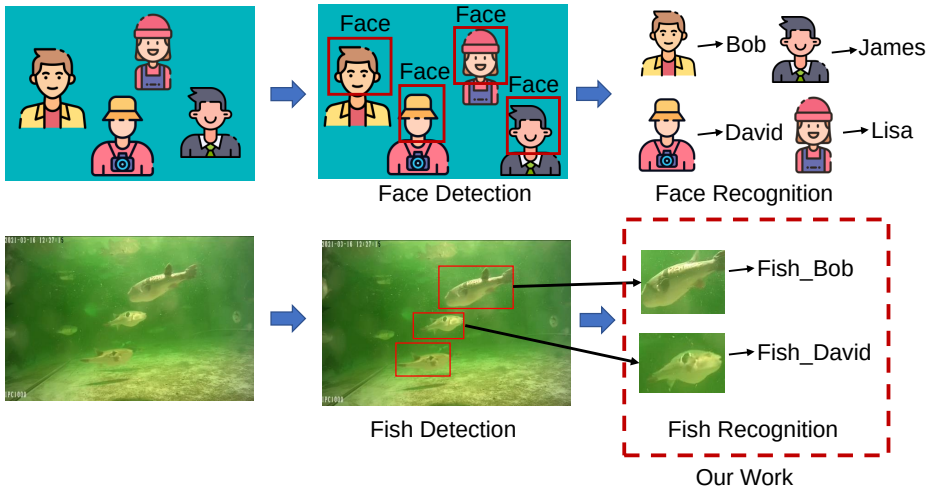

Figure 1. Flowchart of the face recognition and fish individual recognition processes.

To address the aforementioned challenges, Wang proposed a novel method for the real-time individual recognition of swimming fish, which is based on an improved version of YOLOv5 [7]. This method utilizes a convolutional neural network with an embedded attention module to detect and recognize fish in underwater images. Petrellis [8] proposed a shape alignment technique based on a regression tree ensemble machine learning method to solve the problem where fish are almost indistinguishable from the background in low-contrast underwater images. Khan [9] presented the design of FishNet, an automated monitoring system capable of identifying, localizing, and predicting aquatic species and their functional traits; in addition, they open sourced a massively diverse dataset containing 94,532 well-organized images from 17,357 aquatic species. Similarly, Yin introduced LIFRNet, a lightweight backbone network designed specifically for fish visual feature extraction [10]. By incorporating deformable convolution and edge feature learning, LIFRNet adapts to different fish shapes and poses while enhancing feature discrimination. Additionally, Gao proposed FIRN, a new network dedicated to fish detection [11]. FIRN leverages an anchor-free approach and a feature pyramid network to improve accuracy and speed. While these methods demonstrate excellent performance in identifying individual fish under ideal conditions, they often overlook the challenges posed by real underwater environments, such as variations in lighting, blurring, and fish pose. Furthermore, there is currently no robust solution to mitigate the negative impact of low-quality fish data on model training results.

To leverage the potential of low-quality individual fish images in environments where obtaining clear images is challenging, this paper presents FishFace, an underwater fish individual recognition method that incorporates data quality assessment and loss weighting. Through an in-depth analysis of existing individual fish recognition methods, FishFace addresses the limitations by modifying the existing ArcFace loss function [12] to account for the quality factors of underwater input images. The key component of the FishFace loss function is an adjustable hypersphere radius that dynamically adapts based on the calculated image quality index. This adaptive adjustment enables more accurate fish individual recognition, even in low-quality images. Furthermore, FishFace incorporates a quality assessment module in the network architecture to calculate the image quality index used by the FishFace loss. The main contributions of this paper are as follows:

- We designed a fish individual recognition network with a quality assessment module, which can evaluate the quality of fish images well and does not require additional labeling.
- We propose a new loss function named FishFace Loss, which will weigh the loss according to the quality of the image so that the model focuses more on recognizable fish images and less on ideas that are difficult to recognize.
- We collected a dataset for fish individual recognition (WideFish), which contains and annotates 5000 images of 300 fish. This dataset was created to help train and test the fish individual recognition method.

2. Material and Methods

2.1. Data Preparation

Data collection: To meet production needs, it was necessary to construct datasets for training the model on actual aquaculture environments. In this study, we created a fish individual recognition dataset (WideFish), which includes images taken from real scenes and images downloaded from video websites. The real scene videos was taken from the puffer fish breeding pond of Dalian Tianzheng Industrial Co. (Dalian, China), and the acquisition device is shown in Figure 2. The size of the breeding pool was 10 m $\times$ 10 m, the water depth was 1.5 m, and the distance of the lamp from the pool was 3.5 m. Two 8-megapixel underwater cameras were equipped, which were located at the center of the pool and the top of the inner wall on one side for acquiring the video data, which were then captured and uploaded through the monitoring equipment. The data downloaded from the website came from video platforms such as Instagram, YouTube, and Bilibili. The size of all collected videos ranged from 640 $\times$ 480 to 1920 $\times$ 1080 pixels depending on the source of the image. We extracted frames from all collected video clips at 1 fps, as well as manually removed blurred and duplicated images to obtain images as those shown in Figure 3a. It is worth noting that no color calibration was performed during the shooting process.

Data cropping: The images obtained during the data collection phase contained multiple fish while we required single fish images for our final training. Therefore, we needed to crop the data. The cropping method employed was a semi-automatic approach. Specifically, we utilized the YOLOv8 network [13] to detect objects in the processed images. Figure 3b illustrates the detection results of YOLOv8, and then the detected fish were automatically cropped into individual images as shown in Figure 3c.

Data labeling: The cropped images were first collected and then resized to 224 $\times$ 224 pixels. We manually organized fish images with the same ID into corresponding folders, which were named 1 to 300 in sequential order. This process completed the construction of the WideFish dataset. The final dataset consists of 5000 fish images of different qualities, including 1300 koi images, 1200 puffer fish images, 1100 clown fish images, and 1400 grass carp images. The sample distribution of the dataset is shown in Figure 4. In order to facilitate training and testing, we divided the 5000 images into a training set and a testing set. The training set contains 4000 fish images and the test set contains 1000 fish images.

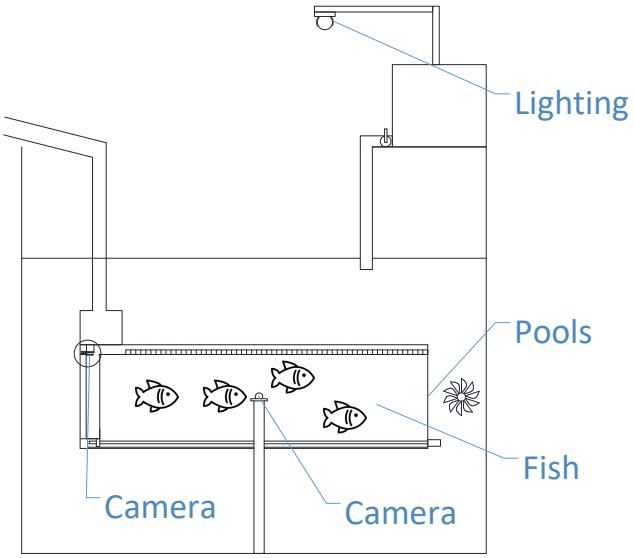

Figure 2. Experimental setup for fish image acquisition in real aquaculture scenarios.

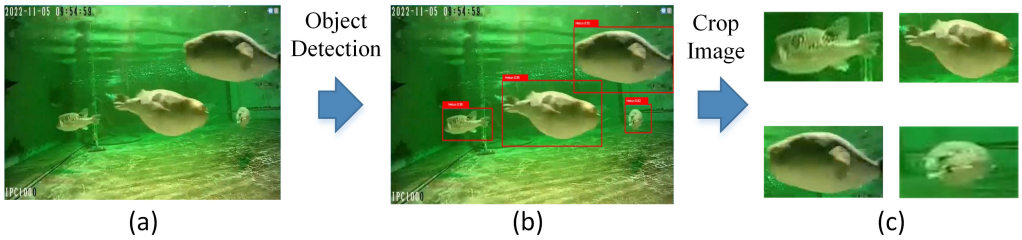

Figure 3. WideFish dataset production process (**a**–**c**).

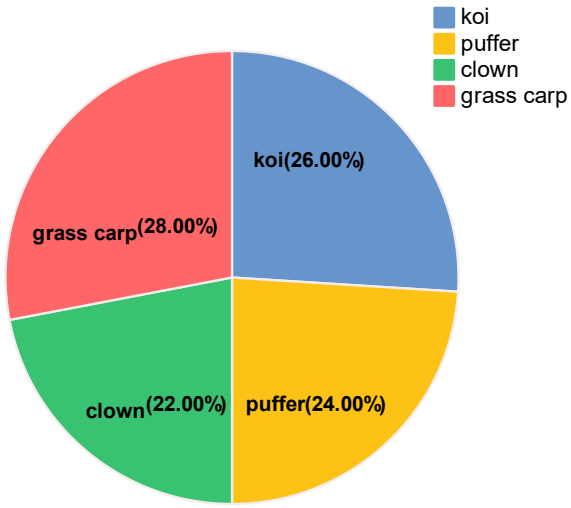

Figure 4. Sample distribution of the WideFish dataset.

2.2. The Proposed Method

In order to solve the problem that low-quality images can bring interference to the training of recognition networks, this paper designed a fish individual recognition network framework (FishFace) with a quality evaluation module, as shown in Figure 5. The whole process is divided into 4 steps.

Step 1: The input of the model is a 224 × 224 image of an individual fish, and the features of the image are first extracted by a backbone network (EfficientNet-B5) [14] to obtain a feature map with information about the individual fish, wherein the last layer of the backbone network is a fully connected layer.

Step 2: The feature maps learned by the backbone network are passed to the quality assessment module to calculate the fish image quality s_i.

Step 3: The feature maps obtained from the backbone network and the fish image quality s_i are passed into the FishFace loss program for model parameter optimization.

Step 4: The final weights are obtained by the network through multiple iterations, and the final weights can be effective for the individual recognition of different quality fish images.

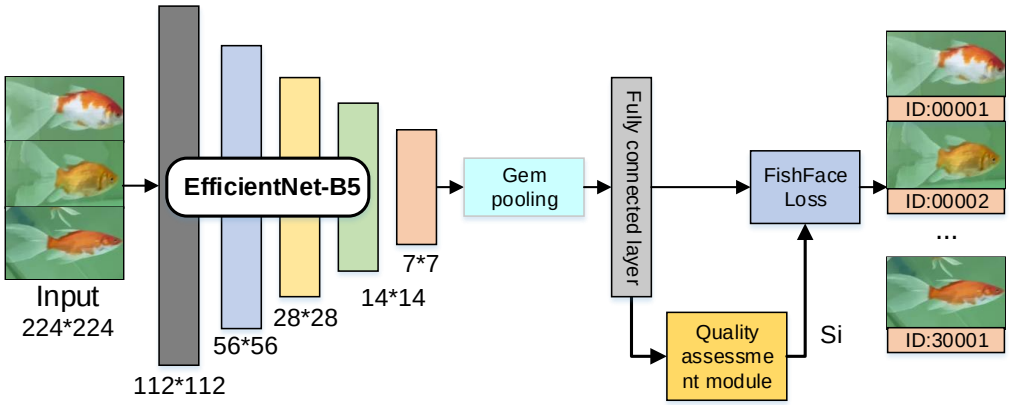

Figure 5. FishFace network structure diagram.

2.3. Improved Feature Extraction Module

The feature extraction part of FishFace was chosen from EfficientNet-B5, a convolutional neural network model proposed by Tan, which achieves a higher performance and efficiency by adjusting the depth, width, and resolution of the network throughout the model. The network structure of Efficientnet-B5 is based on the reverse bottleneck residual block [15] and squeeze-and-excitation [16] block of MobileNetV2 [17], which has 39 convolutional layers and 4 fully connected layers. Compared with other feature extraction models, such as ResNet and DenseNet [18], etc., Efficientnet-B5 uses an AutoML-based model scaling method to find the best network structure under different resource constraints, thus improving accuracy and efficiency. And it uses mixed precision training and tensor cores, which can accelerate the training process. Furthermore, in our preliminary experiments, we found that EfficientNet-B5 provided a good balance between accuracy and efficiency for our fish recognition task, thus making it ideal for our proposed framework.

Unlike the original EfficientNet, we changed all of the global average pooling (GAP) in the network to GeM pooling [19]. The purpose of doing so is that GAP would dilute the combinatorial relationship between the relative positions of different features, which may lead to the loss of some spatial information. And GAP would treat all features equally, which may ignore some of the parts that have more differentiation and robustness.

On the other hand, GeM pooling can retain the essential attributes of the input feature map while amplifying the activation of features with a greater intensity so as to improve the differentiation and robustness of individual fish features. The specific GeM pooling equation is as follows.

$$\mathbf{f}^{(g)} = \left[\mathrm{f}_1^{(g)} \ldots \mathrm{f}_k^{(g)} \ldots \mathrm{f}_K^{(g)}\right]^\top, \mathrm{f}_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k}\right)^{\frac{1}{p_k}} \quad (1)$$

where k represents the index of elements in the feature vector and K is the total number of elements in the feature vector. For each feature map $\mathcal{X}_k$, we obtain the long vector representation of the image by varying p_k, where $p_k \to +\infty$ is the max pooling and $p_k = 1$ is the average pooling. In this paper, the best experimental results were obtained for $p_k = 3$.

The improved EfficientNet-B5 network structure is shown in Table 1. We can see that the network mainly consists of 39 MBConv modules and GeM pooling. Among

them, MBConv is a kind of inverted residual module, which includes the following: a 1 × 1 expansion convolution that expands the input channel number by n times; a 3 × 3 depthwise separable convolution that performs convolution operations on each channel; a squeeze-and-excitation module that adaptively re-weights each channel; a 1 × 1 compression convolution that restores the output channel number to its original size; and a residual connection that adds the input and output. It is worth noting that MBConv1 expands the input channel number by 1 times, and MBConv6 expands the input channel number by 6 times.

Table 1. Improved EfficientNet-B5 network architecture.

	Modules	Resolution	Number of Channels	Number of Layers
1	Conv 3 × 3	224 × 224	32	1
2	MBConv1, 3 × 3	112 × 112	24	3
3	MBConv6, 3 × 3	56 × 56	40	5
4	MBConv6, 5 × 5	28 × 28	64	5
5	MBConv6, 3 × 3	14 × 14	128	7
6	MBConv6, 5 × 5	14 × 14	176	7
7	MBConv6, 5 × 5	7 × 7	304	9
8	MBConv6, 3 × 3	7 × 7	512	3
9	Conv 1 × 1 and Gempooling and FC	7 × 7	1280	1

2.4. Quality Assessment Module

We designed the quality assessment module (shown in Figure 6) as a branch of the backbone to evaluate the quality of fish pictures. The whole quality assessment module is divided into two parts: the first part is shown in Equation (2), specifically the 1 × 1 convolution layer and Softmax; the second part is shown in Equation (3), specifically two 1 × 1 convolution layers, where a BN layer with ReLU activation function is added after the first convolution layer. A Sigmoid layer was added after the second convolution layer. Finally, the image quality output S_i takes values in the range of $[0, 1]$.

$$y_i = Sigmoid(W_1 x_i) + x_i \tag{2}$$

$$S_i = Sigmoid(W_3 ReLU(LN(W_2 y_i))) \tag{3}$$

where x_i is the feature map passed in through the fully connected layer, y_i is the feature map with contextual information, W_1, W_2, W_3 are both examples of a 1 × 1 convolution, the Sigmoid and ReLU are activation functions, and LN is the BatchNorm.

The design idea of the quality assessment module comes from SENet [16], which can adjust the interdependencies between the channels by a weight calibration of different channels based on global contextual information. Currently, some researchers have used SENet to evaluate the quality of images. However, we found, in our design, that SENet includes a downscaling process in the fully connected layer to obtain contextual information, which destroys the direct correspondence between channels and their weights.

To solve this problem in SENet, we designed a quality assessment module by replacing the original fully connected layer of SENet with a 1 × 1 convolution layer, which will not destroy the spatial structure of fish images. In addition, after passing into the quality evaluation module, the feature maps were firstly passed through a 1 × 1 convolution with a Sigmoid layer, which created the feature maps that were involved in image quality evaluation with global contextual features. The purpose of this was to improve the sensitivity and adaptability of the feature maps to image quality changes, as well as to enhance the interaction and information transmission between the feature maps.

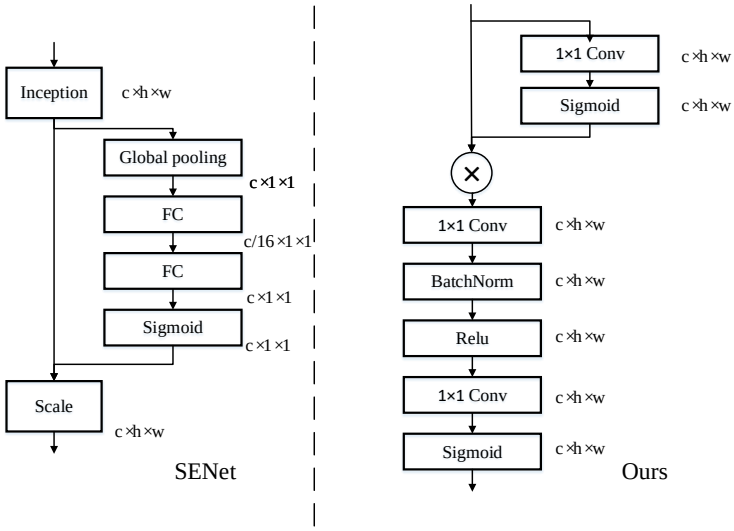

Figure 6. Quality assessment module.

2.5. FishFace Loss

First, before introducing the FishFace Loss proposed in this paper, we need to briefly describe the two most common loss functions in face recognition: Softmax loss [20] and ArcFace Loss.

The traditional classification loss is usually a Softmax loss, as shown in Equation (4). Softmax loss has features such as easy optimization and fast convergence for classification tasks.

Let $f_i \in R^d$ denote the feature vector of the sample x_i, which belongs to the y_i-th class. The feature vector dimension d is typically 512. Let $W_i \in R^d$ denote the j-th column of the weight $W \in R^{d \times n}$, and let $b_i \in R^n$ denote the bias term. Let the class number be n. The softmaxloss is expressed as follows:

$$L_{\text{Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{yi}^T f_i + b_{yi}}}{\sum_{j=1}^{n} e^{W_j^T f_i + b_{yi}}} \tag{4}$$

However, the results are often poor when the Softmax loss is directly used in face recognition tasks. The reason for this is that Softmax loss aims to maximize the log likelihood of all categories in the probability space, i.e., to ensure that all classes are correctly classified. However, it cannot explicitly optimize inter-class and intra-class distances like metric learning.

In response, Deng proposed ArcFace loss, as shown in Equation (5); it is based on Softmax loss, and it is a loss function that is designed to improve the inter-class differentiability while reducing the intra-class distance. Specifically, first, the ArcFace loss makes the bias b of Softmax equal to 0. Then, $W_i^T f_i$ is changed to a $\|W_i\|\|x_i\|\cos(\theta_i)$ conversion. θ_i represents the angle between W_i and x_i. Second, the weights are normalized to the features, i.e., $\|W_i\| = \|f_i\| = 1$. In this case, the prediction depends only on the angle between the features and the weights. Then, the features are multiplied by a constant, i.e., we learn that the features are distributed over a hypersphere of radius S. Finally, an additional boundary penalty m is added to the angle between W_i and f_i. The boundary penalty m is added to the angle theta between the feature vector and the corresponding weight vector of the correct class. The purpose of adding the boundary penalty is to increase the margin between the classes, thus making the decision boundary more distinct. The range of the m value can vary depending on the application and dataset. In general, the value of m is chosen based on the characteristics of the dataset and the complexity of the classification problem. A larger value of m may be suitable for more complex datasets with larger variations in the feature vectors, while a smaller value may be sufficient for simpler datasets.

$$L_{\text{arcface}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{S(\cos(\theta_{yi}+m))}}{e^{S(\cos(\theta_{yi}+m))}+\sum_{j\neq y_i}e^{S\cos(\theta_j)}} \quad (5)$$

By combining all marginal penalties, CosFace [21], SphereFace [22], and ArcFace were all implemented in a federated framework with m_1, m_2, and m_3 as hyper-parameters:

$$L_{\text{arcface}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{S(m_1\cos(\theta_{yi}+m_2)-m_3)}}{e^{S(m_1\cos(\theta_{yi}+m_2)-m_3)}+\sum_{j\neq y_i}e^{S\cos(\theta_j)}} \quad (6)$$

However, the ArcFace loss was set on the premise that the quality of the face pictures was the same and does not take into account the quality differences between samples; thus, it is not suitable for individual fish recognition tasks.

In response to the problems of ArcFace Loss, this paper propose a loss function FishFace loss, as shown in Equation (7). The loss can enhance the intra-class distance while reducing the inter-class difference on the one hand, and it can allow the quality weight to be well quantified on the other hand.

$$L_{\text{ours}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s_iS(m_1\cos(\theta_{yi}+m_2)-m_3)}}{e^{s_iS(m_1\cos(\theta_{yi}+m_2)-m_3)}+\sum_{j\neq y_i}e^{s_iS\cos(\theta_j)}} \quad (7)$$

where S_i is the output of the quality evaluation module with a value range between $[0, 1]$. S is a fixed value, which is equivalent to the inverse of the minimum variation value in Gaussian distribution, and S takes the value of 64. m_1, m_2, and m_3 are hyper-parameters. During the training process, low-quality fish pictures have a smaller S_i and high-quality pictures have a larger S_i. In FishFace loss, each sample can be used as its weight according to its own quality, which can improve the recognition accuracy.

2.6. FishFace Training Strategy

The model training process is shown in Figure 7, which is divided into three main steps and is different from the traditional deep learning training methods.

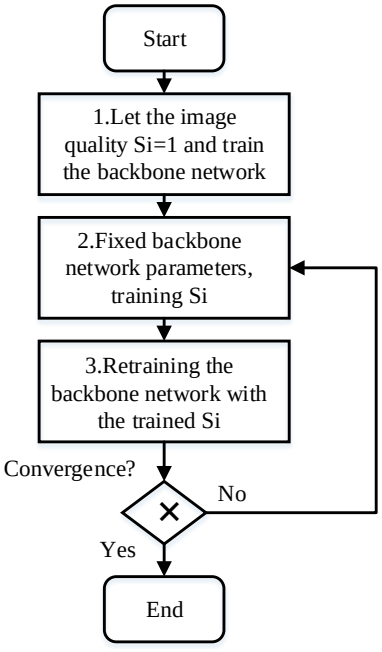

Figure 7. FishFace network training flow chart.

Step 1: In this step, the image quality s_i is fixed at 1 for all training images, which is consistent with the normal training process for individual fish recognition. The primary objective is to train the feature extraction model (M). It is important to note that the quality assessment module does not play a role in this step.

Step 2: Here, the feature extraction model (M) is fixed, and the quality assessment module is incorporated into the training process. The loss function used in this step is given by Formula 6. The main goal is to learn the relevant gradients within the quality assessment module.

Step 3: With the relevant parameters of the quality assessment module fixed, the backbone network is retrained by taking into account the individual fish image quality s_i that corresponds to each sample. The quality assessment module actively participates in the training process.

Once the training process is completed, an evaluation is conducted to determine if the model has converged. If convergence has not been achieved, Steps 2 and 3 need to be iterated until convergence is attained.

The FishFace network adopts a three-step training instead of an end-to-end training approach to avoid the problem of individual fish quality s_i approaching 0. If $s_i = 1$ is not set first, the whole network will train s_i to tend to 0 in order to minimize the loss function. This will cause all fish body images to have very low quality values and no difference. Therefore, we need to set $s_i = 1$ at the beginning stage of training to train the backbone network. After training for a period of time, the backbone network parameters are fixed and s_i is learned. In this way, the image quality values obtained have large differences and good discrimination. By using this three-step training method, the FishFace network can effectively extract individual fish features and perform classification.

2.7. Experimental Setup

The experiments in this paper were conducted in a software environment of Python 3.7 and PyTorch 1.8 on Windows 10 with a computer configured with an Intel Core i7-9700K CPU, 16 GB of RAM, and a GeForce RTX 2080 Ti GPU. Our model training employed the self-made dataset WideFish and the public dataset DlouFish [10]. Data augmentation operations included rotation, translation, scaling, and flipping, which were performed on both datasets. The whole network is initialized as a Gaussian distribution with the mean of the weights being 0 and the standard deviation being 0.1. The loss function is FishFace loss, the optimizer is SGD, the Momentum is 0.9, and the Batchsize is 64. The initial learning rate is 0.01. In the FishFace loss function, through repeated tuning trials, we used $m_1 = 1.0$, $m_2 = 0.3$, and $m_3 = 0.2$, where m_1, m_2, and m_3, respectively, control the angular residuals, the scale factor, and the feature normalization. The algorithm evaluation metrics are Rank1 accuracy and Rank5 accuracy. Specifically, Rank1 accuracy is the percentage of times that the system correctly identifies the fish as the top match, and Rank5 accuracy is the percentage of times that the system correctly identifies the fish among the top five matches.

3. Results

3.1. Performance Comparison between External Models

To evaluate the effectiveness of our FishFace fish individual recognition method, we conducted experiments on DlouFish [10] and WideFish, and then compared our results with state-of-the-art face recognition methods such as Center Loss [23], SphereFace, ArcFace, VGGFace2 [24], Confidence Loss [25], as well as with individual fish recognition methods like LIFRNet and FIRN. Center Loss minimizes the distance between each sample and its corresponding class center while maximizing the distance between different class centers; SphereFace forces the features to be distributed on a hyperspherical manifold by applying an angular margin between the features and the weights of the softmax classifier; ArcFace adds an additional angular margin to make the decision boundaries clearer; VGGFace2 improves feature extraction by improving the VGG network; and Confidence Loss incorporates the confidence of each sample into the softmax loss to minimize the effect of

noisy or hard samples. LIFRNet employs a lightweight deformable convolutional network as the backbone network for recognition, thus effectively capturing the edge information of the fish, which, consequently, enhances fish discrimination. FIRN introduces dilated convolution in the residual block, thus increasing the receptive field and improving the feature extraction. The results are summarized in Table 2, and it shows that our method outperformed all the other methods in terms of Rank1 accuracy with an improvement of 2.60% and 3.12%, and Rank5 accuracy with an improvement of 2.21% and 2.71%. In addition, our algorithm processes an image in an average of only 0.05 seconds, which is second only to ArcFace in terms of speed when compared to comparative methods. These results show that our algorithm can achieve highly accurate and fast fish individual recognition in underwater environments.

Table 2. Performance evaluation of different methods on the WideFish and DlouFish datasets.

Family	Method	WideFish Dataset		DlouFish Dataset		Fps
		Rank1	Rank5	Rank1	Rank5	
Face Recognition Method	Center Loss	87.17	89.72	89.38	91.54	18.5
	SphereFace	89.21	90.24	91.01	92.12	19.3
	ArcFace	90.43	92.38	93.21	93.49	20.1
	VGGFace2	91.72	90.83	92.09	92.11	16.2
	Confidence Loss	91.44	94.94	92.27	94.50	18.5
Fish Recognition Method	LIFRNet	91.34	93.13	90.04	91.10	19.1
	FIRN	90.10	91.17	91.32	92.06	17.6
Proposed method	Ours	94.83	97.64	95.81	96.61	19.4

In order to evaluate the robustness of the method proposed in this paper, experiments were conducted to be compared with three other state-of-the-art algorithms—Center Loss, ArcFace, and VGGFace2—on three commonly used face datasets: LFW [26], CALFW [27], and CPLFW [28]. The results of these experiments are presented in Table 3, and it can be seen from the table that the proposed method performed best or second best in terms of accuracy in Rank1 compared to the other algorithms. These results demonstrate the effectiveness and robustness of the proposed method in handling facial recognition tasks on these datasets. It is worth adding that the datasets commonly used for experiments on low-quality face recognition algorithms are LFW, CALFW, and CPLFW.

Table 3. Performance verification of the FishFace method with different face datasets.

Method	LFW	CALFW	CPLFW
Center Loss	98.75	85.48	77.48
ArcFace	99.83	95.45	92.08
VGGFace2	99.43	90.57	84.01
Ours	99.71	95.91	93.02

3.2. Validation of Internal Modules

To verify the effectiveness of the quality assessment module proposed in this paper, we performed a quality assessment on 16 fish images, which were automatically calculated by the quality assessment module in the FishFace network. The module outputs the quality value of each input image, which ranges from 0 to 1. The quality score optimizes the training process in two ways. First, the value is used as the weight of the FishFace single fish recognition loss, that is, the higher the quality, the greater the loss. In this way, the network learns to pay more attention to high-quality images and ignore low-quality images during the training process. This also conforms to the intuition that high-quality images contain more fish individual recognition information than low-quality images. Secondly, the image quality is used to adjust the sampling strategy of the training data. We use

the image quality as the sampling probability of each sample, which means that high-quality samples have more chances to be selected in the training batches than low-quality samples. This sampling strategy ensures that the model is trained on more representative and informative samples, as well as avoids the negative effects of noise and low-quality samples. The evaluation results are shown in Figure 8. It can be seen that the higher the quality of the fish images, the higher the score; conversely, the lower the quality, the lower the score. In addition, for the same fish, the more positive the image shooting angle, the higher the quality score. And by comparing the training process of Figure 9 and ArcFace, the effectiveness of the quality assessment module was verified.

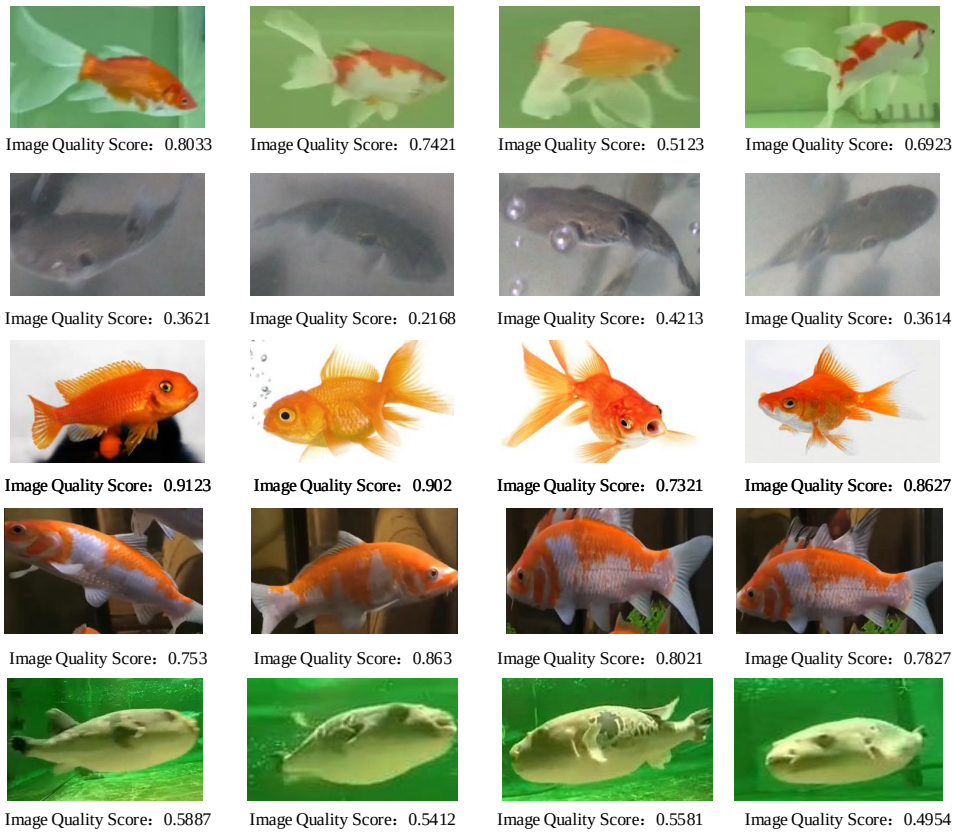

Figure 8. Results of the image quality assessment of fish faces with different qualities.

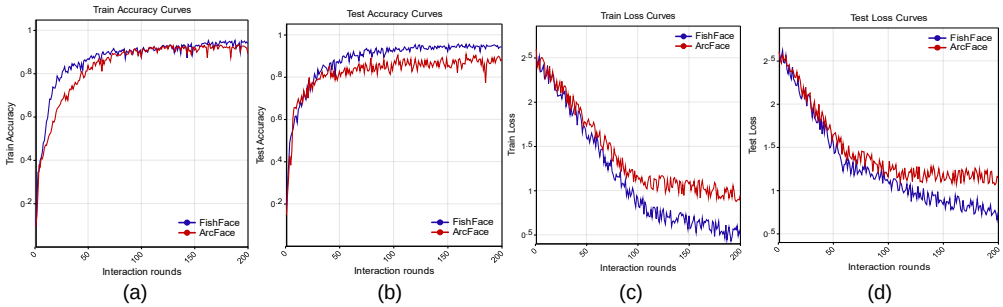

Figure 9. Comparison of the FishFace loss and ArcFace loss training processes (**a**–**d**).

In addition to this, we designed a variant model without the quality assessment module, as shown in Table 4, and the performance of the model had a large fallback in recognition accuracy when the model was not incorporated with the quality assessment module. The experimental results show that the recognition effect of adding the quality assessment module was better than that without the quality assessment module in terms of accuracy and robustness, which proves the effectiveness of the quality assessment module in the fish body recognition problem.

Table 4. Impact of the QA module on performance.

Method	WideFish Dataset		DlouFish Dataset	
	Rank1	**Rank5**	**Rank1**	**Rank5**
W/O Quality Assessment Module	91.34	92.18	92.35	93.49
W/ Quality Assessment Module	94.83	97.64	95.81	96.61

To verify the effectiveness of the FishFace Loss, we compared it with the state-of-the-art loss function ArcFace Loss. We trained two fish individual recognition networks using different loss functions on the same network structure, and we evaluated their performance on the same test dataset. The comparison results are shown in Figure 9. We can clearly see that, under the same network structure, our designed FishFace Loss had better test accuracy and convergence speed than ArcFace Loss. This indicated that FishFace Loss can better optimize the fish feature space, thus making the distance between different categories larger and the distance within the same category smaller.

To verify the effectiveness of the step-by-step training method for the fish body recognition network proposed in this paper, we compared it with the traditional training method. Specifically, the step-by-step training method trains 90, 20, and 90 rounds for the respective three stages. The traditional training method is to train the whole model for 200 rounds. The comparison results are shown in Figure 10. The traditional training method reached a loss value of 1.15 at round 80 and then no longer decreased, while the step-by-step training method was able to reduce the loss value to 0.5. This experiment proves that the distributed training algorithm can better reduce the training loss value and improve the fish body recognition accuracy.

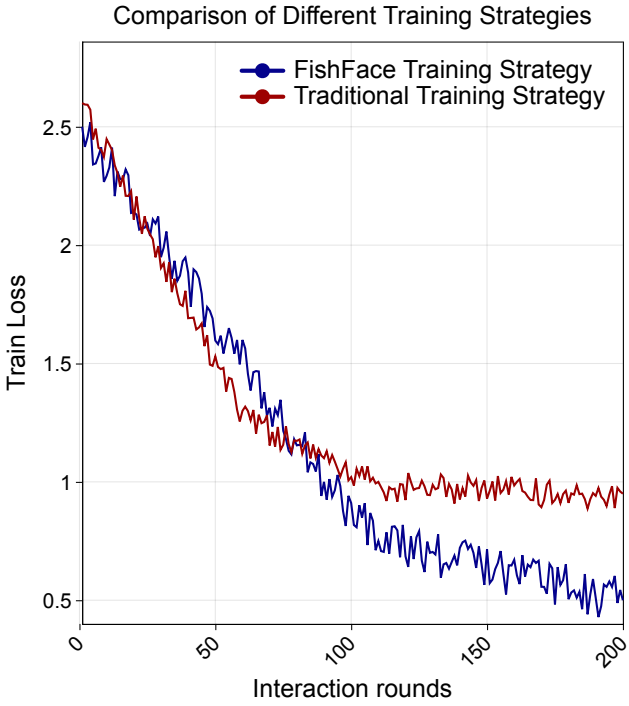

Figure 10. Loss comparison between the traditional training method and our training method.

4. Discussion

4.1. Analysis of the Experimental Results under Different Backbone Networks

To verify the performance of our backbone, we compared it with several other commonly used convolutional neural networks, including Vgg16 [29], ResNet50 [15], MobilenNet v3 [30], and SqueezeNet v2 [31]. We conducted experiments on two datasets, namely the WideFish dataset and DlouFish dataset, which are both large-scale datasets for

fish recognition. The experimental results are shown in Table 5. We can clearly see that Efficient-B5 achieved the highest respective accuracy for both Rank1 and Rank5 on both datasets, and it placed higher than the second place ResNet50 by 1.27%, 2.29%, and 3.48%, 2.29%. This showed that our improved Efficient-B5 has stronger feature extraction and generalization abilities, and that it can adapt to different fish images.

Table 5. Performance comparison of the different backbones.

Backbone	Parameter Quantity	FLOPs	WideFish Dataset		DlouFish Dataset	
			Rank1	Rank5	Rank1	Rank5
VGG16	138.1 M	15.5 G	90.51	92.69	88.41	90.24
ResNet50	25.6 M	3.8 G	93.56	95.35	92.33	94.32
MobileNet v3	2.15 M	0.22 G	91.11	93.18	90.97	92.17
SqueezeNet v2	1.24 M	0.15 G	92.33	93.66	91.23	91.68
Efficient-B5	5.3 M	0.39 G	94.83	97.64	95.81	96.61

4.2. Analysis of the Experimental Results of Different Background Environments

This section examines how the background environment affected the similarity distance of the fish individuals, which is an indicator of the model's fish recognition ability based on the similarity between two fish images. The lower the similarity distance, the more similar the fish are. We applied a deep learning-based method to remove the backgrounds from different fish images, retaining only their outline and texture. Figure 11 shows the similarity distance before and after background removal. We observed that removing the background from one image slightly changed the similarity distance of a single fish. The distance decreased slightly, thus suggesting that the background environment had a minor effect on the model's recognition ability. Removing the background from both images yielded a similar distance value as to that without background removal. This indicated that the background color had a negligible impact on the model. Our method mainly extracted texture features from a single fish, rather than learning features from the background. Therefore, we concluded that the background environment had little influence on the similarity distance of fish individuals, and that the model could effectively distinguish different fish with high accuracy.

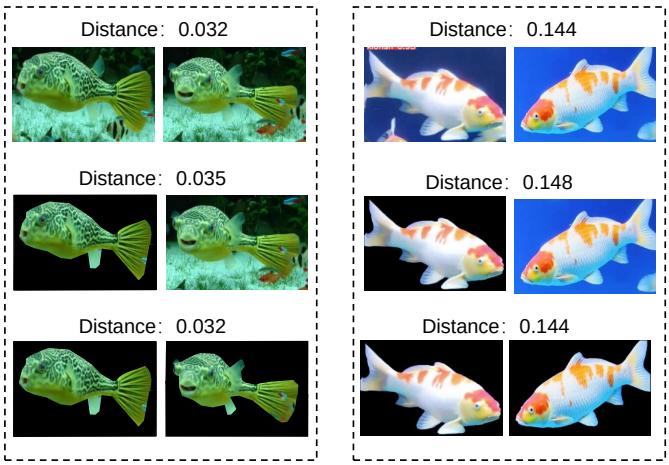

Figure 11. Experimental results of the different background environments.

5. Conclusions

We propose an underwater fish individual recognition method (FishFace) that combines data quality assessment and loss weighting. First, we designed a fish individual recognition network with a quality assessment module, which can evaluate the quality of fish images well and does not require additional labeling; second, we proposed a new loss function, FishFace Loss, which will weigh the loss according to the quality of the

image so that the model focuses more on recognizable fish images and less on images that are difficult to recognize. Finally, we collected a dataset for fish individual recognition (WideFish), which contains and annotates 5000 images of 300 fish. The experimental results show that, compared with the state-of-the-art individual recognition algorithms, Rank1 accuracy was improved by 2.60% and 3.12%, and Rank5 accuracy was improved by 2.21% and 2.71% on the public dataset DlouFish and the proposed WideFish dataset, respectively.

The main advantage of FishFace over existing algorithms is its ability to adaptively adjust to image quality factors. It can handle the problem of individual fish recognition in low-quality images, which has great significance in real life. For example, fish images in aquaculture monitoring videos are often affected by various factors such as lighting conditions, motion blur, and noise, thus making individual fish recognition more difficult. Using FishFace can improve the recognition accuracy of these low-quality images. However, FishFace still has some limitations. First, FishFace needs to calculate the image quality index, which may require additional computing costs and time. Second, FishFace's performance may be affected by the image quality evaluation method. If the image quality evaluation method is inaccurate, the calculated image quality index may also be inaccurate, thereby affecting the performance of FishFace. In addition, FishFace is only for low-quality fish individual images. If the image quality is high, it may not be the best choice.

In our future research, we plan to design a super-resolution fish individual recognition network and change the overall model training to end-to-end so that the fish individual recognition model can be more accurate and faster. In addition, we also plan to combine image segmentation techniques with individual fish recognition to identify individual fish with the background removed, thus reducing environmental interference in recognition tasks.

Author Contributions: Methodology, L.L. and J.W.; conceptualization, L.L. and J.W.; resources, H.K. and T.Z.; data curation, L.L., H.Z., B.Q. and J.W.; writing—original draft preparation, L.L. and J.W.; writing—review and editing, L.L., J.W. and H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (2021YFB2600200), the National Natural Science Foundation of China (31972846), and the Key Laboratory of Environment Controlled Aquaculture (Dalian Ocean University) Ministry of Education (202205 and 202315).

Institutional Review Board Statement: The study was approved by the Ethics Review Committee of Dalian Smart Fisheries Key Laboratory, Approval Code: 2023005, Approval Date: 16 September 2023.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We greatly appreciate the careful and precise reviews by the anonymous reviewers and editors.

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

References

1. Yang, X.; Zhang, S.; Liu, J.; Gao, Q.; Dong, S.; Zhou, C. Deep learning for smart fish farming: Applications, opportunities and challenges. *Rev. Aquac.* **2021**, *13*, 66–90. [CrossRef]
2. Soom, J.; Pattanaik, V.; Leier, M.; Tuhtan, J. A. Environmentally adaptive fish or no-fish classification for river video fish counters using high-performance desktop and embedded hardware *Ecol. Inform.* **2022**, *72*, 101817.
3. Wu, D.; Zhang, M.; Chen, H.; Bhandari, B. Freshness monitoring technology of fish products in intelligent packaging. *Crit. Rev. Food Sci. Nutr.* **2021**, *61*, 1279–1292. [CrossRef]
4. Li, D.; Wang, Z.; Wu, S.; Miao, Z.; Du, L.; Duan, Y. Automatic recognition methods of fish feeding behavior in aquaculture: A review. *Aquaculture* **2020**, *528*, 735508. [CrossRef]
5. Meng, Q.; Zhao, S.; Huang, Z.; Zhou, F. Magface: A universal representation for face recognition and quality assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14225–14234.

6. Sokolova, A.; Savchenko, A.V. Effective face recognition based on anomaly image detection and sequential analysis of neural descriptors. In Proceedings of the 2023 IX International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russia, 17–21 April 2023; pp. 1–5.
7. Wang, Q.; Du, Z.; Jiang, G.; Cui, M.; Li, D.; Liu, C.; Li, W. A Real-Time Individual Identification Method for Swimming Fish Based on Improved Yolov5. 2022. Available online: https://ssrn.com/abstract=4044575 (accessed on 15 October 2023).
8. Petrellis, N.; Keramidas, G.; Antonopoulos, C.P.; Voros, N Fish Monitoring from Low-Contrast Underwater Images. *Electronics* **2023**, *12*, 3338. [CrossRef]
9. Khan, F.F.; Li, X.; Temple, A.J. FishNet: A Large-scale Dataset and Benchmark for Fish Recognition, Detection, and Functional Trait Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 20496–20506.
10. Yin, J.; Wu, J.; Gao, C.; Jiang, Z. LIFRNet: A novel lightweight individual fish recognition method based on deformable convolution and edge feature learning. *Agriculture* **2022**, *12*, 1972. [CrossRef]
11. Gao, C.; Wu, J.; Yu, H.; Yin, J.; Guo, S. FIRN: A Novel Fish Individual Recognition Method with Accurate Detection and Attention Mechanism. *Electronics* **2022**, *11*, 3459. [CrossRef]
12. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
13. Aboah, A.; Wang, B.; Bagci, U.; Adu-Gyamfi, Y. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5349–5357.
14. Maswood, M.M.S.; Hussain, T.; Khan, M.B.; Islam, M.T.; Alharbi, A.G. CNN based detection of the severity of diabetic retinopathy from the fundus photography using efficientnet-b5. In Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 4–7 November 2020; pp. 147–150.
15. 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.
16. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
17. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
18. Huang, G.; Liu, S.; Van der Maaten, L.; Weinberger, K.Q. Condensenet: An efficient densenet using learned group convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2752–2761.
19. Lahuerta, J.J.; Paiva, B.; Vidriales. Depth of response in multiple myeloma: A pooled analysis of three PETHEMA/GEM clinical trials. *J. Clin. Oncol.* **2017**, *35*, 2900. [CrossRef] [PubMed]
20. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. *arXiv* **2016**, arXiv:1612.02295.
21. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
22. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.
23. He, X.; Zhou, Y.; Zhou, Z.; Bai, S.; Bai, X. Triplet-center loss for multi-view 3d object retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1945–1954.
24. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. Vggface2: A dataset for recognising faces across pose and age. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018; pp. 67–74.
25. Shi, Y.; Yu, X.; Sohn, K.; Chandraker, M.; Jain, A.K. Towards universal representation learning for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6817–6826.
26. Huang, G.B.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Proceedings of the Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, 17–20 October 2008.
27. He, R.; Wu, X.; Sun, Z.; Tan, T. Learning invariant deep representation for nir-vis face recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
28. Zheng, T.; Deng, W. *Cross-Pose LFW: A Database for Studying Cross-Pose Face Recognition in Unconstrained Environments*; Technical Report; Beijing University of Posts and Telecommunications: Beijing, China, 2018; Volume 5.
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

30. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
31. Gholami, A.; Kwon, K.; Wu, B.; Tai, Z.; Yue, X.; Jin, P.; Zhao, S.; Keutzer, K. Squeezenext: Hardware-aware neural network design. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1638–1647.