Journal of *Marine Science and Engineering*

MDPI

Article

An Improved Tuna-YOLO Model Based on YOLO v3 for Real-Time Tuna Detection Considering Lightweight Deployment

Yuqing Liu [1], Huiyong Chu [1], Liming Song [2,3,*], Zhonglin Zhang [1,*], Xing Wei [1], Ming Chen [1] and Jieran Shen [4]

[1] College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China
[2] College of Marine Sciences, Shanghai Ocean University, Shanghai 201306, China
[3] National Engineering Research Center for Oceanic Fisheries, Shanghai 201306, China
[4] Liancheng Overseas Fishery (Shenzhen) Co., Ltd., Shenzhen 518035, China
* Correspondence: lmsong@shou.edu.cn (L.S.); m200611361@st.shou.edu.cn (Z.Z.)

Abstract: A real-time tuna detection network on mobile devices is a common tool for accurate tuna catch statistics. However, most object detection models have multiple parameters, and normal mobile devices have difficulties in satisfying real-time detection. Based on YOLOv3, this paper proposes a Tuna-YOLO, which is a lightweight object detection network for mobile devices. Firstly, following a comparison of the performance of various lightweight backbone networks, the MobileNet v3 was used as a backbone structure to reduce the number of parameters and calculations. Secondly, the SENET module was replaced with a CBAM attention module to further improve the feature extraction ability of tuna. Then, the knowledge distillation was used to make the Tuna-YOLO detect more accurate. We created a small dataset by deframing electronic surveillance video of fishing boats and labeled the data. After data annotation on the dataset, the K-means algorithm was used to get nine better anchor boxes on the basis of label information, which was used to improve the detection precision. In addition, we compared the detection performance of the Tuna-YOLO and three versions of YOLO v5-6.1 s/m/l after image enhancement. The results show that the Tuna-YOLO reduces the parameters of YOLOv3 from 234.74 MB to 88.45 MB, increases detection precision from 93.33% to 95.83%, and increases the calculation speed from 10.12 fps to 15.23 fps. The performance of the Tuna-YOLO is better than three versions of YOLO v5-6.1 s/m/l. Tuna-YOLO provides a basis for subsequent deployment of algorithms to mobile devices and real-time catch statistics.

Keywords: tuna detection; catch statistics; lightweight network; attention mechanisms; knowledge distillation; image augmentation

Citation: Liu, Y.; Chu, H.; Song, L.; Zhang, Z.; Wei, X.; Chen, M.; Shen, J. An Improved Tuna-YOLO Model Based on YOLO v3 for Real-Time Tuna Detection Considering Lightweight Deployment. *J. Mar. Sci. Eng.* **2023**, *11*, 542. https://doi.org/10.3390/jmse11030542

Academic Editor: Rafael Morales

Received: 5 February 2023
Revised: 27 February 2023
Accepted: 1 March 2023
Published: 2 March 2023

1. Introduction

Tuna fisheries are known as "golden fisheries", and there are five regional fishery management organizations in three oceans to manage them [1–3]. Due to the depletion of several tuna stocks, stock assessment has been carried out in these regional fishery management organizations, and the resource status of important tuna stocks has been closely monitored, both of which depend on relevant fishery data and scientific observer data submitted by flag states [4–6]. It is a time-consuming and cost-ineffective task in traditional fishery management. Meanwhile, artificial intelligence technologies and deep learning algorithms are gradually replacing part of human labor. In tuna fisheries, they are gradually replacing the work of human observers. Therefore, scientists use computer vision techniques based on deep learning to classify tuna species and estimate tuna sizes to get more accurate data [7]. In addition, electronic observers will probably replace human observers in the near future.

The low detection precision usually results from a large number of species with different shapes, and complex scenarios [8,9] in tuna longline fisheries. Strachan et al. used the image binarization algorithm to differentiate the fish contour and background

by setting the size of the pixel threshold to obtain the texture features of the fish, which were transferred into the detector to classify the fish. However, the detection accuracy of this algorithm was not high [10]. Larsen et al. added and extracted the texture features of three fish species according to the method of Strachan et al. [10] and used the linear discriminant analysis (LDA) algorithm to classify 108 images of these fishes with Top1 accuracy of 76%, which was a significant improvement in the detection accuracy [11]. Wu et al. used traditional image processing methods to extract fish features, which were used for fish identification by an SVM classifier whose Top1 accuracy reached 83.33% and speed reached 5 fps [12]. Li et al. input the pretrained weights into the YOLO model to train their model. The speed was about 12 fps on the same platform, and the Top-1 accuracy reached 93% [13]. Because the accuracy of the network model and the detection speed could not be achieved ideally at the same time, Chen et al. used transfer learning to optimize the VGG16 network model, and the network model accuracy reached 97.66%, while the speed was 10.32 fps [14]. Li et al. compared the performance among GoogleNet [15], AlexNet [16], Resnet [17] and DenseNet [18] and selected the best network, DenseNet, as the detection network with an accuracy of 98.5% and speed of 1 fps [19]. Liu et al. proposed a YOLOv3 squid detection model based on MobileNet v3, and its algorithm speed reached 12 fps while its accuracy was 78.9% [20]. Wang et al. proposed a YOLO v5-L tuna detection model, whose results showed that the YOLO v5-L model had the best performance, and its mean Average Precision (mAP) reached 99.13%, while the speed was 0.82 fps [21]. Generally, 13 fps or more is the standard for real-time detection on mobile devices. It can be seen that when the detection accuracy of the network model is at a high level, the computational speed will be reduced accordingly. In addition, the detection speed reflects the efficiency of real-time detection, so how to achieve real-time detection with high accuracy becomes a challenge [22].

Generally, the YOLO series as a one-stage object detection model can basically reach a high level of accuracy and detection speed. The YOLO v3' structure is currently a relatively classic, concise and highly recognized network. While the YOLO v4 model merely provides some improvement to the YOLO v3 training tricks, YOLO v5 has better flexibility and higher speed than YOLO v4, which provides some improved ideas on the backbone and prediction head of YOLO v3. We compared the detection performance between YOLO v5 and Tuna-YOLO in the following sections. YOLO v6 [23] introduces the RepVGG structure [24] to make the network more suitable for GPU devices, which unfortunately does not meet the application scenarios of our fishing boats. YOLO v7 [25] adopts the idea of reparameterization, which provides a new idea for industrial application, but relatively complicated code and high cost have hindered the application of this network. Therefore, we choose YOLO v3 as the network to be improved at this stage.

The main structure of this article is described as follows: Section 2 introduces the source of the dataset, the setting of the experiment, and the evaluation indexes of the image enhancement algorithm and network performance. Section 3 mainly analyzes the results of the experiments. First, the effect of image enhancement is verified by using three indexes, and then the detection performance between Tuna-YOLO and other models is compared in terms of network complexity and mAP@0.5; four curves of the network before and after knowledge distillation are shown in Section 3.4: P-R curve, F1 scores curve, precision curve and recall curve. Finally, the network performance is verified by using the test dataset. Section 4 mainly discusses the advantages of Tuna-YOLO from the perspective of structural superiority and performance superiority. Section 5 summarizes the achievements and innovations of this study.

Specifically, in this study, due to the special lighting environment on longline vessels, three preprocessing algorithms were sequentially used to enhance the dataset, which improved the quality of the original image and the detection performance of the network in terms of three evaluation indexes. In order to effectively reduce the network complexity and improve the detection and classification accuracy, Tuna-YOLO was proposed based on YOLOv3. The Darknet-53 was replaced by MobileNetv3, and the CBAM attention module

was added. As the teacher network, the vanilla YOLO v3 used knowledge distillation on the backbone to guide the training of Tuna-YOLO. Through the ablation test, it is proved that the detection performance and speed can be further improved without any increase in calculation. Tuna-YOLO can provide technical support for the replacement of manual observers by electronic observers in the future.

2. Materials and Methods

2.1. Image Dataset Resource

All of the image data were from Liancheng Overseas Fishery (Shenzhen) Co., Ltd. and all the fish were put on the deck for shooting to make statistics of the catch. This study selected feature-diverse *Thunnus obesus*, *Thunnus albacares*, *Makaira mazara* and *Xiphias gladius* at a complex environment as four kinds of detection targets. Furthermore, the dataset was divided into training set, test set and validation set by the ratio of 8:1:1. The biological characteristic information of four fish species is shown in Table 1. In order to reduce the risk of data leakage, we avoided the "late split" operation when performing image augmentation to prevent false impressions that the detection performance is excellent.

Table 1. Biological characteristic information of four species.

Species	Schematic Diagram	Individuals	Biological Characteristic
Xiphias gladius		210	Brown-black back and body, small dorsal fin, no gill and pelvic fin, long and thin snout takes up one third of total length
Thunnus obesus		320	Long and thin pectoral fins, big eyes, gray belly, pectoral fins blue-black above, brown below
Thunnus albacares		200	Mid-long pectoral fins, long second dorsal fins, blue-black back, gray abdomen, other fins are yellow
Makaira mazara		150	Long body, strong front body, prominent snout like a sword, two raised crests on both sides of caudal peduncle

2.2. Experiment Set

The Tuna-YOLO was evaluated by using the above dataset. The training process made use of a warm-up strategy, learning rate decay [26], L2 regularization [27] and data preprocessing techniques [28]. The maximum rate of learning was 0.1, which was gradually decreasing. Each network will undergo 200 epochs of training. PyTorch 1.8.1 [29] was used to conduct all experiments on an NVIDIA RTX 3070 graphics card.

2.3. Evaluation Index

2.3.1. Image Enhancement

Because of the low-resolution monitoring equipment and lack of light, all videos taken from tuna vessels were in low definition and it was difficult to detect the target, which would affect the accuracy of tuna species and size detection. Distinguishing tuna is based on the fact that different tuna species have different local feature attributes. However, without clear local feature information, the recognition error rate would become higher. So, image augmentation was used to optimize the texture features of these videos [30,31]. Firstly, saturation adjustment and histogram equalization were used to improve the overall quality of the images. Then, image brightness was increased by gamma correction. Finally, the improved multi-scale Retinex algorithm was selected to improve the image quality.

2.3.2. Improved Lightweight Tuna-YOLO Network Architecture

Vanilla YOLO v3 consists of three parts, i.e., backbone, bottleneck and prediction. In the backbone, Darknet53 extracts feature information by convolution calculation, then the other two parts select a certain pixel in the image as the center point and a suitable loss function according to the prior box distribution. To make the loss value converge as quickly as possible, the size of prior boxes and the stating position of the detection frame in the network training were fine-tuned to minimize for the loss function, and to convert the detection problem to a regression question [32].

The proposed Tuna-YOLO employed MobileNet v3 as backbone. The MobileNet v3 combined the advantages of depth wise separable convolution [33], linear bottleneck in-verted residuals [34], NetAdapt algorithm [35] and SENet [36] structure. However, the SENet is not suitable for object detection because of its global characteristic. Local feature is necessary for object detection because of the complexity of scenes, e.g., different targets in similar background, same targets in different backgrounds. Therefore, the CBAM attention mechanism was used to improve the network's ability to understand local feature information to replace the SENet structure. The structure of Tuna-YOLO was shown in Figure 1.

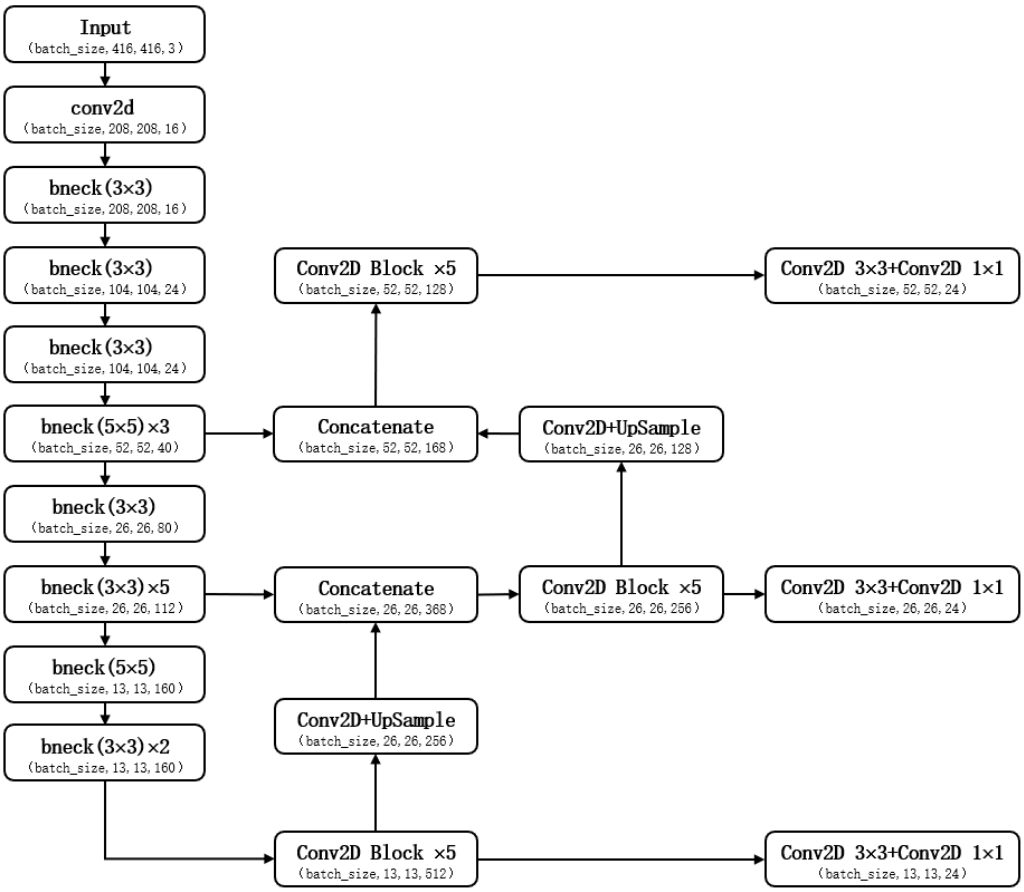
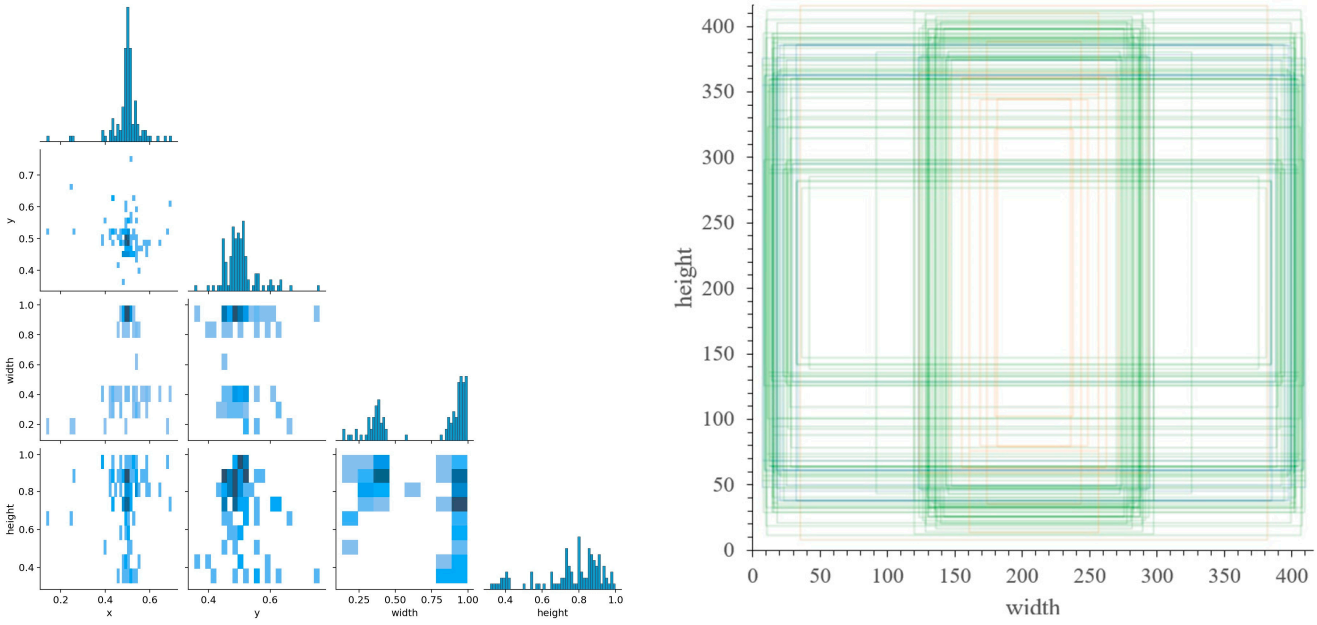


Figure 1. The structure of Tuna-YOLO. The improved MobileNet v3's 6th, 12th, and 15th layer bneck structure were used as a branch to combine with the Neck part of YOLO v3.

In the Tuna-YOLO network, the design of anchor boxes was essential for fitting degree, accuracy and real-time detection efficiency after network model training. In order to simulate the real length and width of the real bounding boxes, K-Means cluster algorithm was used to cluster 9 anchor boxes according to the label. The distribution of all ground truth bounding boxes with label information [37] was shown in Figure 2. We can find the positions of the annotation boxes basically in the center of the image, and the distribution of the annotation boxes is relatively consistent. It can be seen from the size statistics

of the annotation boxes that the targets are mainly large-sized objects, which meet the characteristics of the sample types in the dataset and are conducive to subsequent study. Analysis and research of detection algorithms improve the model precision.



(**a**) (**b**)

Figure 2. The distribution of all ground truth bounding boxes. (**a**) the length and width information of all ground truth bounding boxes and (**b**) the real size and shape of all ground truth bounding boxes.

In total, 9 sizes were obtained from clustering, e.g., (16, 23), (32, 45), (34, 26), (39, 68), (74, 48), (82, 123), (136, 98), (187, 231) and (386, 334).

2.3.3. Knowledge Distillation

The calculations and parameter amounts of the network were reduced significantly after adopting the lightweight design, but so was the detection accuracy. To address this problem, knowledge distillation (KD), a joint training method by transferring "knowledge", was employed to improve the detection accuracy. The KD structure was shown in Figure 3. KD is the process of imitating the distillation in chemistry, using the softmax function with temperature parameters to "distill" the logit output from complex and large networks, so as to generate more information in categories. This part of the in-formation is called "dark knowledge". The additional information guides the simple and small network to learn more knowledge, and the two networks are called the teacher network and the student network, respectively.

To diversify the information distribution output by the teacher network, we used the temperature parameter τ to get soft prediction output by distilling logits output between the teacher network and student network. The same dataset was used because soft prediction output implied the information of the negative samples. With the help of SoftMax active function, the teacher network's class prediction probability distribution could be regarded as the soft target. Similarly, this method was used to get not only the soft prediction output but also the hard prediction output from the student network. As for the soft prediction output, soft prediction output and soft target were used to calculate loss value by loss function L_{soft}, which was a part of total loss. The L_{soft} was defined as:

$$L_{soft} = -\sum_{i}^{N} P_i^T \log\left(Q_i^T\right) \quad (1)$$

where P_i^T is the i-th soft target at time T, Q_i^T is the i-th soft prediction output at time T, N is the total number of samples and $N = 27$ in this paper.

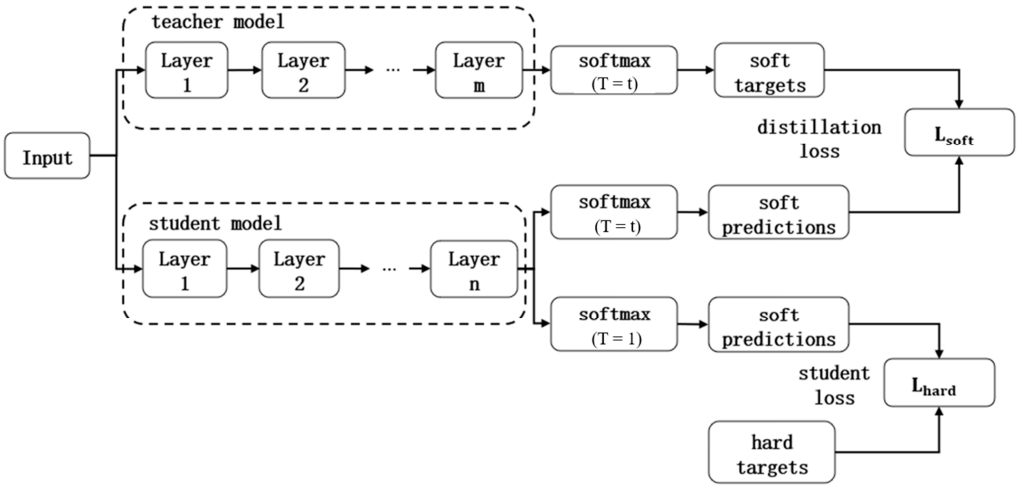


Figure 3. The KD structure. Firstly, a "teacher" network whose network depth and width were much larger than MobileNet v3 was grafted to the original YOLO v3 structure, and trained to reach a good performance. Then, a relatively simple student network, MobileNet v3, was built and then trained by the "dark knowledge" of the superior teacher network, so that the detection performance of the student network was close to that of the teacher network, which was another kind of knowledge transfer.

The hard prediction output and ground truth were used to calculate the loss value by loss function L_{hard}. The L_{hard} was defined as:

$$L_{hard} = -\sum_{i}^{N} C_i^T \log(Q_i^T) \tag{2}$$

where C_i^T is the i-th hard target at time T, $N = 27$ in this paper.

The total loss function was defined as:

$$L_{all} = L_{soft} + L_{hard} \tag{3}$$

In this paper, DenseNet201-YOLOv3 and backbone of improved Tuna-YOLO were selected, respectively, for the teacher network and the student network, as a way to improve the detection performance and to increase the mAP of Tuna-YOLO.

2.4. Methods

To test the enhancement results of different augmentation algorithms mentioned in Section 2.3.1, the images before and after augmentation were compared according to the combination and splitting of algorithms, and three indexes to evaluate the quality of images were used [38], i.e., standard deviation, mean gradient and information entropy.

To evaluate the network computation speed, the index of frames per second (fps) was compared between Tuna-YOLO and other lightweight networks, such as DarkNet53, GhostNet, SqueezeNet, ShuffleNetv1, ShuffleNetv2, MobileNetv1, MobileNetv2, MobileNetv3 and MobileNetv3-ECA [39,40].

In addition, the network performance and computation speed were synthetically compared between the Tuna-YOLO after knowledge distillation and other models, such as DenseNet121-YOLOv3, DenseNet169-YOLOv3 and DenseNet201-YOLOv3. In particular, the speed of these models was evaluated in terms of parameters, floating-point operations per second (FLOPs) and fps, and the detection performance was evaluated in terms of accuracy, recall rate and mAP [41]. The closer the mAP value is to 1, the better the predictive

performance of the network model. Generally, these three indexes can evaluate the detection performance of the network to varying degrees. The mAP reflects the detection accuracy on the basis of IoU, so it is the most important evaluation index. The performance of class prediction can be directly explored from the confusion matrix. The types of prediction mainly include the following four types: True Positive, False Negative, False Positive, and True Negative, which mainly reflect the relationship between the predicted class and the real class, which can be seen in Table 2 for the description.

Table 2. The distribution of classification results.

Confusion Matrix		Predict Label	
		Positive	**Negative**
Real label	positive	X_{TP}	X_{FN}
	negative	X_{FP}	X_{TN}

Accuracy represents the rate of predicting positively in prediction results, which is defined as:

$$P = \frac{X_{TP}}{X_{TP} + X_{FP}} \tag{4}$$

Recall rate represents the rate of predicting positively in all samples, which was defined as:

$$R = \frac{X_P}{X_{TP} + X_{FN}} \tag{5}$$

where X_{TP} represents the number of positive samples that are correctly divided, X_{FP} represents the number of samples that are incorrectly classified as positive samples, X_{FN} represents the number of wrongly classified as negative samples and X_{TN} represents the number of negative samples that are correctly divided.

The equation of *mAP* was defined as:

$$mAP = \frac{1}{m} \sum_{\gamma \in \{0, 0.1, \ldots, 1\}} \max_{R \geq \gamma} P(R) \tag{6}$$

$$IoU = \frac{P \cap R}{P \cup R} \tag{7}$$

where γ is the threshold of *IoU*, m is the number of different samples; γ = 0.5, m = 4 in this paper.

3. Results

3.1. Comparison of Different Image Augmentations

Figure 4 shows the images before and after augmentation. Table 3 shows their respective values of standard deviation, mean gradient and information entropy. The improved Retinex algorithm achieved the best results on the three-evaluation index (Table 3).

Table 3. Comparison of image quality between before and after image augmentation.

Evaluation Index	Original Image	Saturation Adjustment	Histogram Equalization	Gamma Correction	Improved Multi-Scale Retinex
Standard deviation	24.632	45.373	49.231	46.234	**51.289**
Mean gradient	0.213	0.479	0.628	0.0417	**0.642**
Information entropy	3.583	6.597	6.821	5.124	**6.924**

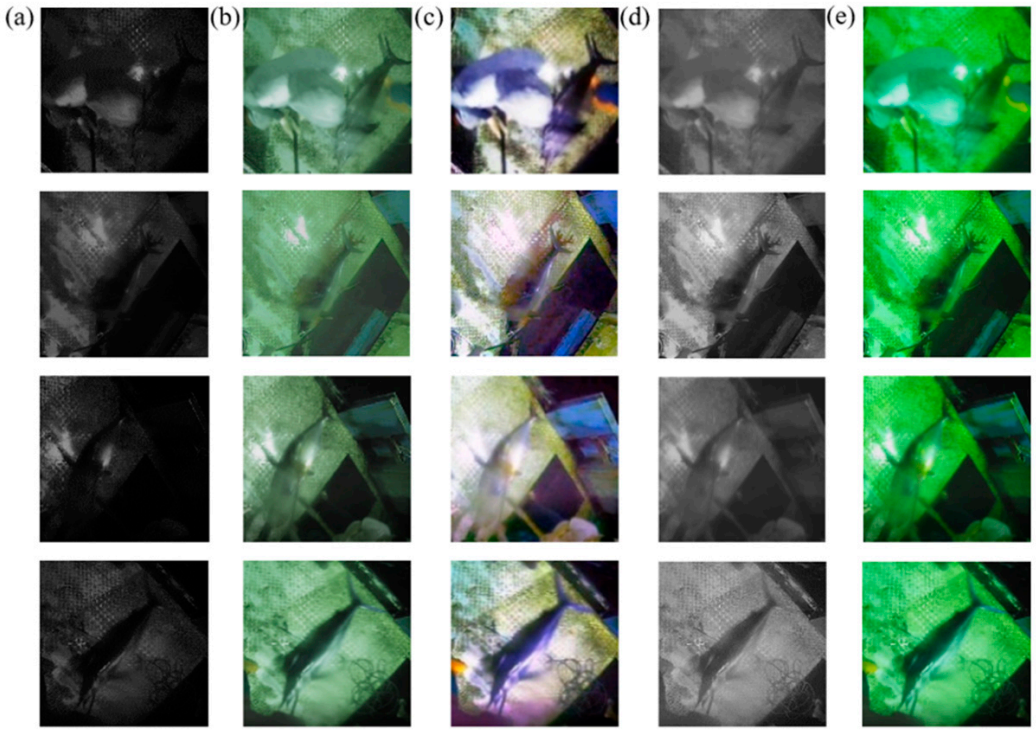


Figure 4. Effect of different data augmentation. (**a**–**e**), respectively, represent original image, saturation adjustment, histogram equalization, gamma correction and improved multi-scale retinex.

In order to verify the superiority of the improved multi-scale Retinex image augmentation algorithm in network model performance, a comparison of the mAP was conducted among Tuna-YOLO, YOLOv3 and DenseNet201 after training. The best mAP was based on the improved multi-scale Retinex algorithm (Table 4).

Table 4. mAP of network model in different augmentation algorithms.

Image Augmentation	YOLOv3	DenseNet201	Tuna-YOLO
Original image	26.85	35.12	23.68
Saturation adjustment	52.79	65.37	48.64
Histogram equalization	63.74	78.59	57.32
Gamma correction	54.44	65.91	52.87
Improved multi-scale Retinex	**79.63**	**94.94**	**78.21**

3.2. Comparison of Detection between Tuna-YOLO and Other Lightweight Network

To verify the detection performance of Tuna-YOLO after knowledge distillation, the same dataset from Liancheng Overseas Fishery (Shenzhen, China) Co., Ltd. was used, and the videos were framed into annotated images, which were input into all lightweight networks. All lightweight networks were deployed on the baseline of YOLOv3 for experiments. The results are shown in Table 5.

Compared with the YOLOv3 based on DarkNet53, the improved Tuna-YOLO in this study reduced Params by 62.3%, FLOPs by 73.5% and mAP by 1.8%, and increased fps by 50.5%. Compared with other lightweight networks, Tuna-YOLO had obvious advantages in terms of fps and mAP. The improvement of fps facilitates the real-time detection of tuna on mobile devices.

Table 5. Performance comparison of different lightweight networks.

Network	Params/MB	FLOPs/G	fps	mAP@0.5/%
DarkNet53	234.74	32.767	10.12	**79.63**
GhostNet	87.33	**8.409**	11.81	64.60
SqueezeNet	97.93	9.861	11.23	65.97
ShuffleNetv1	83.25	8.636	15.12	68.56
ShuffleNetv2	84.36	8.565	14.54	71.67
MobileNetv1	92.15	10.146	12.14	68.54
MobileNetv2	84.94	8.952	14.48	71.23
MobileNetv3	88.48	8.676	15.13	73.59
MobileNetv3-ECA	**82.71**	8.674	14.98	78.17
Tuna-YOLO	88.45	8.676	**15.23**	78.21

3.3. Comparison of Performance between Tuna-YOLOs after Knowledge Distillation and Other YOLOv3s and YOLOv5s

Compared with the original YOLOv3, the knowledge-distilled Tuna-YOLO improved mAP by 7.67%. In general, YOLO v5 is more suitable for small object detection, and the detection performance improves with the increase of network parameters. The YOLOv5-6.1 large's mAP will be higher than that of the knowledge-distilled Tuna-YOLO when using the maximum number of parameters, but it still fails to meet our requirements for detection speed (Table 6). The comparison curves consisting of PR curve, F1 score curve, precision curve and recall curve, are shown in Figure 5.

Table 6. Comparison of model performance.

Model	Params/MB	FLOPs/G	fps	mAP@0.5/%
YOLOv3	234.74	32.767	10.12	79.63
DenseNet121-YOLOv3	106.44	18.033	6.49	91.37
DenseNet169-YOLOv3	128.92	19.945	5.38	92.12
DenseNet201-YOLOv3	151.06	23.263	4.96	**94.94**
YOLO v5-6.1 large	92.34	16.64	7.52	92.84
YOLO v5-6.1 mid	69.26	8.34	16.98	83.26
YOLO v5-6.1 small	**39.24**	**4.32**	**27.39**	64.35
Tuna-YOLO	88.45	8.676	15.23	78.21
Tuna-YOLO after kd	88.45	8.676	15.23	85.74

The model performance of Tuna-YOLO after knowledge distillation has been significantly improved compared with the original YOLOv3 (Figure 5).

3.4. Validation Results of the Network Model

The Tuna-YOLO after knowledge distillation and original YOLOv3 were used to detect the target from the electronic monitoring videos in frames. The detection precision of various tuna species is shown in Table 7, and the comparison of detection results is shown in Figure 6. The comparison of the confusion matrix is shown in Figure 7. The precision of Tuna-YOLO was higher than that of the original YOLOv3 (Table 7).

Table 7. Precision values of tunas on Tuna-YOLO and original YOLOv3.

Model	*Xiphias gladius*	*Thunnus obesus*	*Thunnus albacares*	*Makaira mazara*
YOLOv3	89%	84%	87%	83%
Tuna-YOLO	**97%**	**95%**	**97%**	**98%**

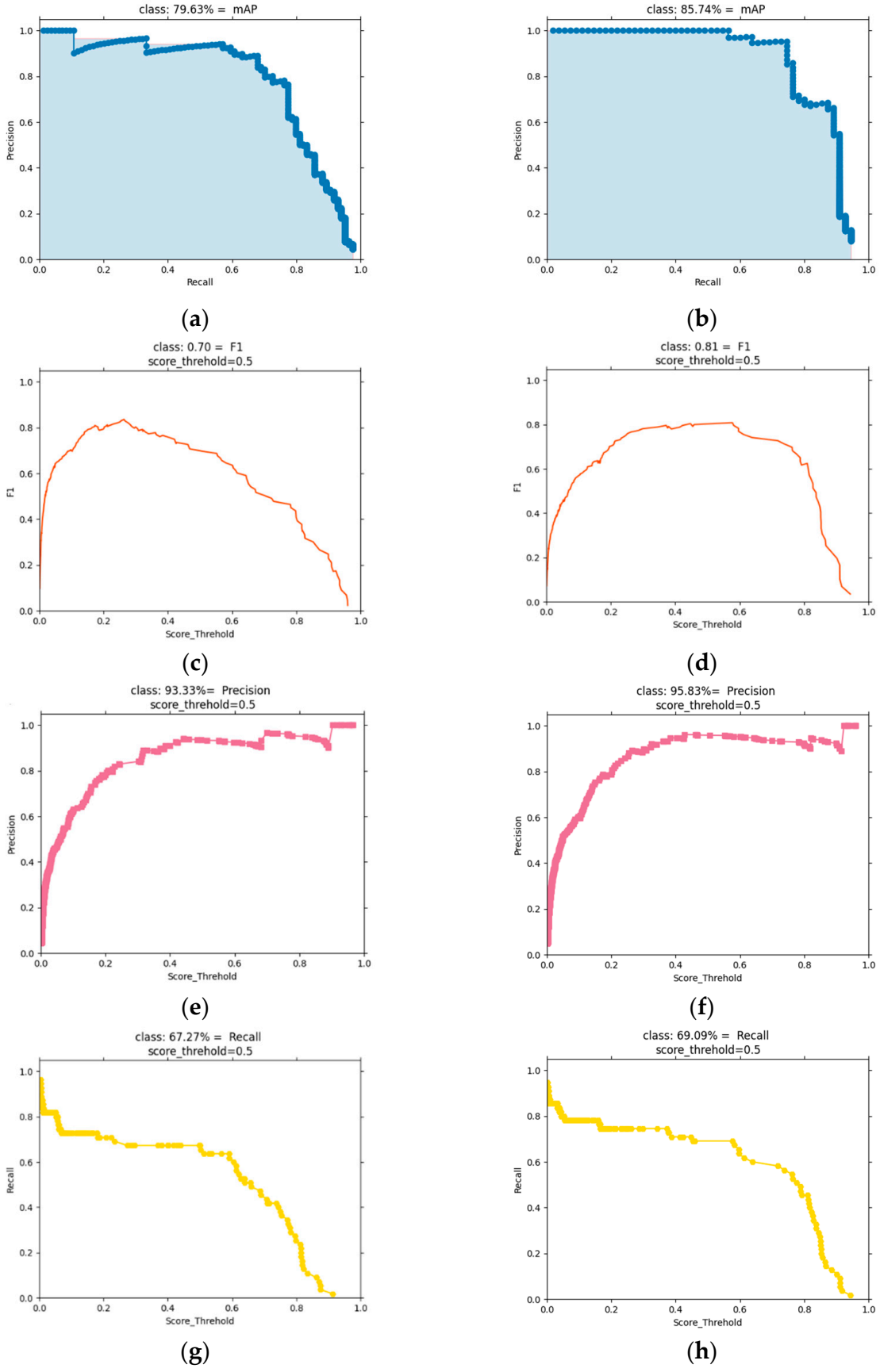


Figure 5. Comparison of network model training results. (**a**–**h**), respectively, represent PR curve, F1 score curve, precision curve and recall curve.

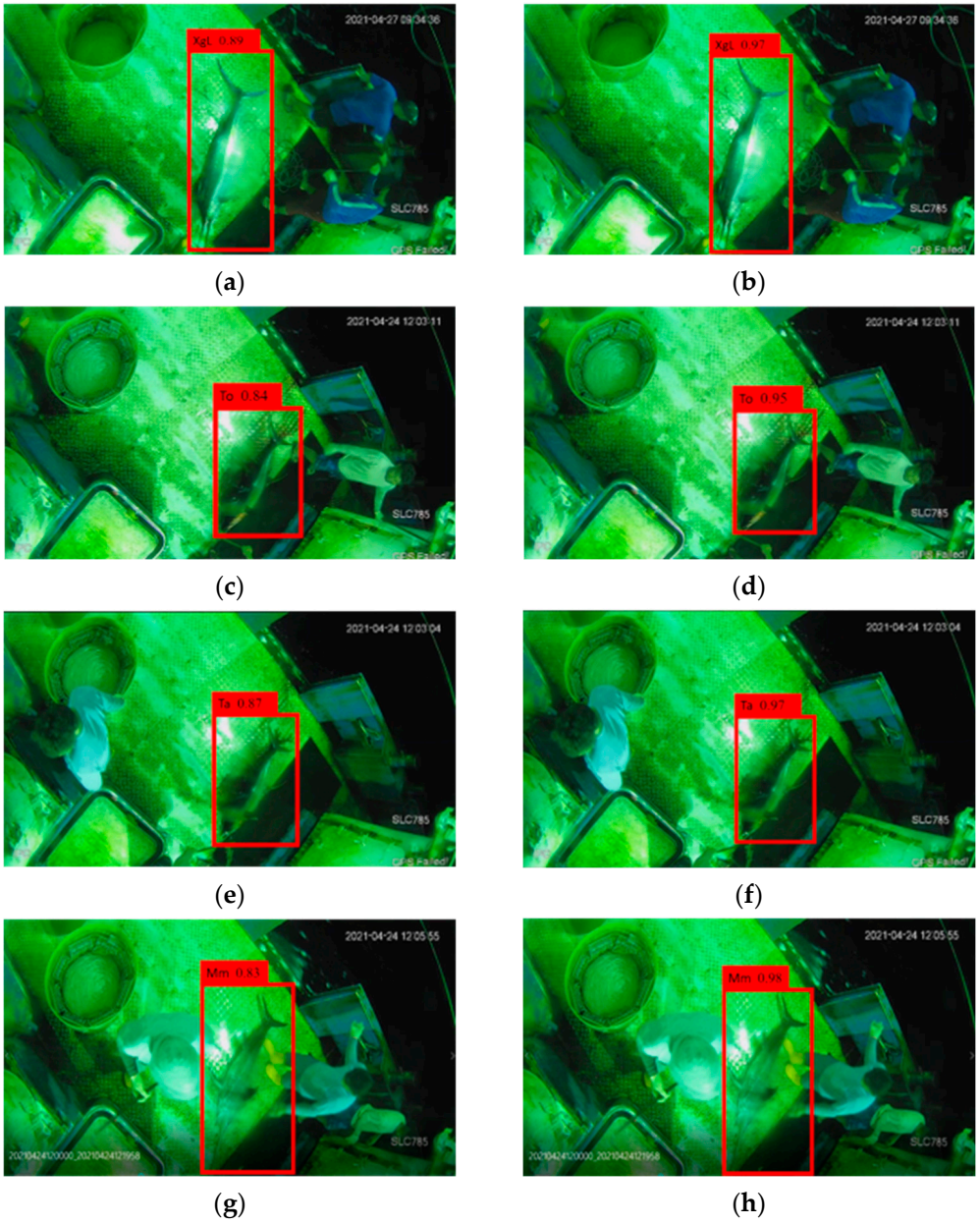


Figure 6. Comparison of test results. (**a**,**c**,**e**,**g**), respectively, represent detection result images by original YOLOv3, and (**b**,**d**,**f**,**h**) are the detection results by Tuna-YOLO.

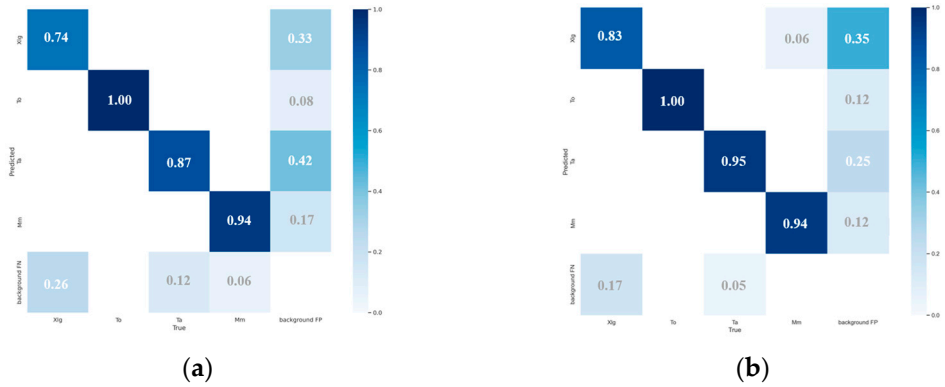


Figure 7. The confusion matrices of YOLO v3 and Tuna-YOLO. (**a**,**b**), respectively, represent the confusion matrix of original YOLO v3 and Tuna-YOLO.

4. Discussion

4.1. The Advantages of Improved Tuna-YOLO

The improved Tuna-YOLO based on YOLOv3 is suitable for tuna detection because the YOLOv3 performs better than Faster-RCNN and SSD in terms of speed and accuracy [42–44]. On the basis of YOLOv3, the Tuna-YOLO has higher detection accuracy and simpler network structure [45]. Jiang et al. [46] and Wang et al. [47] also optimized the original network on the aspect of detection accuracy, but it was practically difficult to deploy on mobile devices because of the many parameters. Jiang et al. [48] integrated the ideas of dense connections, residual connections and group convolution. The mAP indicators on the mini-RD and SAR ship detection dataset (SSDD) reached 83.21% and 85.46%, respectively. Furthermore, compared with different YOLO v5 versions, Tuna-YOLO after knowledge distillation is also superior in comprehensive performance. Tuna-YOLO borrowed the idea of YOLOv3 and replaced the backbone DarkNet53 of YOLOv3 with the MobileNetv3 with CBAM attention module, which reduced the parameter amount of network by 62.3%. Given that the parameter decrease would inevitably lead to a decrease in mAP, knowledge distillation was used to operate knowledge transfer from the teacher network to the student network. The detection precision of knowledge distillation was improved by 6.41%, which perfectly solved the problem of low detection accuracy with reduced model parameters, hence the realization of real-time detection.

4.2. The Comparison of Performance between Tuna-YOLO and Other Models

In this study, the mAP of Tuna-YOLO reached 85.74%, the fps reached 15.23 fps and the accuracy reached 95.83%, which were at a relatively high level. Alessandro Betti [49] presented YOLO-S, whose architecture exploited a small feature extractor based on Darknet20, as well as skip connection, via both bypass and concatenation, and reshape-passthrough layer to avoid the vanishing gradient problem, and promoted feature reuse across the network and combined low-level positional information with more meaningful high-level information. Muksit et al. [50] proposed the YOLO-Fish, which enhanced YOLOv3 by fixing the issue of up sampling step sizes to reduce the misdetection of tiny fish and adding spatial pyramid pooling (SPP) to the first model to add the capability to detect fish appearance in those dynamic environments, respectively. Kazim et al. [51] put forward the improved YOLOv3 by increasing detection scale from three to four, applied K-means clustering to optimize the anchor boxes, novel transfer learning technique and improved loss function to increase the model performance. Gupta et al. [52] raised the YOLO Fish, which used hierarchical techniques in both the classification step and in the dataset, with a mAP of 91.8%. However, the speed was only 3.79 fps. Wang et al. [53] proposed the FML-Centernet model to detect fish in a river. This network improved the efficiency of detection by testing the ratio of positive and negative samples and optimizing the loss function. The mAP of the network reached 85%, and the fps was 10.12 fps. Li et al. [54] proposed an improved fish recognition network model YOLO-V3-Tiny-MobileNet by optimizing the MobileNet and YOLO-V3-Tiny network models, which had shallow feature extraction network layers and insufficient extraction capabilities. The recognition precision and accuracy of the model were 79.3% and 86.5%, respectively. Xu et al. [55] proposed a detection network model (YOLOv3-Corn) for corn leaf diseases and insect pests. By modifying the feature fusion layers of the network model, a new Head (104 $\times$ 104) was constructed to improve the detection accuracy; the detection accuracy of the network model YOLOv3-Corn was 84.34%, and the fps was 8.7 fps. Table 8 shows the specific results of comparison.

Table 8. Comparison with different algorithms based on the YOLO.

Model	mAP@0.5/%	fps
YOLO-S [49]	46.7	8.1
YOLO-Fish [50]	76.56	7.6
Improved YOLO v3 [51]	91.3	5.9
YOLO Fish [52]	91.8	3.79
FML- Centernet [53]	85	10.12
YOLO-V3-Tiny-MobileNet [54]	86.5	9.7
YOLOv3-Corn [55]	84.34	8.7
Tuna-YOLO	85.74	15.23

5. Conclusions

An improved real-time lightweight detection network was proposed for tuna detection based on the YOLOv3 network, which used lightweight design on the backbone and combined the CBAM attention mechanism module on the basis of the MobileNet v3 network structure to build an efficient tuna detection network, Tuna-YOLO. Knowledge distillation was used on the Tuna-YOLO to improve the accuracy of the model. The experimental results showed that the Tuna-YOLO was more streamlined after model compression, which realized the real-time detection of tuna on the mobile devices by increasing the detection speed and provided potential for the replacement of human observers with electronic observers.

Author Contributions: Conceptualization, Y.L. and H.C.; Methodology, H.C.; Software, Z.Z.; Validation, Y.L. and H.C.; Formal Analysis, H.C.; Investigation, L.S.; Resources, J.S.; Data Curation, L.S.; Writing—Original Draft Preparation, H.C.; Writing—Review and Editing, Y.L. and L.S.; Visualization, X.W. and M.C.; Supervision, L.S.; Project Administration, Y.L.; Funding acquisition, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by [National Natural Science Foundation of China] grant number [32273185], [National Key Research and Development Program of China] grant number [2020YFD0901205], [Marine Fishery Resources Investigation and Detection Project of the Ministry of Agriculture and Rural Affairs of China] grant number [D-8006-21-0215], and [R & D Program of CNFC Overseas Fishery Co., Ltd.] grant number [D-8006-20-0180].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their gratitude for the support of the Fishery Engineering and Equipment Innovation Team of Shanghai High-level Local University. The authors also wish to thank Huihui Shen, School of Foreign Languages, Shanghai Ocean University for suggesting improvements in the language.

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

Abbreviations

mAP	mean Average Precision
IoU	Intersection over Union
Params	The number of parameters
FLOPs	Floating Point Operations
fps	Frame Per second
L_{soft}	Loss function based on soft label
L_{hard}	Loss function based on hard label
L_{all}	Total loss function

X_{TP}	the number of positive samples
X_{FN}	the number of wrongly classified as negative samples
X_{FP}	the number of samples that are incorrectly classified as positive samples
X_{TN}	the number of negative samples
P	Precision
R	Recall

References

1. Liam, C. The Tuna 'Commodity Frontier': Business Strategies and Environment in the Industrial Tuna Fisheries of the Western Indian Ocean. *J. Agrar. Chang.* **2012**, *12*, 252–278. [CrossRef]
2. Li, J.; Dai, X. Management strategy for the south Pacific albacore (Thunnus alalunga) in the western and central Pacific Ocean and the countermeasure of China. *J. Shanghai Ocean Univ.* **2022**, *31*, 1190–1198. [CrossRef]
3. Sun, J.; Li, Y.; Xu, L. Resource management and sustainable utilization of southern bluefin tuna. *J. Shanghai Ocean Univ.* **2016**, *25*, 936–944. [CrossRef]
4. Ovando, D.; Libecap, G.D.; Millage, K.D.; Thomas, L. Coasean Approaches to Ending Overfishing: Bigeye Tuna Conservation in the Western and Central Pacific Ocean. *Mar. Resour. Econ.* **2021**, *36*, 1, 91–109. [CrossRef]
5. O'Shea, E.J.A. Changes in habitat preference of tuna species and implication for regional fisheries management: Southern bluefin tuna fishing in the Indian Ocean. *Aust. J. Marit. Ocean Aff.* **2016**, *8*, 117–131. [CrossRef]
6. Zhang, H.; Yang, S.; Fan, W.; Shi, H.; Yuan, S.L. Spatial Analysis of the Fishing Behaviour of Tuna Purse Seiners in the Western and Central Pacific Based on Vessel Trajectory Data. *J. Mar. Sci. Eng.* **2021**, *9*, 322. [CrossRef]
7. Gong, P.; Wang, D.; Yuan, H.; Chen, G.; Wu, R. Fishing Ground Forecast Model of Albacore Tuna Based on LightGBM in the South Pacific Ocean. *Fish. Sci.* **2021**, *40*, 762–767. [CrossRef]
8. Zhang, L.; Zhang, Y.; Zhang, Z.; Shen, J.; Wang, H. Real-Time Water Surface Object Detection Based on Improved Faster R-CNN. *Sensors* **2019**, *19*, 3523. [CrossRef]
9. Hou, Q.; Zhou, C.; Wan, R.; Zhang, J.; Xue, F. Application of Feature Point Matching Technology to Identify Images of Free-Swimming Tuna Schools in a Purse Seine Fishery. *J. Mar. Sci. Eng.* **2021**, *9*, 1357. [CrossRef]
10. Strachan, N.J.C.; Nesvadba, P.; Allen, A.R. Fish species recognition by shape analysis of images. *Pattern Recognit.* **1990**, *23*, 539–544. [CrossRef]
11. Larsen, R.; Olafsdottir, H.; Ersbøll, B.K. Shape and texture based classification of fish species. In *Scandinavian Conference on Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2009.
12. Wu, Y.; Yin, J.; Dai, Y.; Yuan, Y. Identification method of freshwater fish species using multi-kernel support vector machine with bee colony optimization. *Trans. Chin. Soc. Agric. Eng.* **2014**, *30*, 312–319. [CrossRef]
13. Li, Q.; Li, Y.; Niu, J. Real-time detection of underwater fish based on improved YOLO and transfer learning. *Pattern Recognit. Artif. Intell.* **2019**, *32*, 193–203. [CrossRef]
14. Chen, Y.; Gong, C.; Liu, Y. Fish identification method based on FTVGG16 convolutional neural network. *Trans. Chin. Soc. Agric. Mach.* **2019**, *50*, 223–231. [CrossRef]
15. Szegedy, C.; Wei, L.; Yangqing, J.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
17. 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.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 21–26 July 2017; pp. 4700–4708.
19. Li, S.; Yang, L.; Yu, H.; Chen, Y. Underwater Fish Species Identification Model and Real-Time Identification System. *Smart Agric.* **2022**, *4*, 130–139. [CrossRef]
20. Liu, Y.; Zhou, Y.; Huang, L.; Sui, J. Application of lightweight neural network in detection technology of pelagic squid fishing. *Fish. Mod.* **2022**, *49*, 61–71. [CrossRef]
21. Wang, S.; Zhang, S.; Zhu, W.; Sun, Y.; Yang, Y.; Sui, J.; Shen, L.; Shen, J. Application of an electronic monitoring system for video target detection in tuna longline fishing based on YOLOV5 deep learning model. *J. Dalian Ocean Univ.* **2021**, *36*, 842–850. [CrossRef]
22. Li, C.; Yao, J.; Lin, Z.; Yan, Q.; Fan, B. Object detection method based on improved YOLO lightweight network. *Laser Optoelectron. Prog.* **2020**, *57*, 45–53. [CrossRef]
23. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
24. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13733–13742.

25. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
26. Bello, I.; Zoph, B.; Vasudevan, V.; Le, Q.V. Neural Optimizer Search with Reinforcement Learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; JMLR, Inc.: Sydney, Australia, 2017; Volume 70, pp. 459–468.
27. Moradi, R.; Berangi, R.; Minaei, B. A survey of regularization strategies for deep models. *Artif. Intell. Rev.* **2020**, *53*, 3947–3986. [CrossRef]
28. Salamon, J.; Bello, J.P. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [CrossRef]
29. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. *arXiv* **2019**, arXiv:1912.01703.
30. Zhang, M.; Long, T.; Song, W.; Huang, D.; Mei, H.; He, Q. Object Detection of Underwater Fish at Night Based on Improved Cascade R-CNN and Image Enhancement. *Trans. Chin. Soc. Agric. Mach.* **2021**, *52*, 179–185. [CrossRef]
31. Li, B.; Tang, G.; Cui, Z. A Weakly Illuminated Image Enhancement Algorithm Based on Retinex Model. *Comput. Technol. Dev.* **2021**, *30*, 79–84. [CrossRef]
32. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
33. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
34. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv* **2018**, arXiv:1801.04381.
35. Yang, T.J.; Howard, A.; Chen, B.; Zhang, X.; Go, A.; Sandler, M.; Sze, V.; Adam, H. NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications. *arXiv* **2018**, arXiv:1804.03230.
36. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. IEEE Trans. *Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
37. Zheng, Z.; Li, Y.; Lu, P.; Zou, G.; Wang, Z. Application research of improved YOLO v4 model in fish object detection. *Fish. Mod.* **2022**, *49*, 82–88. [CrossRef]
38. Crowley, E.J.; Gray, G.; Storkey, A.J. Moonshine: Distilling with cheap convolutions. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 2888–2898.
39. Khasawneh, N.; Fraiwan, M.; Fraiwan, L. Detection of K-complexes in EEG signals using deep transfer learning and YOLOv3. *Cluster Comput.* **2022**, *1*, 1–11. [CrossRef]
40. Fraiwan, M.; Audat, Z.; Fraiwan, L. Using deep transfer learning to detect scoliosis and spondylolisthesis from X-ray images. *PLoS ONE* **2022**, *17*, e0267851. [CrossRef]
41. Jiang, X.; Yao, H.; Liu, D. Nighttime image enhancement based on image decomposition. *Signal Image Video Process.* **2019**, *13*, 189–197. [CrossRef]
42. He, C.; Li, D.; Wang, S. A lightweight convolutional neural network model for target recognition. *J. Phys. Conf. Ser.* **2020**, *1651*, 012138. [CrossRef]
43. Ge, H.; Dai, Y.; Zhu, Z.; Liu, R. A Deep Learning Model Applied to Optical Image Target Detection and Recognition for the Identification of Underwater Biostructures. *Machines* **2022**, *10*, 809. [CrossRef]
44. Kong, L.; Wang, J.; Zhao, P. YOLO-G: A Lightweight Network Model for Improving the Performance of Military Targets Detection. *IEEE Access* **2022**, *10*, 55546–55564. [CrossRef]
45. Cao, J.; Ren, W.; Zhang, H.; Chen, Z. Candidate box fusion based approach to adjust position of the candidate box for object detection. *IET Image Process.* **2021**, *15*, 2799–2809. [CrossRef]
46. Jiang, Z.; Zhao, L.; Li, S.; Jia, Y. Real-time object detection method based on improved YOLOv4-tiny. *arXiv* **2020**, arXiv:2011.04244.
47. Wang, L.; Duan, J.; Xin, L. YOLOv5 Helmet Wear Detection Method with Introduction of Attention Mechanism. *Comput. Eng. Appl.* **2022**, *58*, 303–312. [CrossRef]
48. Jiang, S.; Zhou, X. DWSC-YOLO: A Lightweight Ship Detector of SAR Images Based on Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 1699. [CrossRef]
49. Betti, A. A lightweight and accurate YOLO-like network for small target detection in Aerial Imagery. *arXiv* **2022**, arXiv:2204.02325.
50. Al Muksit, A.; Hasan, F.; Emon, M.F.H.B.; Haque, M.R.; Anwary, A.R.; Shatabda, S. YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecol. Inform.* **2022**, *72*, 101847. [CrossRef]
51. Raza, K.; Song, H. Fast and Accurate Fish Detection Design with Improved YOLO-v3 Model and Transfer Learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 7–16. [CrossRef]
52. Kalhagen, E.S.; Olsen, Ø.L.; Goodwin, M.; Gupta, A. Hierarchical Object Detection applied to Fish Species. *Nord. Mach. Intell.* **2022**, *1*, 1–15. [CrossRef]
53. Liu, Y.; Wang, Y.; Huang, L. Fish Recognition and Detection Method Based on FML-Centernet Algorithm. *Laser Optoelectron. Prog.* **2022**, *59*, 317–324. [CrossRef]

54. Li, J.; Zhu, K.; Yang, S. Ocean fish recognition in complex scene based on transfer learning. *Comput. Appl. Softw.* **2019**, *36*, 168–174. [CrossRef]
55. Xu, H.; Huang, Y.; Liu, M. Research on pest detection and identification of corn leaf based on improved YOLOv3 model. *J. Nanjing Agric. Univ.* **2022**, *45*, 1276–1285. [CrossRef]