

Article

An Accurate Detection Model of *Takifugu rubripes* Using an Improved YOLO-V7 Network

Siyi Zhou ¹, Kewei Cai ², Yanhong Feng ^{1,*}, Xiaomeng Tang ¹, Hongshuai Pang ¹, Jiaqi He ¹ and Xiang Shi ¹

¹ College of Information and Engineering, Dalian Ocean University, Dalian 116000, China

² College of Mechanical and Electrical Engineering, Dalian Minzu University, Dalian 116600, China

* Correspondence: fengyh@dlo.u.edu.cn

Abstract: In aquaculture, the accurate recognition of fish underwater has outstanding academic value and economic benefits for scientifically guiding aquaculture production, which assists in the analysis of aquaculture programs and studies of fish behavior. However, the underwater environment is complex and affected by lighting, water quality, and the mutual obscuration of fish bodies. Therefore, underwater fish images are not very clear, which restricts the recognition accuracy of underwater targets. This paper proposes an improved YOLO-V7 model for the identification of *Takifugu rubripes*. Its specific implementation methods are as follows: (1) The feature extraction capability of the original network is improved by adding a sizeable convolutional kernel model into the backbone network. (2) Through ameliorating the original detection head, the information flow forms a cascade effect to effectively solve the multi-scale problems and inadequate information extraction of small targets. (3) Finally, this paper appropriately prunes the network to reduce the total computation of the model; meanwhile, it ensures the precision of the detection. The experimental results show that the detection accuracy of the improved YOLO-V7 model is better than that of the original. The average precision improved from 87.79% to 92.86% (when the intersection over union was 0.5), with an increase of 5.07%. Additionally, the amount of computation was reduced by approximately 35%. This shows that the detection precision of the proposed network model was higher than that for the original model, which can provide a reference for the intelligent aquaculture of fishes.



Citation: Zhou, S.; Cai, K.; Feng, Y.; Tang, X.; Pang, H.; He, J.; Shi, X. An Accurate Detection Model of *Takifugu rubripes* Using an Improved YOLO-V7 Network. *J. Mar. Sci. Eng.* **2023**, *11*, 1051. <https://doi.org/10.3390/jmse11051051>

Academic Editor: Ernesto Weil

Received: 13 April 2023

Revised: 9 May 2023

Accepted: 10 May 2023

Published: 15 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: *Takifugu rubripes*; accurate identification; improved YOLO-V7 network; large convolution kernel

1. Introduction

Triggerfish, commonly known as “puffer fish” [1], are a kind of bony fish with a high economic value and are an important aquaculture group in northeast Asia. The usual species are *Takifugu rubripes* (*T. rubripes*), *Takifugu obscurus*, and *Takifugu pseudonmmus*, etc. The mariculture species are mainly *T. rubripes*, and the production mode of combining sea cage and land-based industrialization is adopted. *Takifugu rubripes* is delicious and nutritious and, as a high-quality food, it is in great demand for exports. With the development of digitalization and informatization, the traditional aquaculture fishery management model relying on human resources and experience now encounters limitations [2]. Problems of manual observation and purely empirical methods such as misdetection, missing detection, and untimely feedback happen occasionally. At present, the fish movement posture in water is variable. The underwater environment, lights, and the mutual occlusion between fish bodies reduce the accuracy of static fish identification in breeding ponds, which leads to the problem of a low accuracy of fish identification [3–5]. Traditional methods can no longer meet the needs of precision and intelligence for modern aquaculture [6]. The accurate identification of fish requires more and more attention and the automatic detection and identification of fish underwater are significant for fishery resource assessment and ecological environment monitoring [7–9]. Therefore, this paper focuses on the accurate identification algorithms for fish in underwater images to solve these above problems.

Traditional target detection algorithms typically use a sliding window model, in which a window sequentially slides on the detected image [10]. Feature extraction is carried out on each sliding window, respectively, and machine learning algorithms are used for the extracted features to determine whether the window contains the object. This method means that the feature extraction and matching have certain defects. Meanwhile, the adaptability, accuracy, and detection speed of traditional algorithms are relatively poor [11]. With the development of deep learning [12–14], Hinton’s group won the ImageNet image recognition competition in 2012 by building the AlexNet [15] convolutional neural network to crush the performance of the second-place SVM classifier. As a result, many scholars have shifted their attention from traditional image processing to deep learning target recognition [16,17]. With the advantages of a simple structure, higher efficiency, and higher accuracy, the target detection algorithm based on deep learning has quickly caught up with the traditional target detection algorithms [18,19]. It has become the most mainstream target detection algorithm.

There are currently two main types of mainstream object detection algorithms. The first are known as two-stage algorithm models, such as R-CNN [20], Fast R-CNN [21], and Faster R-CNN [22]. This model first uses heuristics or RPN structures to generate a series of candidate boxes. Subsequently, it uses convolutional neural networks to regress and classify the samples. Using this process, it gains a higher precision, but a lower inference speed. The other algorithm is the single-stage algorithmic model, which uses regression ideas to input images into the convolutional neural network and output the result directly after the detection, such as YOLO [23], SSD [24], and Retina-Net [25]. This algorithm lacks the screening and optimization process of the prediction frameworks, which reduces the accuracy of the detection. Despite this, its detection speed is faster and higher than those of the two-stage methods.

With the fast development of deep learning, it has quickly been applied in fish detection, but there are still many challenges in the model for the accurate identification of fish. Liu et al. [26] detailed a fish recognition detection algorithm based on the FML-CenterNet, which introduces a feature fusion module in a CenterNet algorithm network structure to fuse low-level feature information and high-level feature information. On this basis, they put forward a more complete feature map, but the detection accuracy was not ideal. Cai et al. [27] constructed a CNN model for fish identification, using the ReLU function as the activation function through dropout and regularization, but increased the detection time. Dong et al. [28] detailed a network that mixed the spatial domain attention mechanism and hierarchical streamlined bilinear features together. Its feature extraction network was initialized with the parameters trained on the ImageNet dataset and further fine-tuned using the fish dataset, while the amount of computation was increased.

YOLO (You Only Look Once) [29–31], a classical single-stage detection algorithm, has achieved a good balance between accuracy and speed and is widely used in various target detection tasks. For example, Wu et al. [32] used a modified YOLO model to detect how pine nematode disease affected trees at different stages of infection. Wang et al. [33] used improved YOLO-V4 and binocular positioning for real-time vehicle identification and tracking during an agricultural operation. Qiu et al. [34] used a YOLO-based method to detect sidewalk cracks in real-time drone images. The above results show good accuracy when detecting targets that are more dispersed from the background. However, there will be still many problems if this model is to be directly applied to the accurate recognition of *T. rubripes*:

- (1) Compared to common scenarios, underwater images are affected by lighting, water flow, and water quality, etc., and the fish bodies in the images form a relatively complex background due to overlapping and occlusion, which increases the difficulty of the detection and causes inaccurate detection results.
- (2) In the feature extraction and fusion, the feature map output from each node is not fully utilized and the feature extraction ability can be further strengthened during training.

- (3) Due to the high density of cultured *T. rubripes* and the different target sizes in the images, the detection head of the YOLO-V7 needs to be improved.

In response to the above issues, based on the YOLO-V7 [35] algorithm framework, this paper proposes an accurate identification and detection algorithm for *T. rubripes* to solve the problem of the low accuracy of fish recognition in images.

2. Materials and Methods

2.1. Dataset

2.1.1. Data Acquisition and Image Features

This paper collected experimental data from the breeding ponds of the Dalian Tianzheng Breeding Factory, which raises different sizes of *T. rubripes* in ponds, and finally collected videos of the *T. rubripes*. The light in the breeding ponds was relatively fixed and soft. To avoid the influence of vertical light, the camera shot the water surface from bottom to top at a 30-degree angle. Considering the changes in light and turbidity, etc., the camera used a zoom lens and was kept in auto mode. The captured video size was 1920×1080 and one frame was extracted from the video every six seconds, eventually selecting 3870 images of *T. rubripes*.

2.1.2. Image Annotation and Dataset Production

The open-source script LabelImg (<https://github.com/tzutalin/labelImg>, accessed on 11 June 2021) on GitHub was used to annotate the dataset. After running the LabelImg script, the target samples in each image were marked. With these produced datasets, an XML file containing the target type and coordinate information was generated and trained. An example of an annotated image is shown in Figure 1. In the production of the dataset, 3870 images of *T. rubripes* were used. These comprised 3096 pictures used for the training dataset and 774 images used for the validation dataset.



Figure 1. Example of annotated image. (The green frame represents the labeled *Takifugu rubripes*).

2.2. Related Works

2.2.1. YOLO-V7

As the most typical representation of a single-stage object detection algorithm, the YOLO algorithm is based on deep neural networks for object recognition and localization. It uses a single CNN model to achieve end-to-end target detection, takes the whole image as the input into the network structure, and directly regresses the location of the bounding box and the category to which it belongs in the output layer. The YOLO-V7 network represents a continuous improvement over the previous the YOLO series, which provides a good balance between the accuracy and operating speed. The YOLO-V7 network consists of four main modules: input, backbone, head, and prediction. It adopts strategies such as extended efficient layer aggregation networks (E-ELAN), model scaling for concatenation-based models [36], re-parameterized convolution [37], and other techniques. The algorithm structure of YOLO-V7 is shown in Figure 2.

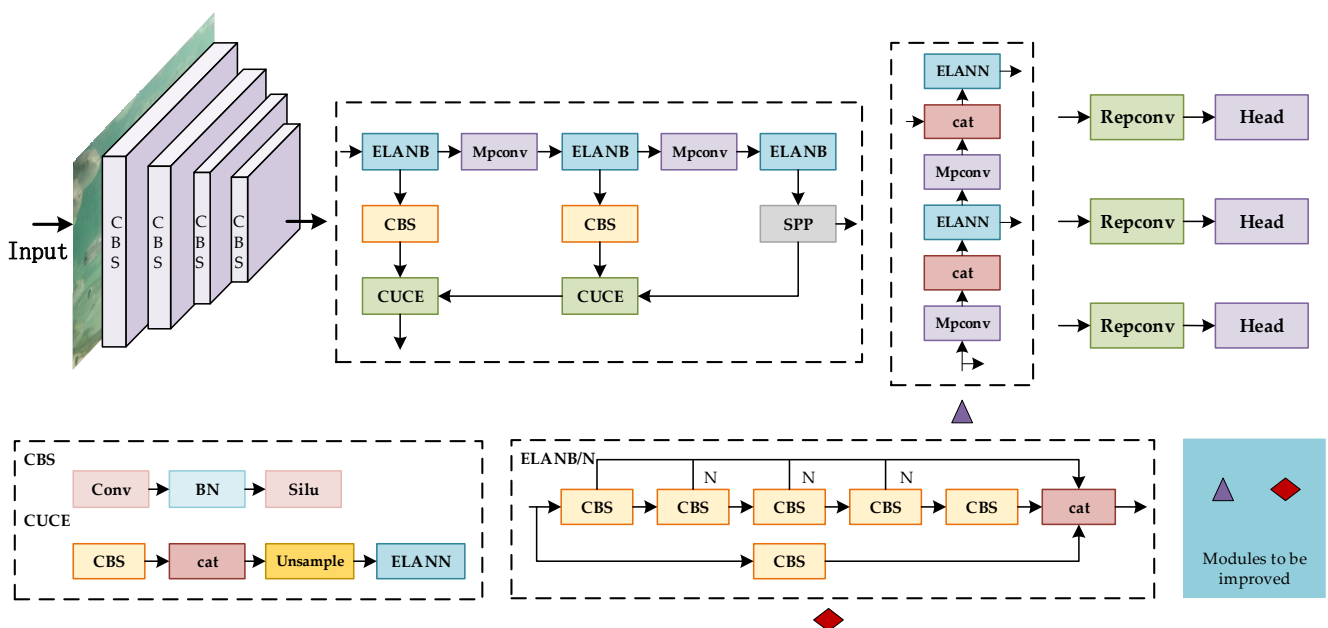


Figure 2. The overview architecture of YOLO-V7.

E-ELAN is a computational block in the YOLO-V7 backbone network that can guide different groups of computation blocks to learn more diverse characteristics. In large-scale ELANs, the network always reaches an equilibrium state, regardless of the gradient direction, path length, and total number of blocks. However, such equilibrium states may be destroyed and the usage of the main parameters will be reduced if the blocks are stacked endlessly. The E-ELAN algorithm uses expansion, random scrambling, and merging cardinality to continuously enhance the ability of the network learning without destroying the original gradient path, and to also guide the different computational block groups to learn more diverse features. The primary purpose of the model scaling is to adjust the specific properties of the model and generate models of different sizes to meet the needs of varying inference speeds. When scaling the model for a cascade-based model, only the depth in the computational block needs to be scaled, and the rest of the transport layer is scaled with the corresponding width. When scaling the depth factor of a calculated block, the change in the output channel of the block is calculated and makes the same changes to the transition layer. RepConv without constant connection is used to redesign the architecture of the reparameterized convolution and proposed to generate coarse to acceptable hierarchical labels with guidance from the prediction results of the guidance head, which are used to assist the learning of the guidance head. This paper presents an

improved algorithm based on YOLO-V7, and the research content flowchart is shown in Figure 3.

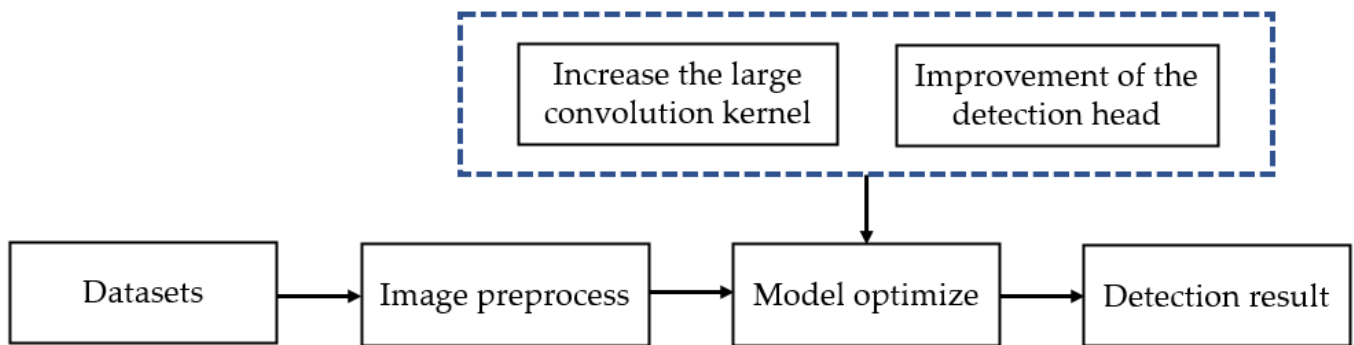


Figure 3. Research content flowchart.

2.2.2. Evaluation Metrics

An experiment needs performance indexes to evaluate an algorithm model. According to the evaluation indexes of the neural network model [38], this paper uses accuracy, recall rate, F_1 score, and average precision as its evaluation indicators. The calculations of the accuracy, recall, F_1 score, and average precision are shown in Equations (1)–(4).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$AP = \int_0^1 \text{Precision} \times \text{Recall} \, dr \tag{4}$$

$$AP_{@0.5:0.95} = \frac{1}{10} (AP_{@0.50} + AP_{@0.55} + \dots + AP_{@0.90} + AP_{@0.95}) \tag{5}$$

TP is the real example, that is, the sample correctly identified as *T. rubripes*; FP is the false positive example, which is the incorrectly identified sample of the *T. rubripes*; FN is the false counter example, that is, the sample wrongly identified as the background; TN is the true counter example, that is, the sample correctly identified as the background; ‘r’ represents the integral variable, which is used to determine the integration of precision * recall and is between 0 and 1; AP is the size of the area under the curve drawn by the accuracy–recall ratio (P-R); $AP_{@0.5}$ is the average of the accuracy at different recall values when the Intersection Over Union (IOU) is 0.5; and $AP_{@0.95}$ is the average of the accuracy at different recall values when the IOU is 0.95. $AP_{@0.5:0.95}$ is the average of the ten values, $AP_{@0.50}$, $AP_{@0.55}$, ..., $AP_{@0.90}$, and $AP_{@0.95}$, and the calculation formula is shown in Equation (5).

2.3. The Proposed Algorithm

This paper proposes an algorithm that is an improvement on YOLO-V7, based on data set characteristics. The sample code and pseudocode of the proposed algorithm are provided in Appendix A.

To solve the problem of the receptive field not being significantly improved after adding depth, it is necessary to increase the convolutional kernel. Compared to a large number of small convolutional kernels, a small number of large convolutional kernels can improve the receptive field and optimize the network backbone model. By these means,

the proposed algorithm will capture more effective information and enhance the ability of the feature extraction (Figure 4).

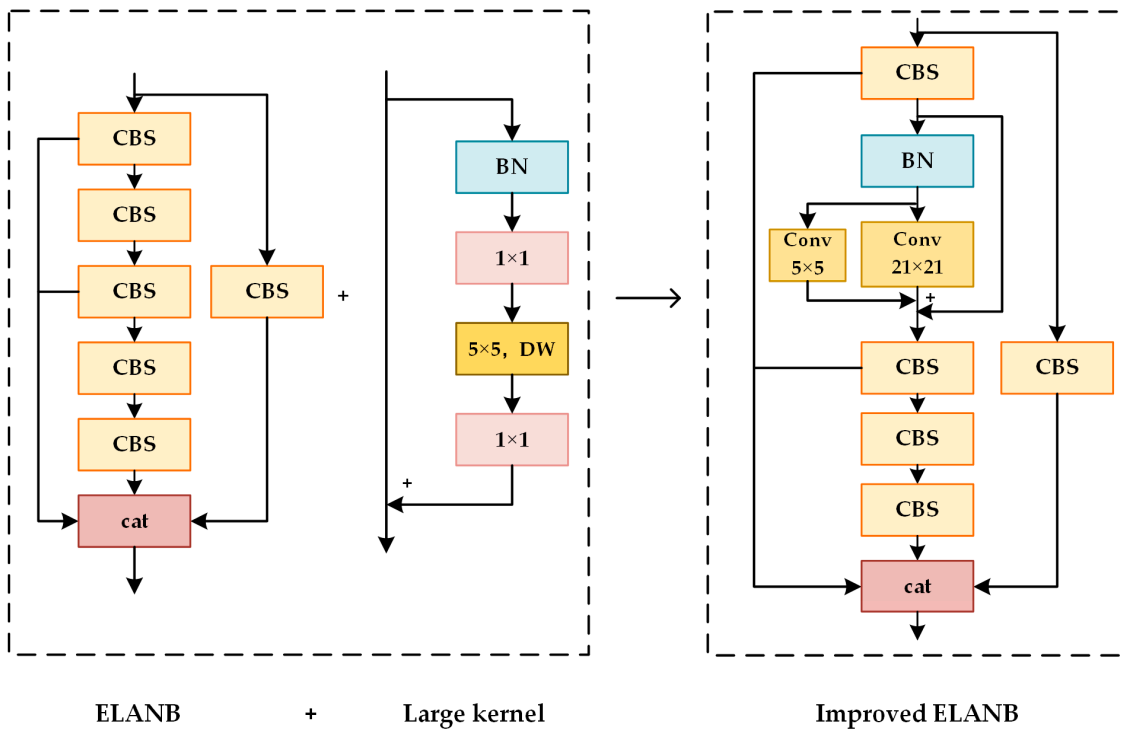


Figure 4. Improvements in feature extraction capability.

Based on the difficulty of the feature loss caused by the excessive occlusion between the targets, this paper upgrades the original detection head. With the progressive information flow, it clears up issues such as the multi-scale and insufficient extraction of small targets, thereby improving the accuracy of the target detection task (Figure 5).

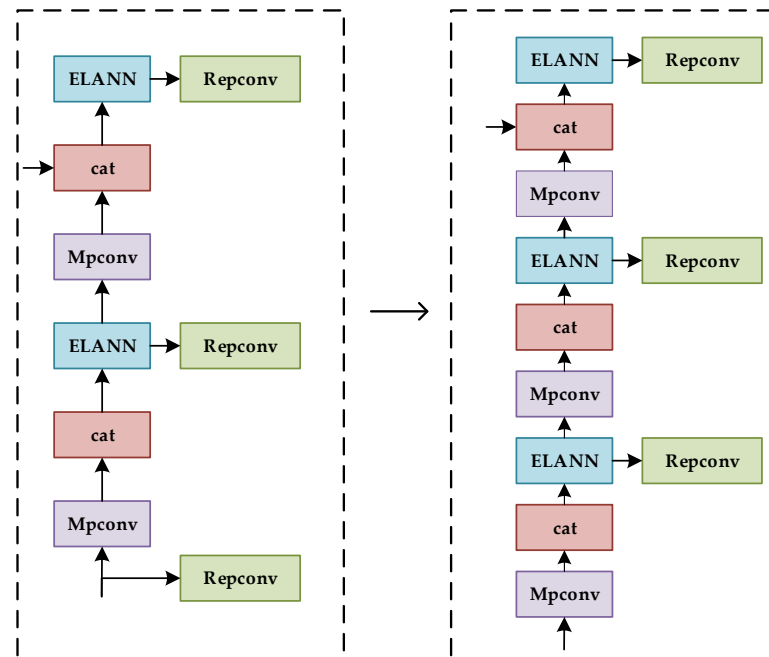


Figure 5. Improvements to the detection head.

2.3.1. Improvements in Feature Extraction Capabilities

During the experiment, there were missed and incorrect results when the original algorithm was used to detect the *T. rubripes*. After the evaluation, we found that most of the false detection or missed detection occurred in the area of dense fish. In order to capture the features clearly, this study hoped to expand the effective receptive field of the model. According to the effective receptive field theory, the size of the receptive field is proportional to the size of the convolution kernel, and the square root of the convolution kernel layers has a positive effect as well. Therefore, we considered that increasing the receptive field by adding depth was more effective for improving the receptive field than directly expanding the convolution kernel. For example, while ResNet [34] looks like it can go very deep, or go up to hundreds or thousands of layers, its effective depth is not very deep and a lot of its signals are from the shortcut layer, which does not increase its effective depth. However, if the convolution kernel size is increased, its effective receptive field tends to go up very significantly.

This paper added a large convolution kernel into an ELANB module to expand the effective receptive field of YOLO-V7, so that we could train a larger area. The structure of the improved algorithm is shown in Figure 4. We increased the convolution kernel of the second layer from 3×3 to 21×21 . This upgraded the extraction ability and achieved a more precise recognition result.

2.3.2. Improvement of the Detection Head

The head of the original YOLO-V7 algorithm has three sizes: large, medium, and small. Based on the actual distribution and sample situation of *T. rubripes*, the experiment found that obscured targets increased the error rate during the feature extraction. Furthermore, short distances caused interference and reduced the precision of the target detection, making the target extraction insufficient. Thus, this study proposed adding an object detection layer, abandoning the initially extracted feature map, deepening the layer based on the original, and further informing the information flow. The improved feature map found it easier to inform the feature information of the target and achieve the purpose of improving the detection accuracy, as shown in Figure 5.

3. Results

The software environment of our experiment is shown in Table 1. Considering the GPU memory limitation after adding large convolution kernels during the training, the batch size was set to 16. In order to analyze the training process perfectly, our study selected 300 iterations in the experiment. During the test, a batch of images was chosen with the same resolution in the training phase to verify the algorithm.

Table 1. Experimental environment.

Configuration	Parameter
CPU	Intel Xeon(R) Gold 5128R
GPU	Nvidia RTX 3090 Ti
Operating system	Ubuntu 20.04
Development environment	Pycharm 2022.2
Accelerated environment	CUDA11.1

3.1. Analysis of Training Results

Our study statistically analyzed the test results of the verification set. The sample number of *T. rubripes* in the verification set was 2280, the conf threshold (target confidence threshold) was set to 0.25, and the IOU was set to 0.45 for verification. The results are displayed in Table 2. In the improved model, the accuracy, recall rate, and F1 score were all improved by varying degrees. Among them, the accuracy increased by five percent, the recall by eight percent, and the F1 score by seven percent. The main reasons for this

increase in the recall rate were the significant increase in the TP and the significant decrease in the FN, which meant that more and more *T. rubripes* were correctly identified. This demonstrates the effectiveness of the improved algorithm.

Table 2. Comparison of evaluation indicators before and after improvement.

Conf-Thresh = 0.25 IOU = 0.5	Precision	Recall	F1-Score	TP	FP	FN
YOLO-V7	0.91	0.82	0.86	1859	174	421
Improved YOLO-V7	0.96	0.94	0.95	2154	79	126

3.2. Algorithm Performance Evaluation

3.2.1. Pre-Training

The pre-training dataset was composed of 3870 *T. rubripes* images, for which a weight file was obtained and the parameters during the pre-training set were as follows: firstly, this study set the initial learning rate to 0.01, made the weight attenuation 0.0005, set the batch size to 16, and performed 300 iterations to generate the pre-training weight files. The dataset was trained using transfer learning. Figure 6 shows the difference between using these transfer learning strategies and not using them. It points out that transfer learning improved the AP by 1.02%, which indicates that this strategy is effective.

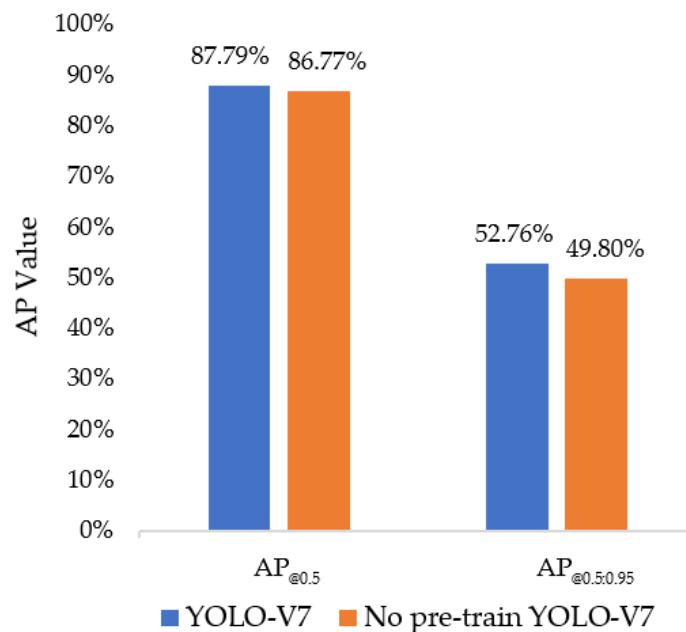


Figure 6. Pre-training comparison chart.

3.2.2. Performance Comparison to Improve Feature Extraction Capabilities

The experiment changed the effective receptive field of YOLO-V7 by changing the network structure of the ELANB. Although adding convolution kernels improved the precision, it also required more running memory. We selected convolution kernels with the sizes of 13×3 , 17×17 , 21×21 , and 27×27 for comparison. We comprehensively compared the number of parameters and running speeds and finally decided on a size of 21×21 for the convolution kernel. The following experiments were conducted to add the number of improved modules, replacing one, two, and four enhanced modules. The results of this experiment are shown in Figure 7. Considering the running speed and accuracy, we finally replaced two ELANB blocks. After strengthening the feature map, the trained AP increased from 87.7% to 91.3%. These results show that the enhanced ELANB block improved the correction of the training. At the same time, the large convolution kernel

strengthened the ability of the feature extraction and positively reduced the occurrence of error detection.

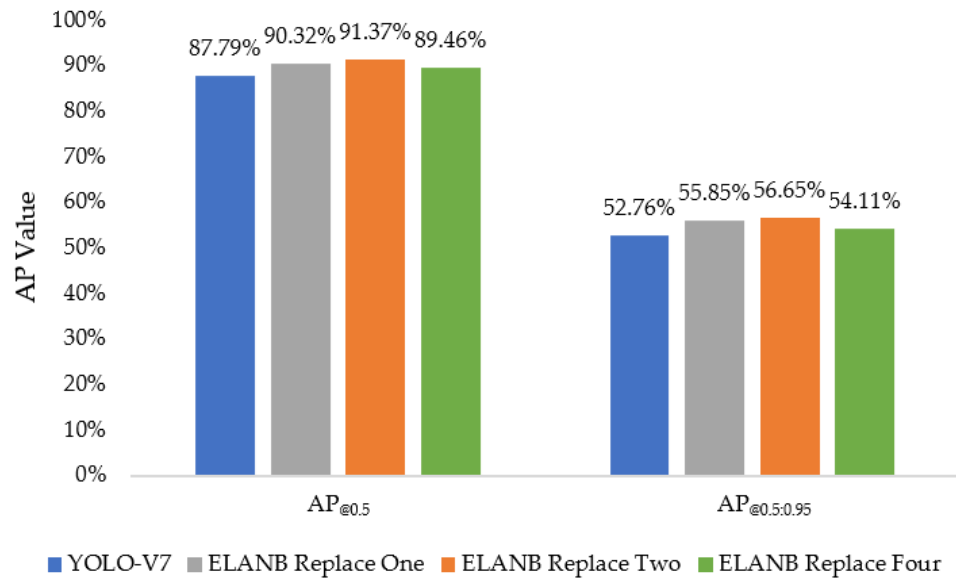


Figure 7. Improved feature extraction capability performance comparison.

3.2.3. Performance Comparison to Improve Head

This paper improved the detection head of the original YOLO-V7, abandoned the feature map of the original, and deepened it based upon the last feature extraction. Through this, the progressive information flow, multi-scale problem [39,40], and inadequate extraction of the small target information were effectively solved during the detection. The results of the enhanced comparative experiment are shown in Figure 8. The AP of the improved training increased from 87.79% to 89.81%. This shows that the enhanced head improved the detection accuracy.

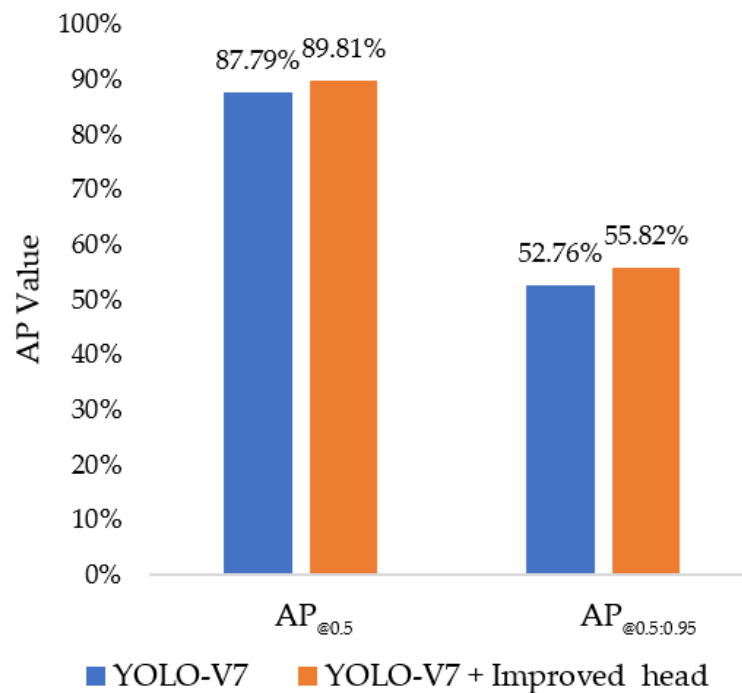


Figure 8. Comparison of the results of the improved detection head.

3.2.4. Comparison of Network Pruning Performance

Takifugu rubripes detection is a challenging detection task. Still, the network structure of YOLO-V7 is large and its accuracy has been improved by increasing the convolution kernel. This task selected the depth of the convolutional layer. Under the premise of ensuring the detection accuracy, we pruned the network appropriately. Table 3 shows the total amount of computation for each algorithm in a billion floating-point operations (GFlops). The results show that, after pruning the network, the whole calculation amount of the model was reduced by about 35%.

Table 3. Comparison of the number of network pruning parameters.

Model	YOLO-V7	Feature Extraction	Improved Head	Network Pruning	GFLOPs
1	✓				104.8
2	✓	✓			109.9
3	✓		✓		119.4
4 (Ours)	✓	✓	✓	✓	68.2

3.3. Performance Comparison of the Overall Algorithm

Table 4 compares the results of the different improvement strategies. The results show that the AP50 increased from 87.79% to 92.86%, with an increase of 5.07%. Through a series of improved operations, we successfully improved the detection accuracy for *T. rubripes*.

Table 4. Ablation experiment comparison results.

Model	YOLO-V7	Feature Extraction	Improved Head	Network Pruning	AP@0.5 (%)	AP@0.5:0.95 (%)
1	✓				87.79%	52.76%
2	✓	✓			91.37%	55.82%
3	✓		✓		89.81%	56.65%
4 (Ours)	✓	✓	✓	✓	92.86%	57.94%

In order to further analyze the performance of the proposed method, we compared it with YOLO-V5, Faster R-CNN, and SSD. We used the same training, verification, and test set to compare the five networks, the results of which are shown in Table 5. The improved YOLO-V7 has a higher accuracy than the other models.

Table 5. Comparison with current mainstream detection algorithms.

Model	AP@0.5 (%)	AP@0.5:0.95 (%)
YOLO-V5	87.11%	51.80%
Faster R-CNN	88.71%	53.55%
SSD	82.26%	46.43%
YOLO-V7	87.79%	52.76%
Ours	92.86%	57.94%

Figure 9 compares the PR curves of the *T. rubripes* before and after the improvement of YOLO-V7. The closed area of the PR curve before the improvement was much smaller.

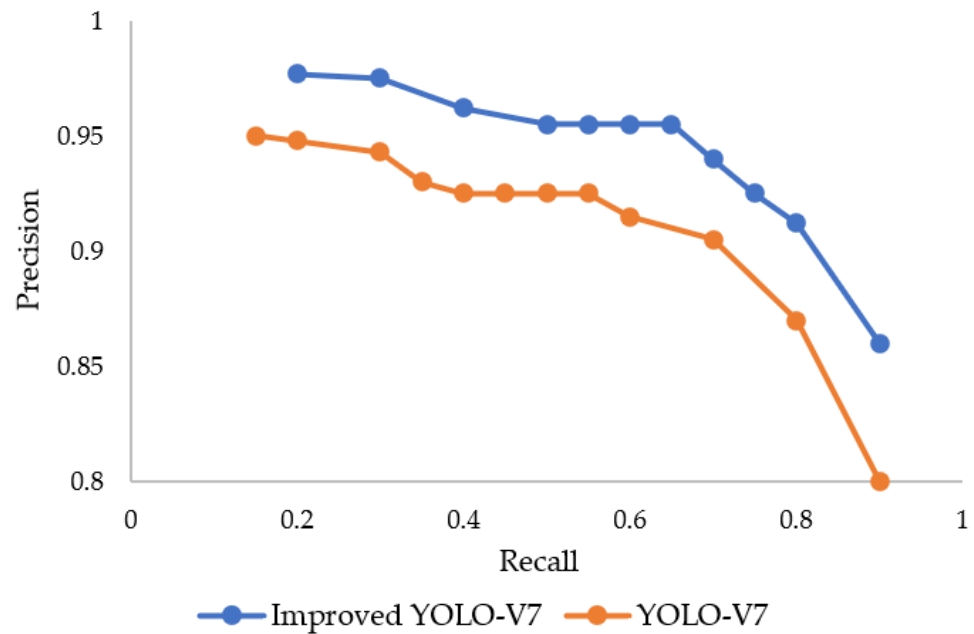


Figure 9. Comparison of PR curves of the model.

Figure 10 shows an example image of the before and after comparison results of some of the test picture improvements. The red box represents the detection result of the original YOLO-V7 and the purple box represents the result of the improved YOLO-V7. The yellow box indicates that the *T. rubripes* could be correctly detected after improving the algorithm. The modified YOLO-V7 network could effectively improve the detection accuracy for the *T. rubripes*. The reason for this was that the enhanced network increased the ability to process feature maps. In this way, *T. rubripes* could be detected correctly when they occurred at higher densities, with an overlapping environment between the fish.



Figure 10. Comparison of test picture results.

The results of the whole screen, before and after the improvement, are shown in Figure 11. Under the interference of high density, the grid units in the feature map extraction network did not perform accurately enough to predict the target, resulting in omissions and error detection in the original YOLO-V7. After the improvement, the detection effect was improved, 47 targets were obtained from the original analysis, 56 targets were obtained after the improvement in picture (a), 40 targets were obtained from the original analysis,

and 47 targets were obtained after the improvement in picture (b). This shows that the improved network was better than the unimproved one.

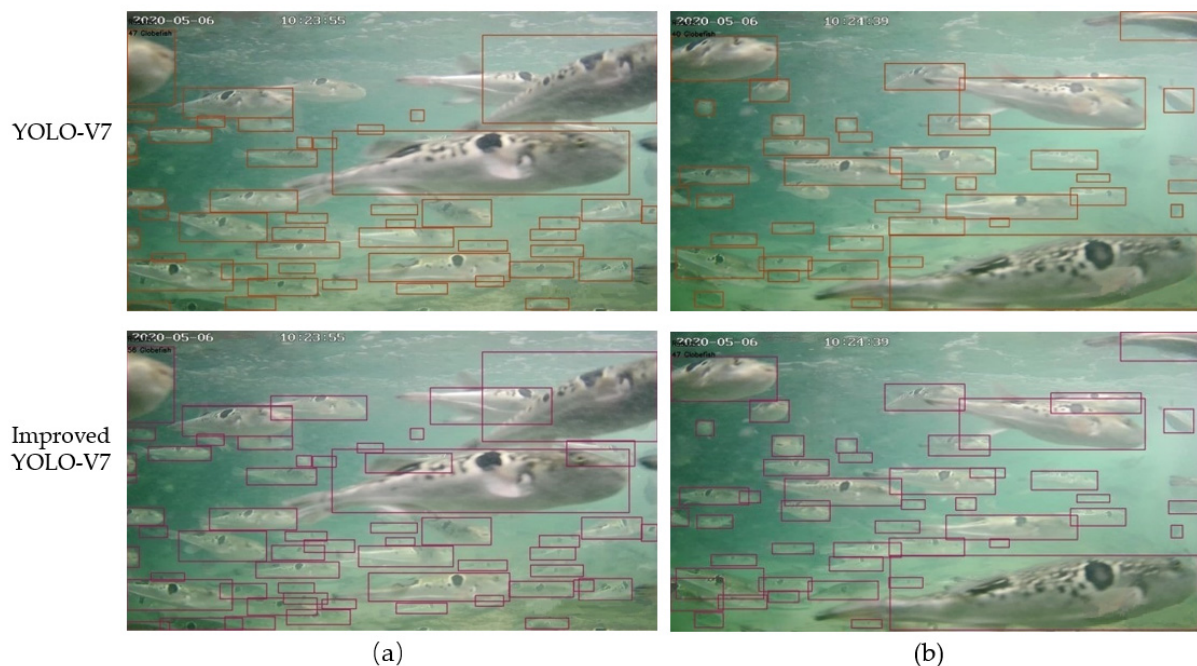


Figure 11. Comparison of test results. (a) 47 targets were obtained from the original analysis, 56 targets were obtained after the improvement and (b) 40 targets were obtained from the original analysis, and 47 targets were obtained after the improvement. (The first line shows the detection result of the original YOLO-V7 and the second line shows the result of the improved YOLO-V7).

4. Conclusions and Discussion

In this paper, we proposed an improved YOLO-V7 network for accurately detecting *T. rubripes*. The feature extraction ability and detection head of YOLO-V7 were all modified in our new model. The new model ameliorated the situation of low-quality underwater images and more overlapping and dense identification targets, leading to a higher precision. The experimental results show that the AP increased from 87.79% to 92.86%, with a total increase of 5.07%, which means that the improved YOLO-V7 network was better the original version.

This paper effectively solved the problem of fish identification needing to be improved in the cultivation of *T. rubripes* when the background is relatively simple. The recognition of fish images with a complex background is not as good as that with a single background in a practical situation. The intelligence of aquaculture still needs more research and exploration, and our next step will be studying fish identification in complex backgrounds.

Author Contributions: Conceptualization, S.Z.; methodology, S.Z. and K.C.; software, S.Z. and K.C.; validation, S.Z. and X.T.; formal analysis, S.Z. and K.C.; writing—original draft preparation, S.Z.; writing—review and editing, S.Z.; visualization, S.Z., J.H. and X.S.; supervision, Y.F. and H.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the province scientific research project Education Department of Liaoning [JL201917].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: The authors are grateful for the support from the Education Department of Liaoning Province, China.

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

Appendix A

Algorithm A1: The pytorch style code of the improved LgConv & LKDeXt module

```

1: class LgConv(nn.Module):
2:     def __init__(self, in_channels, dw_channels, block_lk_size, small_kernel, drop_path,
small_kernel_merged=False):
3:         super().__init__()
4:         self.pw1 = conv_bn_relu(in_channels, dw_channels, 1, 1, 0)
5:         self.pw2 = conv_bn(dw_channels, in_channels, 1, 1, 0)
6:         self.large_kernel = ReparamLargeKernelConv(dw_channels, dw_channels,
block_lk_size, 1, dw_channels, small_kernel, small_kernel_merged)
7:         self.lk_nonlinear = nn.ReLU()
8:         self.prelkb_bn = get_bn(in_channels)
9:         self.drop_path = DropPath(drop_path) if drop_path > 0. else nn.Identity()

10:    def forward(self, x):
11:        out = self.prelkb_bn(x)
12:        out = self.pw1(out)
13:        out = self.large_kernel(out)
14:        out = self.lk_nonlinear(out)
15:        out = self.pw2(out)
16:        return x + self.drop_path(out)

17: class LKDeXt(nn.Module):
18:    def __init__(self, c1, c2, n=1, True, g=1, e=0.5):
19:        super().__init__()
20:        c_ = int(c2 * e)
21:        self.cv1 = Conv(c1, c_, 1, 1)
22:        self.cv2 = Conv(c1, c_, 1, 1)
23:        self.cv3 = Conv(2 * c_, c2, 1)
24:        self.m = nn.Sequential*(LgConv(c_, c_, 21, 5, 0.0, False) for _ in range(n))
25:    def forward(self, x):

26:        return self.cv3(torch.cat((self.m(self.cv1(x)), self.cv2(x)), dim=1))

```

Algorithm A2: The pseudocode of the improved ConvBlock & LKDeXt

```

1: def ConvBlock(x):
2:     x = Conv(x)
3:     x = Batch_norm(x)
4:     x = ReLU(x)
5:     return x
6: def LgConv(x):
7:     y = Batch_norm(x)
8:     y = ConvBlock(y)
9:     y = ReparamLargeKernelConv (y)
10:    y = ReLU(y)
11:    y = ConvBlock(y)
12:    return x + dropout(y)

13: def LKDeXt(x):
14:    y = Conv(concat(LgConv(Conv(x)),Conv(x),dim=1))
15:    return y

```

References

- Guo, R.; Zhang, X.; Su, H.; Liu, H. The research status of nutrition value and by-products utilization of puffer fish. *J. Food Sci. Technol.* **2018**, *3*, 113–116.
- Yang, D.; Zhang, S.; Tang, X. Research and development of fish species identification based on machine vision technology. *Fish. Inf. Strategy* **2019**, *31*, 112–120.
- Sun, L.; Wu, Y.; Wu, Y. Multi-objective fish object detection algorithm is proposed to study. *J. Agric. Mach.* **2019**, *50*, 260–267.
- Tu, B.; Wang, J.; Wang, S.; Zhou, X.; Dai, P. Research on identification of freshwater fish species based on fish back contour correlation coefficient. *Comput. Eng. Appl.* **2016**, *52*, 162–166.
- Wan, P.; Zhao, J.; Zhu, M.; Tan, H.; Deng, Z.; Huang, S.; Wu, W.; Ding, A. Freshwater fish species identification method based on improved ResNet50 model. *J. Agric. Eng.* **2021**, *12*, 159–168.
- Liu, S.; Li, G.; Tu, X.; Meng, F.; Chen, J. Research on the development of aquaculture production information technology. *Fish. Mod.* **2021**, *48*, 64–72.
- Zhao, Z.; Liu, Y.; Sun, X.; Liu, J.; Yang, X.; Zhou, C. Compositing FishNet: Fish detection and species recognition from low-quality underwater videos. *IEEE Trans. Image Process.* **2021**, *30*, 4719–4734. [[CrossRef](#)]
- Li, S.; Yang, L.; Yu, H.; Chen, Y. Underwater fish species identification model and real-time recognition system. *J. Intell. Agric.* **2022**, *4*, 130–139.
- Wang, W.; Jiang, H.; Qiao, Q.; Zhu, H.; Zheng, H. Research on fish recognition and detection algorithm based on deep Learning. *J. Inf. Technol. Netw. Secur.* **2020**, *33*, 6157–6166.
- Sun, S.; Zhao, J. Pattern Recognition and Machine Learning. *J. Sci. Technol. Publ.* **2021**, *322*, 154.
- Li, J.; Xu, L. Research hot trend prediction model based on machine learning algorithm comparison and analysis, the BP neural network, support vector machine (SVM) and LSTM model. *Mod. Intell.* **2019**, *33*, 23–33.
- Amanullah, M.; Selvakumar, V.; Jyot, A.; Purohit, N.; Fahlevi, M. CNN based prediction analysis for web phishing prevention. In Proceedings of the International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 1–3 December 2022; pp. 1–7.
- Althubiti, S.A.; Alenezi, F.; Shitharth, S.; Reddy, C.V.S. Circuit manufacturing defect detection using VGG16 convolutional neural networks. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1070405. [[CrossRef](#)]
- Alyoubi, K.H.; Shitharth, S.; Manoharan, H.; Khadidos, A.O.; Khadidos, A.O. Connotation of fuzzy logic system in underwater communication systems for navy applications with data indulgence route. *Sustain. Comput. Inform. Syst.* **2023**, *38*, 100862. [[CrossRef](#)]
- Krizhevsky, A.; Sutskever, I.; Hinton Geoffrey, E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
- Naseer, A.; Baro, E.N.; Khan, S.D.; Vila, Y. A novel detection refinement technique for accurate identification of nephrops norvegicus burrows in underwater imagery. *Sensors* **2022**, *12*, 4441. [[CrossRef](#)] [[PubMed](#)]
- Shitharth, S.; Prasad, K.M.; Sangeetha, K.; Kshirsagar, P.R.; Babu, T.S.; Alhelou, H.H. An enriched RPCO-BCNN mechanisms for attack detection and classification in SCADA systems. *IEEE Access* **2021**, *9*, 156297–156312. [[CrossRef](#)]
- Sun, H.; Li, Y.; Lin, Y. Significant target detection based on deep learning review. *J. Data Acquis. Process.* **2023**, *38*, 21–50.
- Qian, C. Target detection algorithm based on depth of learning research progress. *J. Wirel. Commun. Technol.* **2022**, *31*, 24–29.
- Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Washington, DC, USA, 23–28 June 2014; IEEE: Piscataway, NJ, USA, 2014.
- Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
- Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
- Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
- Liu, Y.; Wang, Y.; Hunag, L. Fish recognition and detection based on FML-Centernet algorithm. *Laser Optoelectron. Prog.* **2022**, *59*, 317–324.
- Cai, W.; Pang, H.; Zhang, Y.; Zhao, J.; Ye, Z. Recognition model of farmed fish species based on convolutional neural network. *J. Fish. China* **2022**, *46*, 1369–1376.
- Dong, S.; Liu, W.; Cai, W.; Rao, Z. Fish recognition based on hierarchical compact bilinear attention network. *Comput. Eng. Appl.* **2022**, *5*, 186–192.
- Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
- Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.

31. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO series in 2021. *arXiv* **2021**, arXiv:2107.08430.
32. Wu, K.; Zhang, J.; Yin, X.; Wen, S.; Lan, Y. An improved YOLO model for detecting trees suffering from pine wilt disease at different stages of infection. *Remote Sens. Lett.* **2023**, *14*, 114–123. [[CrossRef](#)]
33. Wang, L.; Li, L.; Wang, H.; Zhu, S.; Zhai, Z.; Zhu, Z. Real-time vehicle identification and tracking during agricultural master-slave follow-up operation using improved YOLO v4 and binocular positioning. *Proc. Inst. Mech. Eng.* **2023**, *237*, 1393–1404. [[CrossRef](#)]
34. Qiu, Q.; Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom. Constr.* **2023**, *147*, 104745. [[CrossRef](#)]
35. 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.
36. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13024–13033.
37. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style convnets great again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737.
38. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F score, with implication for evaluation. In Proceedings of the European Conference on Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005; pp. 345–359.
39. Khan, S.D.; Basalamah, S. Multi-Scale person localization with multi-stage deep sequential framework. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 1217–1228. [[CrossRef](#)]
40. Khan, S.D.; Alarabi, L.; Basalamah, S. A unified deep learning framework of multi-scale detectors for Geo-spatial object detection in high-resolution satellite images. *Arab. J. Sci. Eng.* **2022**, *47*, 9489–9504. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.