Journal of
Marine Science and Engineering

MDPI

Article

Underwater Biological Detection Based on YOLOv4 Combined with Channel Attention

Aolun Li [1], Long Yu [1,*] and Shengwei Tian [2,*]

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830000, China; alenli@stu.xju.edu.cn
[2] School of Software, XinJiang University, Urumqi 830000, China
* Correspondence: yul@xju.edu.cn (L.Y.); tsw@xju.edu.cn (S.T.)

Abstract: Due to the influence of underwater visual characteristics on the observation of underwater creatures, the traditional object detection algorithm is ineffective. In order to improve the robustness of underwater biological detection, based on the YOLOv4 detector, this paper proposes an underwater biological detection algorithm combined with the channel attention mechanism. Firstly, the backbone feature extraction network CSPDarknet53 of YOLOv4 was improved, and a residual block combined with the channel attention mechanism was proposed to extract the weighted multi-scale effective features. Secondly, the weighted features were repeatedly extracted through the feature pyramid to separate the most significant weighted features. Finally, the most salient weighted multi-scale features were used for underwater biological detection. The experimental results show that, compared with YOLOv4, the proposed algorithm improved the average accuracy of the Brackish underwater creature dataset detection by 5.03%, and can reach a detection rate of 15fps for underwater creature video clips. Therefore, it is feasible to apply this method to the accurate and real-time detection of underwater creatures. This research can provide technical reference for the exploration of marine ecosystems and the development of underwater robots.

Keywords: YOLOv4; underwater biological detection; channel attention; multi-class

Citation: Li, A.; Yu, L.; Tian, S. Underwater Biological Detection Based on YOLOv4 Combined with Channel Attention. *J. Mar. Sci. Eng.* **2022**, *10*, 469. https://doi.org/10.3390/jmse10040469

Academic Editor: Hiroshi Kuroda

Received: 25 February 2022
Accepted: 24 March 2022
Published: 26 March 2022

1. Introduction

In ocean observation research, the detection, location, and tracking of marine organisms is one of the most basic tasks for deeply exploring the marine ecosystem. Tracking and studying the habitats of underwater populations and any changes in their species is critical for the conservation and use of marine resources [1].

One of the most common ways to observe underwater life is to collect video and analyze it. The main research work can be divided into manual observation, traditional object detection, and object detection based on deep learning. Among them, manual observation cannot provide a comprehensive observation, cannot capture the real underwater biological state and behavior, and requires expensive equipment and human resources. The traditional underwater biometrics algorithm has a good application ability for some underwater scenes with single object and few changes. In an early study, M. M. M. Fouad et al. [2] used improved support vector machines (SVMs) to classify tilapia with better accuracy than other machine learning techniques at the time. To detect moving underwater objects, Shen et al. [3] proposed a hierarchical background model from the perspective of bionics with reference to the eyes of frogs, which have a good detection ability. Considering the limited underwater computing resources, Wang et al. [4] proposed a fast method to segment underwater images using a modified Markov Random Field (MRF) model combined with hard clustering. To address the problem of low underwater visibility, Shevechenko et al. [5] proposed a fish detection framework by capturing noisy videos in low-visibility water and using a suitable background removal method. Although the above-mentioned traditional methods have made some practical progress in some specific cases, it is difficult

to obtain a satisfactory underwater multi-object and multi-class recognition rate due to underwater images often have a series of problems compared with ordinary optical images, such as poor visibility, non-rigid deformation, high-frequency changes of appearance under different illumination, and different viewing angles, etc. [6].

With the rapid development of deep learning technology, many scholars have applied convolutional neural networks (CNN) to the field of computer vision. They are mainly divided into two categories; one is object detection algorithm based on candidate region (i.e., two-stage object detector), and the other is regression-based object detection algorithm (i.e., one-stage object detector). The two-stage object detector first selects candidate regions for the input image, and then classifies and regresses the candidate regions to achieve object detection. The one-stage object detector omits the candidate region generation step, and directly integrates the process of feature extraction, object classification, and position regression into a convolutional neural network, which simplifies the object detection process into an end-to-end regression problem. Single-stage detection frameworks include SSD [7], YOLO series [8–10], and RetinaNet [11]. Two-stage detection frameworks include FPN [12] and Faster R-CNN [13]. In underwater fish detection tasks, the two main goals of the research are to improve the detection accuracy and speed. Krizhevsky et al. [14] first proposed a deep CNN model whose accuracy surpassed the second-place model in the ILSVRC2012 competition by 30%. However, the real-time detection of the model cannot be guaranteed. Therefore, some scholars have used a single-stage object detection algorithm in underwater fish detection. Sung et al. applied the architecture of YOLO and proposed a vision-based convolutional neural network real-time fish detection system [15]. Wang et al. proposed a detector for classifying and detecting fish images using YOLOv2 [16]. Performing image augmentation to generate depth information, Liu et al. developed an underwater fish detection and tracking strategy [6]. These methods solve the task of the real-time detection of underwater fish to a certain extent, but the model still needs to be improved in terms of real-time performance, detection accuracy, and detection speed.

This paper focuses on improving the accuracy of underwater biological detection tasks and ensuring real-time detection. In addition, most of the previous studies focused on the detection of fish, but underwater life is diverse, and the detection of fish alone cannot provide a comprehensive understanding of the underwater environment. Based on this situation, we extended the study of underwater organisms to six categories: big fish, small fish, starfish, shrimp, jellyfish, and crabs. The main contributions of this paper are as follows:

(1) The YOLO model has taken into account both accuracy and efficiency since it was proposed. By embedding an attention mechanism into the well-performing YOLOv4 detection network, the model detection ability and quality can be effectively improved with a relatively small increase in the amount of computation;

(2) We expanded the types of real-time detection of underwater creatures to six categories, and used the improved YOLOv4 model to locate, detect and classify different types of underwater creatures;

(3) We evaluated the experiments on the Brackish dataset, and the results show that our network model outperformed the baseline by 5.03%. Therefore, our method outperforms the original baseline method.

The rest of the paper is organized as follows: Section 2 summarizes the basic model used in this paper and the improvements made, Section 3 designs the experiments and analyzes the experimental results to demonstrate the effectiveness of the proposed model, and Section 4 summarizes the full text and the future research direction of development.

2. Model Structure

2.1. Basic Model

YOLOv4 is used in this process. It tests the impact of a large number of advanced technologies on object detection performance based on the YOLOv3 object detection architecture, and adds practical techniques to the architecture to achieve the best balance

between detection speed and accuracy. According to the test results in the COCO object detection dataset, YOLOv4 is faster and more accurate than YOLOv3, which improves the mAP and FPS of YOLOv3 by 10% and 12%, respectively. Meanwhile, compared with other advanced object detection methods, such as EfficientDet, YOLOv4 has twice the detection speed and comparable performance [17].

The YOLOv4 detector is able to predict the location of the bounding box and the probability of its class directly from the entire image through just one neural network, which turns object detection into a regression problem. Therefore, it is very suitable for real-time detection of applications. YOLOv4 mainly includes CSPDarkNet53 feature extraction backbone network, SPPNet (Spatial Pyramid Pooling Network), PANet (Path Aggregation Network) multi-scale prediction network, and network prediction head (YOLO Head). Among them, CSPDarknet53 consists of 5 large residual blocks. The number of small residual units contained in these 5 large residual blocks is 1, 2, 8, 8, and 4, respectively, that is, two CBM convolution modules and one CSPX. The convolutional modules together form a large residual block. After the last feature layer output of CSPDarknet53, SPPNet is added, and four different scales of maximum pooling are used for processing. The pooling kernel sizes of maximum pooling are 13×13, 9×9, 5×5, 1×1 (1×1 means no processing); this pooling operation can improve the inference ability of the model with little increase in the inference time of the model. PANet consists of two parts, upsampling and downsampling, making full use of feature fusion. YOLOv4 has 3 prediction heads, and the output shapes of the 3 prediction heads are (13, 13, 255), (26, 26, 255), (52, 52, 255), which are used to extract multi-scale features for object detection. The complete network structure is shown in Figure 1. This network structure can train a deep learning model to predict the bounding box of the object and the probability of its class from the original image.

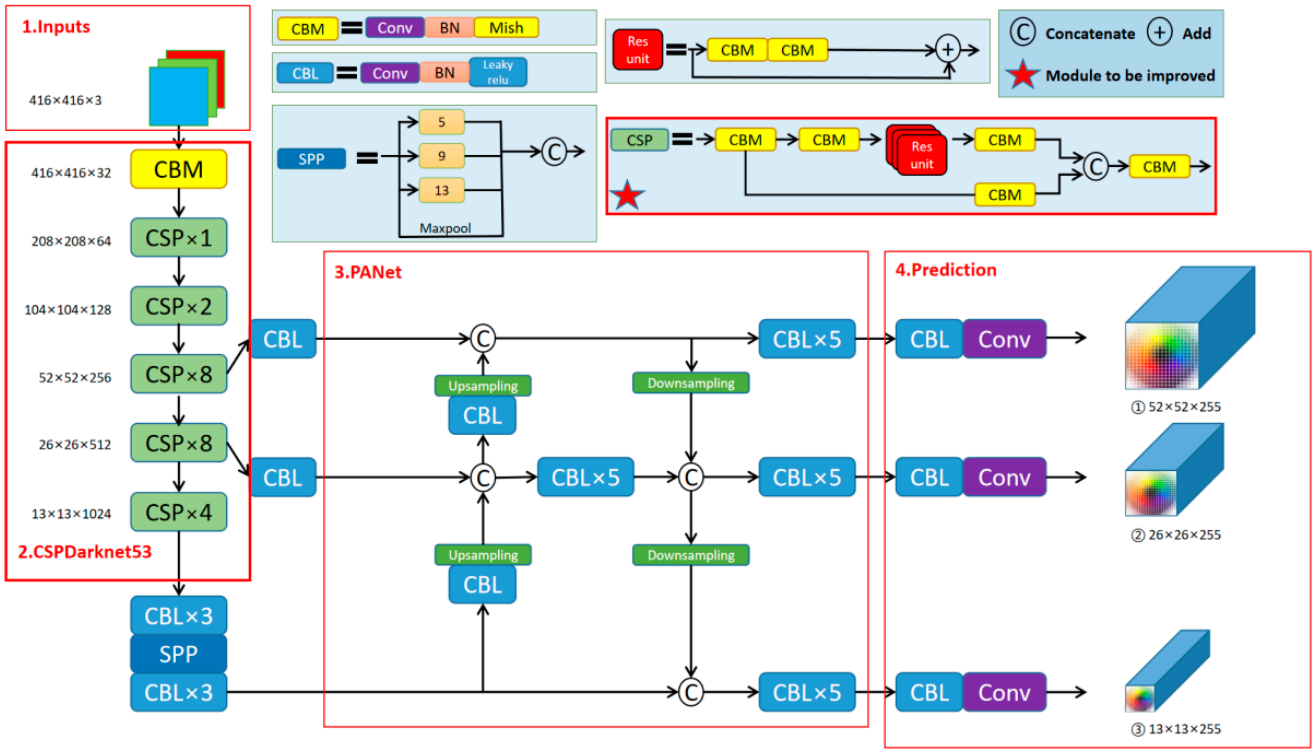

Figure 1. The network structure of YOLOv4.

2.2. Channel Attention

During each convolution process, some complex interference information will inevitably be distributed on some channels, resulting in reduced network performance. The attention mechanism has been widely used in neural networks [18–20]. With the in-depth study of the channel attention mechanism, the channel weight of each piece of channel

information is adjusted, and different weights are assigned to each piece of channel information. The channel information is screened, which can effectively alleviate the influence of the interference information. A typical representative of channel attention is SENet (Squeeze-and-Excitation Network) [21]. This module mainly includes two key parts: global information embedding (Squeeze), and adaptive recalibration (Excitation). This enables the neural network to learn to perform feature recalibration using global information to selectively emphasize important features and suppress unimportant features, in order to enhance the representation capability of the entire network. The Squeeze operation compresses the spatial dimension according to the channel information, that is, compresses the global spatial information into a channel descriptor with a global receptive field. Its output dimension matches the number of input feature channels, which is used to characterize the global distribution of response on the feature channel. The statistical information $\mathbf{z} \in \mathrm{R}^{\mathrm{C}}$ is generated by compression on the spatial dimension W × H, where the c-th element of $\mathbf{z}$ is calculated as shown in Equation (1):

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{W \times H} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{1}$$

where $\mathbf{F}_{sq}(\cdot)$ is the Squeeze operation, and $\mathbf{u}_c$ is the c-th feature. The Excitation operation aims to capture the dependencies of the channels, and generates weights for each feature channel by learning to explicitly model the correlation between feature channels. This operation needs to capture the nonlinear interaction between channels, and at the same time, two fully connected layers are required to prevent the model from becoming complicated and to improve generalization. Therefore, the Excitation operation is mainly composed of two fully connected layers and two activation functions, as shown in Equation (2):

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \mathrm{Sigmoid}(\mathbf{W}_2 \times \mathrm{ReLU}(\mathbf{W}_1, \mathbf{z})) \tag{2}$$

where $\mathbf{F}_{ex}$ is the Excitation operation, $\mathrm{Sigmoid}(x) = \frac{1}{1+e^{-x}}$ and $\mathrm{ReLU}(x) = max(0, x)$ are the activation functions.

The final output is obtained by multiplying the respective weights by the input channels, as shown in Equation (3):

$$\mathbf{X}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \cdot \mathbf{u}_c \tag{3}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_c]$, $\mathbf{F}_{scale}(\mathbf{u}_c, s_c)$ refers to the product of the corresponding channels between the feature map $\mathbf{u}_c \in R^{H \times W}$ and the scalar s_c. Its structure is shown in Figure 2.

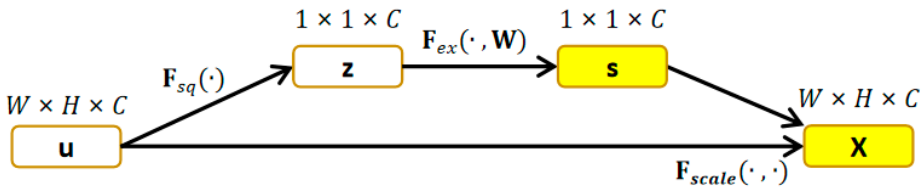

Figure 2. The structure of SENet.

Therefore, the channel attention mechanism significantly improves the classification ability of the model by explicitly modeling the interdependencies between feature channels and adaptively extracting the weights of different channels. This paper uses it in the feature extraction backbone network, attempting to improve its ability to extract image features.

2.3. Proposed Model

The proposed model is an improvement on YOLOv4 based on the dataset features, and this study took three main steps to optimize the network model. In this section, we will describe how to customize the network model structure and use it for detection algorithms.

First, since there are detection targets, such as small fish and shrimp, in the dataset that are much smaller than the image background, we adjust the grid size of the last convolutional layer. The grid is adjusted from the original 7×7 grid to an improved 9×9 grid, which enables detection of smaller underwater creatures.

Second, in order to use YOLOv4 on the Brackish dataset, we adjusted the number of filters in the last convolutional layer. The filters here represent the depth of the output layer, which is closely related to the number of categories in the dataset. The calculation of the number of filters is shown in Equation (4):

$$N_{filters} = (N_{cls} + N_{coords} + 1) \times N_{mask} \quad (4)$$

where N_{cls} is the number of detected object categories, N_{coords} is the number of bounding box coordinates, and N_{mask} is the number of anchor indexes. In this study, there were 6 types of objects, 4 bounding box coordinates (x, y, w, h), and 3 anchor indexes, so the number of filters was 33.

Finally, the channel attention mechanism is embedded in the backbone feature extraction network to extract the weighted multi-scale effective features, thereby improving the detection accuracy of the model. The improvement of the backbone feature extraction network part is shown in Figure 3.

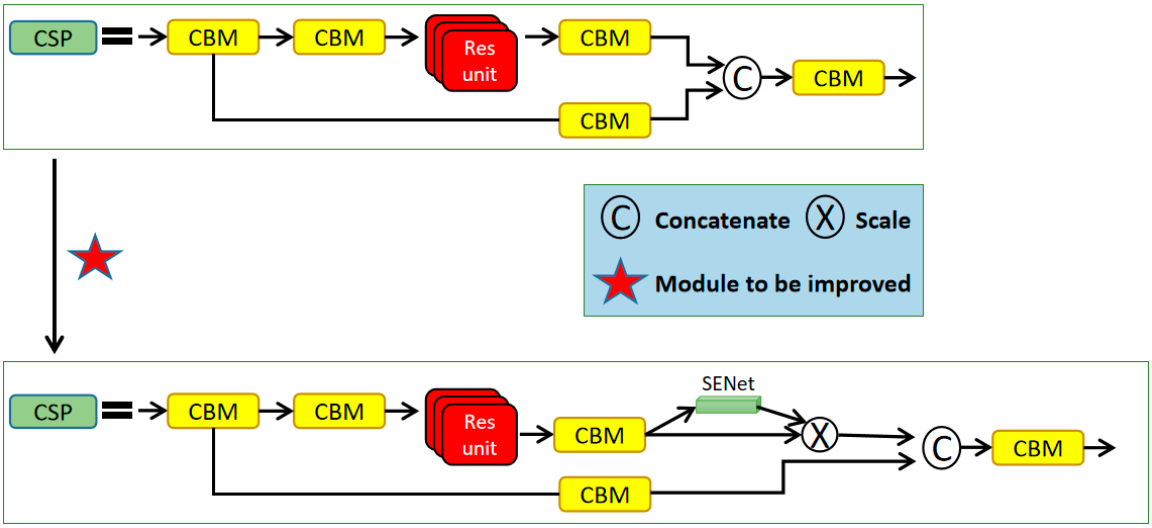

Figure 3. Improvement of the proposed method on backbone feature extraction network.

3. Experiment and Analysis

3.1. Datasets

This paper adopted the Brackish dataset, a publicly available underwater dataset that contains 89 video clips captured in temperate seawater with varying visibility [22]. The provider used a bounding box annotation tool [23] to manually annotate 14,518 frames, resulting in 25,613 annotations.

The dataset was categorized into six classes of underwater creatures: fish, small fish, starfish, shrimp, jellyfish, and crabs. Due to the turbid water quality and the relatively similar visual appearance of various fish, it is difficult for relevant experts to distinguish the specific species of fish from the video, so these kinds of fish were roughly divided into fish and small fish. After converting the video into 15,084 pictures by frame, it was randomly divided into training, validation, and test data according to the proportion of 80%, 10% and 10%. The number of species included in each dataset is shown in Table 1.

Table 1. The details of partitioning the dataset.

Species	Train	Val	Test	Total
Fish	2622	307	312	3241
Small_fish	7722	1024	809	9555
Crabs	415	76	57	548
Shrimp	5206	665	667	6538
Jellyfish	520	55	62	637
Starfish	4100	501	492	5093
Total	20,585	2628	2399	25,612

3.2. Experimental Details

To fit the input channel of YOLOv4, the image resolution was rescaled to 416 × 416. At the same time, in order to prevent the occurrence of detection overfitting, data augmentation methods were also used for the training set, including image flipping, scaling, and random erasing.

In this paper, for the backbone network CSPDarkNet53, the pre-training model of the classification task under the Image-Net large dataset was used to speed up the network convergence speed. According to the experimental conclusion of a previous study [17], it is pointed out that the training stage was divided into two stages in order to obtain better results. The first stage froze the network parameters of the backbone network CSPDarkNet53 and adjusted the parameters of the non-backbone network, and the optimizer used Adam. This stage is a coarse adjustment stage, and the maximum learning rate should be set to be larger than the second stage, i.e., set to 0.001. Since only the non-backbone network parameters were adjusted, the Batch size was set to 32. The second stage released the network parameters of the backbone network CSPDarkNet53, and the optimizer used Adam; this stage is the fine-tuning stage. The maximum learning rate was set to be smaller than the first stage, i.e., set to 0.0001, and the Batch size was based on the performance of the graphics card, since the entire network parameters needed to be adjusted, so it was set to 8. Each epoch was trained for 50 epochs.

The experiments in this paper were all performed on a high-performance platform equipped with NVIDIA GeForce RTX 3090 GPU, using 64-bit Windows 10 operating system, AMD 5900X CPU, and 16 GB of memory. The programming language used for training and testing was Python, the deep learning framework was Pytorch, and it took about 12 h to train the improved model.

3.3. Results and Analysis

To evaluate the algorithm proposed in this paper, we used the average precision (*AP*) of the trained model to detect underwater creatures as a reference for comparison. The accuracy rate (*P*) and recall rate (*R*) of object recognition are used as the coordinate axes, and the closed area enclosed by the corresponding curve is the *AP* value. *P*, *R*, and *AP* are calculated by Equations (5)–(7):

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$AP = \int_0^1 P(R)\mathrm{dR} \tag{7}$$

where TP refers to the positive samples predicted by the model to be positive, and TN refers to the negative samples predicted to be negative by the model. FP refers to the negative samples predicted by the model to be positive, and FN refers to the positive samples predicted to be negative by the model.

The primary metric is the AP@[IoU = 0.5 & 0.75]. When the IoU threshold between the bounding box and the ground truth is 0.75, it is denoted AP_{75}; when the IoU threshold is 0.5, it is denoted AP_{50}, which is the primary metric of another large object detection dataset, PASCAL VOC [24]. The original model and the model proposed in this paper were evaluated on the Brackish dataset using these two metrics (see Table 2).

Table 2. Results of the models evaluated on the Brackish categories and compared by AP_{75} and AP_{50} metrics.

Model	Categories	AP_{75}	AP_{50}
YOLOv2 fine-tuned [22]	Brackish	0.0984	0.3110
YOLOv3 fine-tuned [22]	Brackish	0.3893	0.8372
YOLOv4	Brackish	0.5293	0.9739
ours	Brackish	0.5796	0.9761

In order to further verify the validity of the model in the Brackish dataset, this paper also evaluated improvements in the detection accuracy of various organisms in the dataset after the model was improved, as shown in Figure 4. It can be seen that the improvement in the detection accuracy for the starfish class was significantly higher than that of other classes, proving that the proposed model is better for the detection of objects with unique shapes, and that which are almost stationary.

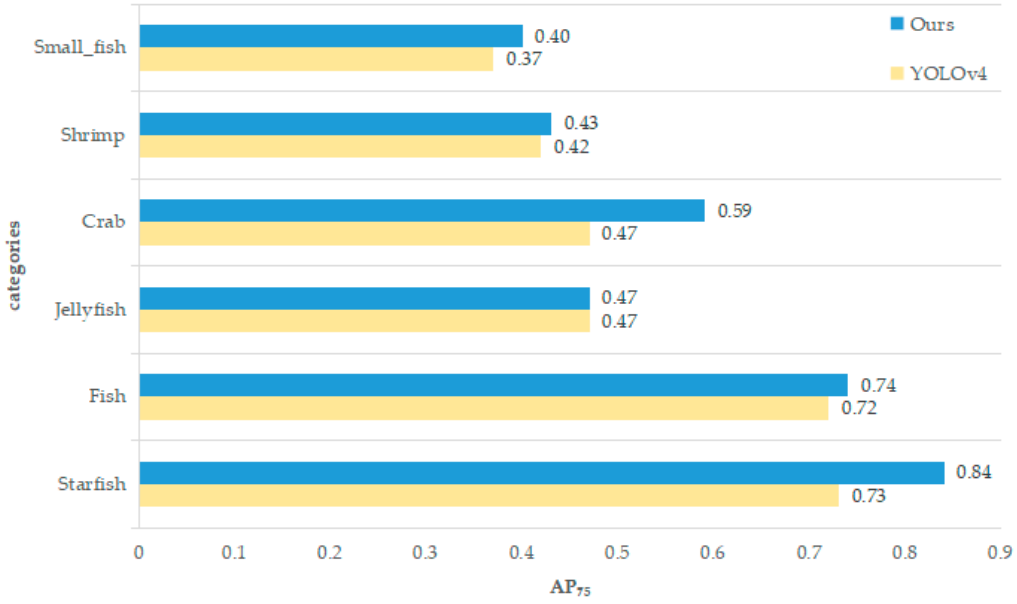

Figure 4. Comparison of detection accuracy AP_{75} results for each category.

The final results show that the improved model proposed in this paper has great potential in underwater biological detection. Compared with the original YOLOv4 algorithm, the improved model detection accuracy AP_{75} was improved by 5.03%. In addition, the detection rate of the improved network structure for video reached 15fps, and each image took about 0.049 s, which also proves its real-time performance. Figure 5 shows the results of the proposed method for the detection of six organisms under water.

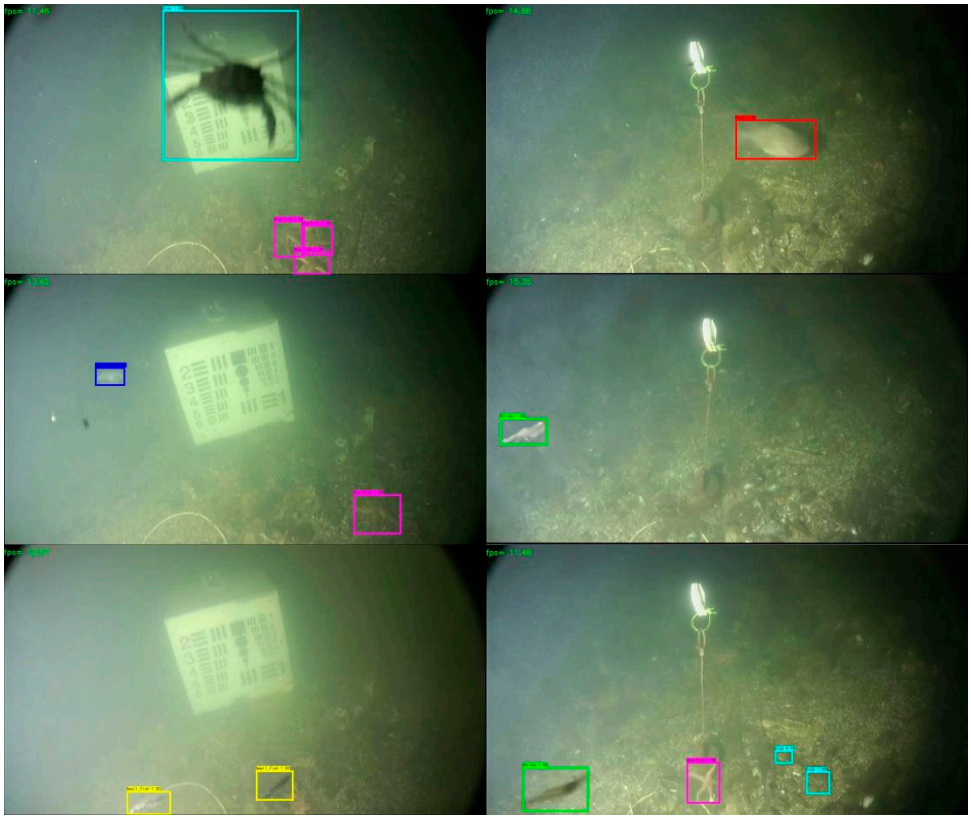

Figure 5. The detection results of our method in Brackish dataset. The detected crab in light blue box, fish in red box, jellyfish in dark blue box, shrimp in green box, small fish in yellow box and starfish in pink box.

4. Conclusions

Based on the excellent detection accuracy and speed advantages of YOLOv4, this paper adopted the object detection architecture for real-time underwater biological detection. At the same time, the network model was optimized, and a channel attention mechanism was embedded in the backbone feature extraction network. This enabled the algorithm to weigh the feature image channels of different types of underwater creatures in order to achieve more accurate automatic detection and classification, which is conducive to the monitoring of fishing activities and the effective protection of marine biological environments. The experimental results show the performance of the algorithm in terms of accuracy, robustness, and real-time performance. YOLOv4 combined with the channel attention mechanism achieved a higher accuracy while meeting the real-time requirements. In the future, further research will be conducted on deep neural network detection models with higher accuracy, faster speed, and lower computational cost.

Author Contributions: Conceptualization, A.L. and L.Y.; methodology, A.L.; software, A.L.; validation, L.Y. and S.T.; formal analysis, A.L.; investigation, A.L.; resources, L.Y.; data curation, A.L.; writing—original draft preparation, A.L.; writing—review and editing, L.Y.; visualization, A.L.; supervision, S.T.; project administration, A.L.; funding acquisition, L.Y. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable

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

References

1. Christensen, J.H.; Mogensen, L.V.; Galeazzi, R.; Andersen, J.C. Detection, localization and classification of fish and fish species in poor conditions using convolutional neural networks. In Proceedings of the 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Porto, Portugal, 6–9 November 2018.
2. Fouad, M.M.M.; Zawbaa, H.M.; El-Bendary, N.; Hassanien, A.E. Automatic nile tilapia fish classification approach using machine learning techniques. In Proceedings of the 13th International Conference on Hybrid Intelligent Systems (HIS 2013), Gammarth, Tunisia, 4–6 December 2013; pp. 173–178.
3. Shen, J.; Fan, T.; Tang, M.; Zhang, Q.; Sun, Z.; Huang, F. A biological hierarchical model based underwater moving object detection. *Comput. Math. Methods Med.* **2014**, *2014*, 609801. [CrossRef] [PubMed]
4. Wang, Y.; Fu, L.; Liu, K.; Nian, R.; Yan, T.; Lendasse, A. Stable Underwater Image Segmentation in High Quality via MRF Model. In Proceedings of the OCEANS 2015 - MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–4.
5. Shevchenko, V.; Eerola, T.; Kaarna, A. Fish detection from low visibility underwater videos. In Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1971–1976.
6. Liu, S.; Li, X.; Gao, M.; Cai, Y.; Nian, R.; Li, P.; Yan, T.; Lendasse, A. Embedded Online Fish Detection and Tracking System via YOLOv3 and Parallel Correlation Filter. In Proceedings of the OCEANS 2018 MTS/IEEE, Charleston, SC, USA, 22–25 October 2018.
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. *Ssd: Single shot multibox detector. European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 January 2016; pp. 779–788.
9. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 January 2017; pp. 7263–7271.
10. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804. 02767.
11. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 January 2017; pp. 2117–2125.
13. Faster, R.C.N.N. Towards real-time object detection with region proposal networks. *Adv. Neur. Inf. Process. Syst.* **2015**, *28*, 2969239–2969250.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neur. Inf. Process. Syst.* **2012**, 25. [CrossRef]
15. Sung, M.; Yu, S.C.; Girdhar, Y. Vision based real-time fish detection using convolutional neural network. In Proceedings of the OCEANS 2017-Aberdeen, Aberdeen, UK, 19–22 June 2017.
16. Wang, M.; Liu, M.; Zhang, F.; Lei, G.; Guo, J.; Wang, L. Fast classification and detection of fish images with YOLOv2. In Proceedings of the OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, Japan, 28–31 May 2018.
17. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
18. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. *Adv. Neur. Inf. Process. Syst.* **2014**, 27.
19. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neur. Inf. Process. Syst.* **2017**, 30.
21. 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, 18–22 January 2018; pp. 7132–7141.
22. Pedersen, M.; Bruslund Haurum, J.; Gade, R.; Moeslund, T.B. Detection of marine animals in a new underwater dataset with varying visibility. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 January 2019; pp. 18–26.
23. Bahnsen, C.H.; Møgelmose, A.; Moeslund, T.B. The aau multimodal annotation toolboxes: Annotating objects in images and videos. *arXiv* **2018**, arXiv:1809.03171.
24. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]