Improved YOLOv5 Algorithm for Real-Time Prediction of Fish Yield in All Cage Schools

Lei Wang 1,2, Ling-Zhi Chen 3, Bo Peng 4,* and Ying-Tien Lin 5,*

Abstract: Cage aquaculture makes it easier to produce high-quality aquatic products and allows full use of water resources. Therefore, cage aquaculture development is highly valued globally. However, the current digitalization level of cage aquaculture is low, and the farming risks are high. Research and development of digital management of the fish population in cages are greatly desired. Real-time monitoring of the activity status of the fish population and changes in the fish population size in cages is a pressing issue that needs to be addressed. This paper proposes an improved network called CC-YOLOv5 by embedding CoordConv modules to replace the original ConV convolution modules in the network, which improves the model’s generalization capability. By using two-stage detection logic, the target detection accuracy is enhanced to realize prediction of the number of fish populations. OpenCV is then used to measure fish tail lengths to establish growth curves of the fish and to predict the output of the fish population in the cages. Experimental results demonstrate that the mean average precision (mAP) of the improved algorithm increases by 14.9% compared to the original YOLOv5, reaching 95.4%. This research provides an effective solution to promote the intelligentization of cage aquaculture processes. It also lays the foundation for AI (Artificial Intelligence) applications in other aquaculture scenarios.

Keywords: cage; deep-sea aquaculture; intelligent management; internet of things technology; sensors

1. Introduction

The decline in marine fishery resources is becoming more and more serious, which leads to the increasing attention of countries around the world to the development of sustainable pelagic fisheries. How to quickly, efficiently, and accurately assess the amount of fishery resources is a key problem at present. With the progress of science and technology, high-frequency sonar has been widely used in fishery resources survey and evaluation. Compared with traditional resource survey methods, underwater acoustic detection technology has the advantages of being fast, efficient, and large scale, as well as causing no damage to the survey object [1,2]. There are many studies on the measurement of fish spatial distribution in some areas through scientific detection instruments. For example, sonar can be used to obtain images close to optical image quality in dark and turbid water [3]. Han et al. [4] generated acoustic images with better visual effects by preprocessing the image data collected by the device with image coordinate transformation, data interpolation, and image enhancement. Based on the acoustic image, Ibrahim et al. [5] proposed a fish-target-tracking method based on sonar by combining the background difference method. The acoustic image formed by the sonar is used to subtract the background to obtain the fish target.
Although the above method can achieve a certain effect of fish target detection, the sonar equipment is expensive, and the collected data need to be processed by supporting software on acoustic images. The processing procedure is cumbersome, and it is easy to be interfered with by other underwater moving objects (such as aquatic plants, non-fish organisms, etc.). The accuracy of fish detection in the natural environment has not been effectively improved. Based on the artificial feature method, the fish target in the image to be measured is extracted by using the pre-designed fish species feature information, including the fish body contour and texture information. In the environment of cultivating fry, according to Ojala et al. [6], currently in the resource survey, the most used method is to evaluate the resource by the echo integral method. However, this estimation is relatively rough, and it is impossible to check whether the echo is only fish, and the error is significant [7]. If the fish target is too small and the echo signal is weak, the statistical error for some sonar-attached software is very large [8].

Underwater target detection methods can be mainly divided into two categories: one is based on manual features, and the other is based on deep learning methods [9]. The method based on manual features mainly relies on manual design and extraction of visual features of the target, which are usually more intuitive, such as color, texture, shape, etc. Commonly used manual features include local binary patterns (LBPs) [10], scale invariant feature transformations (SIFTs) [11], directional gradient histograms (HOGs) [12], and so on. Based upon these manual features, various detectors can be designed to match and identify specific targets. Taking HOG as an example, it forms a feature descriptor by counting the gradient direction distribution in local areas of the image. Villon et al. [10] utilized HOG features and combined them with support vector machine (SVM) classifiers to detect coral reef fish in collected images. This method, to some extent, alleviates the problem of decreased detection accuracy caused by object occlusion and overlap in underwater images [13]. However, the design of manual features largely relies on algorithm designers’ understanding of relevant domain knowledge and cannot automatically learn features. At the same time, the process of manually designing features is relatively complex and requires a large amount of human and material resources. Therefore, underwater target detection methods based on manual features have good recognition performance for specific targets, but their generalization ability is weak. Another limitation of this method is its poor adaptability to scene changes and emerging targets.

Underwater object detection technology based on deep learning modeling has begun to rise and develop. Deep learning can automatically learn feature representations from large-scale annotated data, avoiding the complex task of manually designing features [14]. Currently, deep-learning-based object detection frameworks are mainly divided into the one-level method and two-level method. Representative first level methods include YOLO series, SSD, and so on [15]. These first level methods utilize convolutional neural networks (CNNs) to learn the overall position coordinates and categories of targets and can directly predict the positions and categories of all targets in the image. The advantage is faster detection speed, suitable for deployment on mobile or embedded platforms. However, the accuracy is relatively low, especially in the detection of small targets. The two-level method represents the Faster R-CNN [16] series of models, and the field of object detection has gone through a development process from CNN to faster R-CNN [17], R-FCN [18], and other networks. These networks have improved detection speed, but there are still certain drawbacks. Later, Zeng et al. [17] added Faster R-CNN to the Regional Proposal Network (RPN), a new network called Faster R-CNN AON [19], to improve the prediction speed and accuracy. On this basis, Song et al. improved Mask R-CNN by adding segmentation tasks, achieving multitask unification of detection and segmentation [20]. For underwater scenes, Mask R-CNN improved the accuracy of target detection in complex underwater environments. These networks are widely used in image object detection tasks. However, the above methods have high hardware requirements and are not suitable for lightweight or mobile devices. Therefore, the YOLO [21] series models based on regression have emerged.
YOLO adopted a single-stage detection framework to directly predict the probability of bounding boxes and categories, which is faster. From YOLO to YOLO9000 [22], then to YOLOv3 [23] and YOLOv4 [24,25], performance has been continuously improved and become the preferred choice for practical applications. In addition to the YOLO series model, the single-stage detection network SSD [26] also performs well. These lightweight models enable object detection to be deployed on mobile or embedded devices. Chen et al. [27] also specifically proposed the YOLOv4-UW model, using the adversarial occlusion methods to enhance the model’s robustness to occlusion. Among them, the fry is detected based on the method of digital image feature extraction and processing, and the shape and regional position of the fry are extracted by the Chain Code and Corners–Harris Stephen algorithm, respectively. In these experiments, the image dataset was divided into two parts. One part of the dataset contained only a single fish. The artificial neural network was used to train the feature extraction model of the fish group, and then the trained model was adopted to detect the mutual occlusion of the fish group in the other part of the dataset. However, the above two methods limit the size of the fish and the environment in which they live and cannot be extended to complex wild scenes. Spampinato et al. [28] proposed a method of fish detection, tracking, and counting based on underwater low-quality video. This method used a dynamic background updating algorithm and the adaptive Gaussian mixture model to realize fish detection and counting. It has a good detection effect on slow-moving objects, but the false detection rate of non-targets (such as seaweed, background rock, etc.) is very high.

Recently, Wang [29] used an improved YOLOV5s network to detect fish with abnormal behavior. By combining multi-level features and adding feature mapping, the detection accuracy was improved, reaching 76.7%. Muksit [30] offered two large-scale real-world marine environment fish detection datasets: DeepFish and OzFish. These two datasets contain high-definition fish images from various marine habitats, with complex backgrounds and lighting changes. Two improved fish detection models were developed based on YOLOv3: YOLO-Fish-1 and YOLO-Fish-2. YOLO-Fish-1 increased the detection accuracy of small fish by optimizing the up-sampling stride. YOLO-Fish-2 further enhanced the model’s robustness by adding a spatial pyramid pooling module on top of YOLO-Fish-1. The models were trained and tested on the two datasets, respectively. Results showed that the YOLO-Fish models outperformed the original YOLOv3, especially in the detection accuracy of small fish and complex scenes. The average precision reached 76.56% for YOLO-Fish-1 and 75.7% for YOLO-Fish-2. Wang et al. [30] incorporated the CBAM attention mechanism and the SPPFCSPC module based on the YOLOv7 model. Compared with the original YOLOv7, ACFP-YOLO improved the mAP (mean average precision) by 1.64% while only losing 2.62% of the detection speed, maintaining relatively good real-time performance while improving the detection accuracy. However, the tracking accuracy from previous studies can still be improved.

The main goal of the present research is to achieve real-time automatic estimation of fish output with better detection accuracy during cage aquaculture to assist aquaculture personnel in remote management and real-time control of feeding amounts, greatly reducing labor costs. Considering the strengths and weaknesses of previous studies, this paper used an improved YOLOv5 algorithm to detect dead fish in different scenarios by subtracting the number of dead fish from the total number of fries previously known. In addition, underwater cameras are installed to take pictures of fish shoals. Using deep learning and OpenCV can improve the accuracy of image feature extraction and fish body length measurement in fish shoal scenarios. According to the body length–body weight formula and the number of fish shoals, the output of the fish population in the cages can be predicted.
2. Materials and Methods

2.1. Improved YOLOv5 Algorithm Detection Principle

Object detection algorithms based on deep learning are mainly divided into two categories: two-stage detector and one-stage detector. Among them, the common two-stage detectors include Faster R-CNN. This kind of algorithm usually has high detection accuracy, but the speed is slow, while the single-stage detectors include SSD, YOLO series, and so on. In the application scenario of power grid timeliness, in order to capture images more accurately, this paper uses YOLOv5 algorithm v7.0 as the detection model and uses its own labeled dataset to train it to obtain a more accurate model. The YOLOv5 algorithm provides five different network models, which are YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. YOLOv5n is a new model. Its network depth is deeper than YOLOv5s, but the number of channels has been reduced from 0.50 to 0.25. As a result, the number of iterative repetitions and convolutional kernel channels are both affected, leading to a decrease in detection accuracy. After comprehensive evaluation, we selected the YOLOv5s network as the research subject. The architecture of YOLOv5s includes three core elements: output, middle layer, and top layer, which work together to provide accurate information and make effective predictions on the feature data. The network parameters of these models are shown in Table 1.

Table 1. Network Model Parameters.

<table>
<thead>
<tr>
<th>Network Model</th>
<th>Network Depth</th>
<th>Network Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5n</td>
<td>0.33</td>
<td>0.25</td>
</tr>
<tr>
<td>YOLOv5s</td>
<td>0.33</td>
<td>0.5</td>
</tr>
<tr>
<td>YOLOv5m</td>
<td>0.67</td>
<td>0.75</td>
</tr>
<tr>
<td>YOLOv5l</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>YOLOv5x</td>
<td>1.33</td>
<td>1.25</td>
</tr>
</tbody>
</table>

2.2. Improved CoordConv-YOLOv5 for Technical Solution Innovation

Using CoordConv as shown in Figure 1, it does not bring too much of an increase in computation and does not completely abandon translation invariance, but allows the network to learn moderate translation invariance and dependence.

![Figure 1. CoordConv module structure. X_c and Y_c are coordinate channels, representing the x and y coordinates of the original input, are used for traditional convolution to perceive the spatial information of the feature map during the convolution process.](image-url)
The CoordConv YoLov5 structure diagram is provided in Figure 2. Letterbox adaptive image scaling technology is widely used in the input terminal (Input). It can reduce the original image to a certain size as needed and add gray borders to the remaining area to maintain the aspect ratio of the original image, while reducing the distortion of the image, thereby improving the training efficiency of the model. The standard size is 640 × 640. Backbone’s infrastructure includes three key elements: CoordConv, BatchNorm2d, and SiLU, which together form the central architecture of C3. The C3 architecture consists of two independent components, one of which is represented by a standard convolution (Bottleneck) and the other by a special convolution (bottleneck). In the v6.0 version, the SPPF structure replaces the spatial pyramid pooling (SPP) structure in the v5.0 version and is placed at the end of the backbone network to improve detection performance. The SPPF structure can effectively reduce the number of channels of the input features and divide them into two independent parts. The upper part uses three maximum pooling layers of 5 × 5, while the lower part does not need to be pooled. Finally, the feature maps of the two parts can be spliced to form a CoordConv so as to achieve effective complementarity between different feature maps.

Figure 2. CoordConv YoLov5 Structure Diagram.

Head’s predictive ability depends on the path aggregation network (PANet). It takes FPN as the premise and effectively fuses multiple features to convert low-dimensional data into more complex structures and provide more comprehensive feature representation. The Head model is used to estimate the information in the dataset and to create the corresponding threshold range and classification.

2.3. Length Estimation Algorithm

Firstly, we need to load the trained YoLov5 model, define a function for calculating the distance, and firstly obtain the focal length of the camera from the object with known distance. The focal length is fixed so that it can be used to calculate the distance between the fish and the camera.

\[ D = \frac{(W \times f)}{w} \] (1)
where D is the distance between the object and the camera; W is the actual width of the object; f is the focal length of the camera; and w is the width of the object in the image. The program first reads the image and then preprocesses the image to more easily detect objects in the image. It then uses OpenCV’s contour detection function to find contours in the image and iterates through each contour. For each contour, the program calculates the bounding box of the contour, and then draws the bounding box on the image. Next, it calculates the size of the object and adds text to the image displaying the object’s size. Finally, the program displays the processed image using OpenCV’s image display functionality and saves the processed image to a new file using OpenCV’s image save functionality. In actual applications, we can optimize the algorithm to improve performance and accuracy. Here are some suggestions: Camera calibration: In the current implementation, we directly used the focal length as a parameter. To obtain more accurate results, camera calibration can be performed to obtain more camera intrinsic parameters, such as distortion coefficients. This will help to improve the accuracy of distance estimation. OpenCV’s cv2.calibrateCamera() function can be used to implement camera calibration. Multi-frame fusion: To improve the stability of distance measurement, multi-frame fusion technology is adopted. By collecting detection results from consecutive multi-frame images, the average or weighted average value of the fish distance is calculated. This can reduce errors and improve stability. Adaptive threshold adjustment: Detection results may be affected under different scenes and lighting conditions. Based on image brightness, contrast, and other features, the confidence threshold is adaptively adjusted to improve detection accuracy. Result smoothing: There may be noise or mutations in distance estimation. We can apply some filtering algorithms (such as Kalman filters or moving average filters) to smooth the results and improve estimation stability. The body length of the fish is saved in a csv file. The fish growth curve is outputted through the program. Figure 3 provides the average fish growth curve and underwater camera setup. The underwater camera is fixed at the corner of the cage, a rectangular acrylic channel is set, the distance between the camera and the rectangular channel is fixed, and then the perspective is fixed. The length of the fish can be measured.

Figure 3. Average Fish Growth Curve and Underwater Camera Setup.

The description of the data collection device and database application model is as follows: The perception and execution layer need to consider the main data: the surface monitoring data and the underwater fish school length data, respectively. The pictures taken by the camera are stored as current time .csv files after processing. The transmission layer considers: the data collected by the perception layer can be smoothly and completely transferred to the host. The application layer is used to implement various functional modules of the complete system. The display interface is the boundary between the system and users. Real-time monitoring and historical data query are performed through mobile terminals and PC terminals. The device logical architecture is shown in Figure 4.
The description of the data collection device and database application model is as follows: The perception and execution layer need to consider the main data: the surface monitoring data and the underwater fish school length data, respectively. The pictures taken by the camera are stored as current time .csv files after processing. The transmission layer considers: the data collected by the perception layer can be smoothly and completely transferred to the host. The application layer is used to implement various functional modules of the complete system. The display interface is the boundary between the system and users. Real-time monitoring and historical data query are performed through mobile terminals and PC terminals. The device logical architecture is shown in Figure 4.

2.4. Real-Time Prediction Algorithm of Fish Yield

In fish farming, cage farming accounts for more than 60% of fish farming. Therefore, there is an urgent and common demand for farmers to understand the living status and growth curve of fish and estimate the yield of fish. However, the data error measured by both ultrasonic and optical detection is large. In this paper, it is combined with the actual breeding experience through reverse logic to predict the number of fish, so as to realize the prediction of fish production.

Purchasing fry is a critical link in aquaculture as choosing healthy, quality fry directly impacts growth rate, survival rate, and yield in later stages. Since fry are extremely small in size, with purchase quantities reaching over tens of thousands per batch, a scientific and reasonable counting method must be adopted to ensure counting accuracy and improve work efficiency. Common fry counting methods include the following: (1) Direct counting, tallying one by one, which is accurate but time-consuming; (2) Weighing method, converting weight to quantity, which is faster but results in higher error; (3) Sampling counting...
method, randomly sampling from the fry population for counting, and then extrapolating the total, simple operation and error within acceptable range. Therefore, fishermen will ensure accurate and efficient counting when purchasing fry. After purchasing and breeding fry, fishermen will obtain a total fish population number with relatively small error. In daily farming practices, without fish escaping, by subtracting the number of dead fish, the final number obtained is the total fish population.

The number of dead fish is measured by the improved YoLov5 every minute through the picture of the water surface of the cage taken by the camera. On the basis of the detected dead fish, different types of dead fish are detected by YoLov5. After 6 pm, the max-imam value of a certain type of dead fish today is obtained. The total number of dead fish is estimated by subtracting the number of dead fish from the total number of fries. After 6 pm, the number of dead fish was cleared and recounted. The body length of fish was measured by OpenCV, and the weight of fish was obtained according to the formula of body length and weight, and the growth curve was drawn to predict the overall yield of fish. The flow chart of the real-time prediction algorithm is provided in Figure 5.

![Figure 5. Real-time prediction algorithm for fish school yield.](image)

2.5. Dataset Collection and Processing

The original image dataset used in this paper is derived from on-site breeding and web crawlers. After sorting, the pictures with dead fish are classified and integrated, including three common situations: cage dead fish, common cage dead fish, and lake dead fish. Compared with the existing open-source datasets, this dataset has the following characteristics: rich scene, diverse background environment, and diverse data distribution. For the application of target detection, this paper proposes a multi-scale domain adaptive method. The commonly used labeling open-source software was used to label the dead fish, golden pomfret dead fish, and perch dead fish in the image one by one, and a total of about 30,260 targets were labeled. The annotation content includes the coordinates of the rectangular bounding box and stores it as an XML text file for training and testing the YoLov5 model. Meanwhile, the corresponding program is written, and the constructed dataset is divided into training set, verification set, and test set according to the ratio of 8:1:1. Among them, the training set contains 24220 images, while the validation set and the test set contain 3020 images, respectively, as listed in Table 2.
Table 2. Experimental Data Division.

<table>
<thead>
<tr>
<th></th>
<th>Dead Fish in the Cage</th>
<th>Dead Fish in Ordinary Cage</th>
<th>Dead Fish on the Lake Surface</th>
<th>Grouper Dead Fish</th>
<th>Golden Pomfret Dead Fish</th>
<th>Perch Dead Fish</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>7050</td>
<td>1270</td>
<td>8650</td>
<td>2350</td>
<td>3640</td>
<td>1260</td>
<td>24,220</td>
</tr>
<tr>
<td>Test set</td>
<td>880</td>
<td>160</td>
<td>1070</td>
<td>300</td>
<td>450</td>
<td>160</td>
<td>3020</td>
</tr>
<tr>
<td>Validation set</td>
<td>880</td>
<td>160</td>
<td>1070</td>
<td>300</td>
<td>450</td>
<td>160</td>
<td>3020</td>
</tr>
<tr>
<td>Total</td>
<td>8810</td>
<td>1590</td>
<td>10,790</td>
<td>2950</td>
<td>4540</td>
<td>1580</td>
<td>30,260</td>
</tr>
</tbody>
</table>

By using the dead fish dataset, a series of experiments were carried out to evaluate the effectiveness of our proposed method. These experiments are divided as follows: (1) applying them to known datasets to evaluate their accuracy; (2) we also compare their results with known target detection techniques to assess superiority; (3) through the ablation test, we can further explore the feasibility of our proposed improved scheme in the detection of dead fish.

2.6. Experimental Environment and Parameter Settings

The experimental environment of this paper is Ubuntu18.04LTS, Ubuntu18.04LTS, intel (R) Core (TM) i9-10900K, RTX1060Ti GPU, 16G RAM, Python3.7 programming language CUDA10.2. Acceleration library, Pytorchv1.7.0 deep learning framework. In the training stage, warmup training preheating and cosine attenuation strategy are utilized to dynamically adjust the learning rate. The experimental hyperparameters are set as provided in Table 3.

Table 3. Experimental Hyperparameter Settings.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Batch training size</td>
<td>32</td>
</tr>
<tr>
<td>Momentum</td>
<td>0.937</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.0005</td>
</tr>
<tr>
<td>Epoch</td>
<td>200</td>
</tr>
<tr>
<td>Image size</td>
<td>640 × 640</td>
</tr>
</tbody>
</table>

2.7. Evaluating Indicator

This paper discusses how to evaluate the effect of the network model through four main indicators: prediction effect (Precision), recall rate (Recall), parallel ratio (IoU), and mean average precision (mAP). The mAP represents the average accuracy of each experiment, which can be measured by the PR curve and the total area of their surrounding areas. The mAP is the sum of the accuracy of each species. The equations are listed below:

\[
P = \frac{TP}{TP + FN} \times 100\% \quad (2)
\]

\[
R = \frac{TP}{TP + FN} \times 100\% \quad (3)
\]

\[
AP = \sum_{i=1}^{n-1} (R_{i+1}R_i) P_{\text{interp}}(R_{i+1}) \quad (4)
\]

\[
P_{\text{interp}}(R_{i+1}) = \max_{R \geq R_{i+1}} \left( P(R) \right) \quad (5)
\]

where TP (true positive) represents the number of samples correctly classified by the network, FP (false positive) denotes the number of incorrectly detected samples, and FN (false negative) means the number of missed detected samples. P(R) represents the value of accuracy P when the recall rate is R, and S denotes the total number of categories.
3. Results

3.1. Verification

The loss function is an important parameter used to evaluate the prediction accuracy of the deep learning model. It plays an important role in analyzing and judging the advantages and disadvantages of the training process, the convergence degree of the model, and whether it is over-fitting. When using the PyTorch framework, the loss function can be regarded as a layer in the model definition, but in practical use, it is more regarded as a function in the forward propagation process. It can be seen from Figure 6 that in the process of training and verification, we conducted a comparative analysis of the loss function value, accuracy, recall rate, mAP, and other verification indicators of the YoLov5 model and the improved model YoLov5. During the training process of 200 epochs, the loss function values of both models showed a downward trend and eventually tended to be flat. However, the loss function value of the improved YoLov5 is usually lower than that of the original model of YoLov5. YoLov5 has a lower initial loss value because of the loading of pre-training parameters.

![Figure 6. Comparison of loss function values, accuracy, recall, and mAP between YoLov5 model and improved model YoLov5.](image)

It can be seen from Table 4 that the performance of the improved YoLov5 model has been improved. Overall, compared with YoLov5, the improved YoLov5 improves the accuracy, recall, and mAP by 3.5%, 5.5%, and 14.9%, respectively. Through comparison, it is found that the improved YoLov5 has a certain performance improvement in the recognition reliability of each category, which further proves the effectiveness of the CoordConv module in extracting diversity features. We used our self-built dead fish dataset to evaluate the improved YoLov5 algorithm in this paper. The experimental results show that our proposed algorithm has more advantages than common object detection algorithms.

<table>
<thead>
<tr>
<th></th>
<th>YoLov5</th>
<th>Improving YoLov5</th>
<th>Accuracy Rate</th>
<th>Recall</th>
<th>mAP0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>√</td>
<td>--</td>
<td>95.7</td>
<td>90.9</td>
<td>79.3</td>
</tr>
<tr>
<td></td>
<td>--</td>
<td>√</td>
<td>99.2</td>
<td>96.4</td>
<td>95.4</td>
</tr>
</tbody>
</table>

Table 4. Performance comparison between original YoLov5 and improved YoLov5 on the dataset.
The comparative experiments show that the model proposed in this paper can accurately perform the target detection task in the complex environment of the cage. Figure 7 shows the corresponding test results.

Table 4. Performance comparison between original YoLov5 and improved YoLov5 on the dataset.

<table>
<thead>
<tr>
<th>Original YoLov5</th>
<th>Improving YoLov5</th>
<th>Accuracy Rate</th>
<th>Recall</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>95.7</td>
<td>90.9</td>
<td>79.3</td>
</tr>
<tr>
<td>Improved YoLov5</td>
<td></td>
<td>99.2</td>
<td>96.4</td>
<td>95.4</td>
</tr>
</tbody>
</table>

Figure 7. First level dead fish detection algorithm results.

Figure 8a is the original image, and Figure 8b is the detection result of the improved YoLov5 model again under the logic of detecting the dead fish model above, and the verification index of the dead fish classification model can achieve relatively ideal results. The reason for this is that the difficulty of detection in the range of box selection is lower than that of the multi-type detection model, and the performance of the model is relatively good. The dead fish dataset includes three types of datasets, which have better similarity due to external light interference. In addition, the number of instances of each category in the entire dataset is also uneven, resulting in limited performance during model training. In the training process of the classification counting model, the CoordConv module can improve the ability of the backbone to extract the effective feature information of the dead fish target in the target area. Therefore, the prediction accuracy of the constructed classification counting model is better than that of the improved model, which ensures real-time performance and improves the detection accuracy.
YOLOv3 also have the problem of low accuracy. COC-YoLov5 shows a good balance between accuracy and processing efficiency, which verifies the effectiveness of the proposed algorithm.

The next step is to further optimize the model in order to achieve the same detection efficiency as YoLov5 and maintain the advantage of accuracy (as shown in Table 5).

### Table 5. Comparison of experimental results under different algorithms.

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>mAP</th>
<th>FPS (Frames per Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN</td>
<td>70.93</td>
<td>15</td>
</tr>
<tr>
<td>SSD</td>
<td>66.47</td>
<td>36</td>
</tr>
<tr>
<td>YOLOv3</td>
<td>75.56</td>
<td>38</td>
</tr>
<tr>
<td>YOLOv4</td>
<td>76.21</td>
<td>29</td>
</tr>
<tr>
<td>YoLov5</td>
<td>79.3</td>
<td>126</td>
</tr>
<tr>
<td>CoordConv-YoLov5</td>
<td>95.4</td>
<td>121</td>
</tr>
</tbody>
</table>

The method CoordConv-YoLov5S proposed in this paper is visually compared with the detection effect of the original various algorithms on the dataset, as shown in Figure 9. Because the target of dead fish in the image is small and similar, and the background is also complex, the SSD detection method performs poorly in dead fish detection and cannot accurately detect the type of dead fish. Several other methods can detect dead fish and dead fish species relatively accurately. In the same case, the method proposed in this paper can accurately distinguish the target and detect it all, which intuitively shows the advantages of the method.
Figure 9. Comparison of Algorithm Effects.

Based upon the statistical results of the fish number and body length, the output of the fish population can be obtained through the algorithm as shown in Figure 10.

Figure 10. Dish stocks corresponding to days and production.

4. Discussion

Cage aquaculture is an important direction for the development of global marine fisheries, but the current digitalization level of cage aquaculture is relatively low and unable to monitor the status of fish populations in cages in real time. To realize intelligent management of caged fish populations, AI video monitoring technology is adopted to monitor the number, body length, and output of fish populations in cages; determine fish growth curves; and optimize farming management.

In terms of acoustic detection, hydroacoustic detection is an important means of fishery resource surveys. Some of the current underwater acoustic instruments are independently developed by universities and research institutes, consisting of receiving, transmitting, and analyzing ends. The EY60 series has a wide frequency range and can...
be equipped with transducers of different frequency bands to achieve resolution from plankton to large fish. This can accurately reflect the distribution and activity of fish shoals. The application of these devices has greatly improved the alignment of global hydroacoustic technology with international advanced standards, providing strong support for the monitoring and assessment of fishery resources. The next step is to further strengthen the data analysis and interpretation capabilities of hydroacoustic systems, establish fishery resource databases, and realize the digitalization and intellectualization of resource surveys [31,32]. In the direction of optical detection, due to the complex underwater environment (such as turbid water, uneven illumination, ocean current interference, etc. [33]) and insufficient datasets, the model training of underwater target detection is much more difficult. From the perspective of the development trend in cage monitoring technology, optical monitoring has the advantage of intuitive imaging, but it has many disadvantages: the light wave attenuates rapidly in seawater, the detection distance is short, and it is only 1~2 m in the turbid sea area, which cannot meet the requirements of the whole cage detection. The artificial light source used at night not only affects the growth of fish, but also increases the power consumption.

In future research and implementation, the quality and quantity of algorithm training datasets are crucial for ensuring detection accuracy. Therefore, a large amount of image data from different aquaculture environments should be collected, with strict quality control of data annotation, and data augmentation methods should be used to expand the dataset scale. At the same time, choose appropriate pre-trained models and fine-tune using custom datasets to improve model adaptability to aquaculture scenarios. In algorithm design, techniques like multi-scale training, CutMix, etc., can be considered to enhance model generalization capability, and network structures can be adjusted to prevent overfitting. Image preprocessing optimization and camera calibration are also important, as environmental changes pose another huge challenge for algorithms. Environmental parameter mapping models can be constructed to improve algorithm adaptability. For system implementation, processes need to be simplified, stability improved, and continuous improvements made to adapt to complex real environments. More effective image enhancement attention modules and larger models may be future research directions for improving target detection accuracy, as well as model compression methods to reduce model size for deployment on mobile and embedded devices.

Compared with similar papers, the main advantages of this paper are application scenarios that are more clearly focused, addressing needs more targeted. It also directly aimed to predict the output and size of cage fish populations, which are highly practical applications. By adopting collaborative optimization in algorithm technology, model parameters can be dynamically adjusted according to different tasks for higher algorithm performance, which is cost-effective economically, facilitating promotion and application. Disadvantages include the following: the dataset is singular, mainly annotated images targeting cage environments, and it has limited generalization capability. In addition, the model is insufficient in terms of environmental complexity, and it lacks noise data and multi-source information fusion, etc. The prediction system is simple and lacks production environment testing. Follow-up efforts will expand dataset scope, construct hybrid sensor monitoring platforms, etc., enabling the system to have broader application prospects in engineering practice.

5. Conclusions

This paper used an improved YoLov5 algorithm for real-time prediction of fish yield in all cage schools. The innovations of the paper are mainly found in three aspects: (1) This paper proposes an improved YoLov5 target detection algorithm by embedding CoordConv modules and adopting adaptive image scaling methods to improve detection effects in complex cage environments. Experimental results show that the mAP of the improved algorithm is 14.9% higher than the original YoLov5 with high detection accuracy. (2) A fish output prediction system is designed. First, the improved algorithm detects the
number of dead fish, then OpenCV measures fish tail lengths to establish growth curves, combined with the initial number of fries minus deaths to finally predict the output of the fish population in cages. This system enables real-time monitoring of fish in cages. (3) This paper first uses the improved algorithm to detect the number of dead fish, then uses body length to calculate body weight, combined with the initial number of fries minus deaths, and predicts fish output based on growth curve models. This system realizes real-time monitoring of caged fish to support aquaculture decisions. The improved YoLov5 algorithm in this research conducts intelligent monitoring of the number, body length, and output of caged fish, representing innovative work combined with practical applications, cost-effective and of great significance for the promotion and development of cage aquaculture. Follow-up efforts will continue to optimize the algorithms and system to expand the application scope. Furthermore, the quality and quantity of algorithm training datasets are crucial for ensuring detection accuracy. Therefore, in the future, a large amount of image data from different aquaculture environments should be collected, with strict quality control of data annotation, and data augmentation methods should be used to expand the dataset scale. In the algorithm design, techniques like multi-scale training, CutMix, etc., can be considered to enhance model generalization capability, and network structures can be adjusted to prevent overfitting. More effective image enhancement attention modules and larger models may be future research directions for improving target detection accuracy, as well as model compression methods to reduce model size for deployment on mobile and embedded devices.

Author Contributions: Conceptualization: L.W. and Y.-T.L.; methodology: B.P. and Y.-T.L.; data analysis, L.W. and L.-Z.C.; software, L.W. and B.P.; writing—original draft preparation, L.W. and B.P.; writing—review and editing, Y.-T.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by Research on Key Technologies of Comprehensive Marine Observation/Monitoring Data Integration and Shared Services in the South China Sea (SML2021SP102), Marine Economy Development Project of Guangdong Province (GDNRC[2023]44), and the Science Foundation of Donghai Laboratory (Grant No. DH-2022KF01016).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are available from the authors upon request.

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

References


33. Cai, L.; Sun, Q.; Xu, T.; Ma, Y.; Chen, Z. Multi-AUV Collaborative Target Recognition Based on Transfer-Reinforcement Learning. IEEE Access 2020, 8, 39273–39284. [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.