fishes

MDPI

Article

Method for Non-Contact Measuring the Weight of Sturgeon in Intensive Aquaculture

Junjie Hu [1,2], Kai Lin [3,*], Shiyu Zhang [1,*], Rui Zhang [1,2], Hongsong Li [4] and Runqiu Xia [1]

[1] School of Instrument Science and Opto-Electronics Engineering, Beijing Information Science and Technology University, Beijing 100192, China; 2022020142@bistu.edu.cn (J.H.); 2024020340@bistu.edu.cn (R.Z.); xrq@bistu.edu.cn (R.X.)
[2] Fisheries Science Institute, Beijing Academy of Agriculture and Forestry Sciences & National Engineering Research Center for Freshwaters, Beijing 100068, China
[3] Key Laboratory of Equipment and Informatization in Environment Controlled Agriculture, Ministry of Agriculture and Rural Affairs, China
[4] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China; lihongsong@bit.edu.cn
* Correspondence: linkai@baafs.net.cn (K.L.); zhangshiyu@bistu.edu.cn (S.Z.)

Abstract: Weight information plays a pivotal role in sturgeon breeding and production management. However, manual measurement is time consuming and labor intensive due to the immense size of the sturgeon. Due to the unique body shape of the sturgeon, traditional image segmentation algorithms struggle to extract the necessary features from sturgeon images, which makes them unsuitable for this particular species. Moreover, accurately measuring weight in an occlusion environment is difficult. To address these challenges, an improved YOLOv5s model with a context augmentation module, focal-efficient intersection over union, and soft non-maximum suppression was proposed in this paper. To validate the model's feasibility, the improved YOLOv5s model was first pre-trained using the sturgeon dataset, followed by further training on the occlusion dataset for segmentation tasks. Based on the phenotypic data obtained from the improved model, a multilayer perceptron method was used to estimate the sturgeon's weight accurately. Experimental results demonstrated that the average precision of the improved YOLOv5s model reached 89.80% under occlusion conditions, and the correlation coefficient of noncontact weight measurement results reached 89.80%. The experimental results showed that the improved algorithm effectively performs segmentation of sturgeon in occlusion conditions and can accurately estimate the mass.

Keywords: sturgeon; weight; occlusion; in situ measurement

Key Contribution: This paper introduces an enhanced YOLOv5s model incorporating a context augmentation module, Focal-EIoU, and Soft-NMS, achieving 89.80% precision in segmenting sturgeons and estimating their weight under occlusion conditions. This development represents a notable advancement in the management of sturgeon breeding and production.

Citation: Hu, J.; Lin, K.; Zhang, S.; Zhang, R.; Li, H.; Xia, R. Method for Non-Contact Measuring the Weight of Sturgeon in Intensive Aquaculture. *Fishes* **2024**, *9*, 458. https://doi.org/10.3390/fishes9110458

Academic Editor: Vincent Kerzérho

Received: 10 October 2024
Revised: 5 November 2024
Accepted: 8 November 2024
Published: 10 November 2024

1. Introduction

China is the world's largest producer and trader of sturgeons from aquaculture and production operations, and their production was 121,875,000 tons in 2021, which accounted for about 85% of global production [1]. Sturgeons are a group of large fish known for their high economic value and are widely utilized in the food processing and catering industries due to their nutritious meat and valuable caviar [2]. The weight of sturgeons is a crucial biological characteristic in aquaculture and an important trait in breeding programs. Additionally, it informs decisions on feeding amounts, antibiotic dosages, optimal grading and harvesting times, and water environment monitoring [3]. The conventional approach for measuring sturgeon weight involves manual weighing, which is a time-consuming,

labor-intensive, and costly method with limited result consistency. The traditional mechanical motion weighing method may injure the sturgeon [4]. Therefore, a noncontact sturgeon weight measurement strategy is urgently needed [5]. Noncontact automatic estimation methods of fish mass usually involve three steps: Fish images are preprocessed; fish features are extracted from images and feature values such as body length are calculated; and the feature values are fitted to construct a prediction model to estimate the fish mass [6].

In recent years, noncontact methods for weight estimation based on vision systems have become popular in fisheries [7]. For example, Nicolo Tonachella and colleagues employed the YOLOv4 algorithm to extract phenotypic data from fish, used fitting methods to establish the relationship between fish length and weight, and successfully estimated the fish weight [8]. John Reidar Mathiassen et al. utilized a 3D machine vision system combined with a fitting approach, incorporating 2D and 3D features, to achieve more accurate high-speed weight estimation of whole herring [9]. Lu Zhang et al. employed the GrabCut algorithm for image segmentation and the Back Propagation Neural Network for fish mass estimation and improved accuracy in fish weight prediction [2]. Yinfeng Hao et al. applied the active contour method for tail fin removal and utilized partial least squares to improve fish mass estimation accuracy [10]. The researchers mentioned above conducted noncontact measurements of fish weight, but they did not consider the substantial occlusion of fish in actual aquaculture environments. However, other researchers focused on occlusion in noncontact fish weight measurement. Xiaoning Yu and coworkers proposed an attention-based fully convolutional instance segmentation network for instance segmentation of fish contours and integrated it with a regression fitting approach to estimate fish weight in aquaculture environments accurately [11]. Tianye Zhang and his team successfully applied a DL-YOLO algorithm combined with machine learning techniques to solve fish occlusion issues and enabled precise, automated biomass estimation for free-swimming fish [12]. However, the accuracy of the weight prediction methods used in the said studies, which rely on regression fitting or machine learning techniques, can still be improved. Although Yunhan Yang and colleagues applied the latest deep learning approaches for weight prediction, they did not address the issue of occlusion [13].

Although vision-based weight inspection methods are widely used, adult sturgeons are large with spindle-shaped bodies, flat ventral, pentagonal trunks, and asymmetrical caudal fins [14], which makes the weight measurement method suitable for others not appropriate for sturgeons. Moreover, the segmentation of sturgeons in culture tanks often faces the problem of occlusion, which leads to a decline in the accuracy of segmentation methods. In response to these challenges, a sturgeon dataset for weight measurement was established in this paper. Based on this dataset, an improved YOLOv5 model was proposed to enhance the segmentation accuracy of cultured sturgeons. This model employed context feature information fusion to improve the accuracy of sturgeon target detection by effectively utilizing input features through enhanced techniques. Ultimately, based on the phenotypic data of sturgeons obtained from the improved YOLOv5s segmentation, deep learning techniques were employed to develop the weight measurement model.

2. Materials and Methods

2.1. Overall Framework

Figure 1 illustrates an overview of the proposed weight estimation framework for the sturgeon, segmented into three main components: dataset, segmentation, and calculation. Firstly, sturgeon images are acquired using an image acquisition device and subsequently partitioned into training sets, validation sets, and test sets. Secondly, the segmentation phase involves image preprocessing followed by sturgeon feature extraction using an enhanced YOLOv5s model with the context augmentation module (CAM). Additionally, optimization techniques such as soft non-maximum suppression (Soft-NMS) and focal-efficient intersection over union loss (Focal-EIoU) are employed to obtain precise segmentation outputs. Finally, at the calculation stage, three methods are used to estimate sturgeon weight from extracted features. The first involves multivariate function fitting

to capture weight relationships. The second uses machine learning algorithms—support vector machine (SVM), random forest (RF), and k-nearest neighbors (KNN)—to offer a broader comparison. Lastly, a deep learning approach with a multilayer perceptron (MLP) is employed for enhanced predictive accuracy.

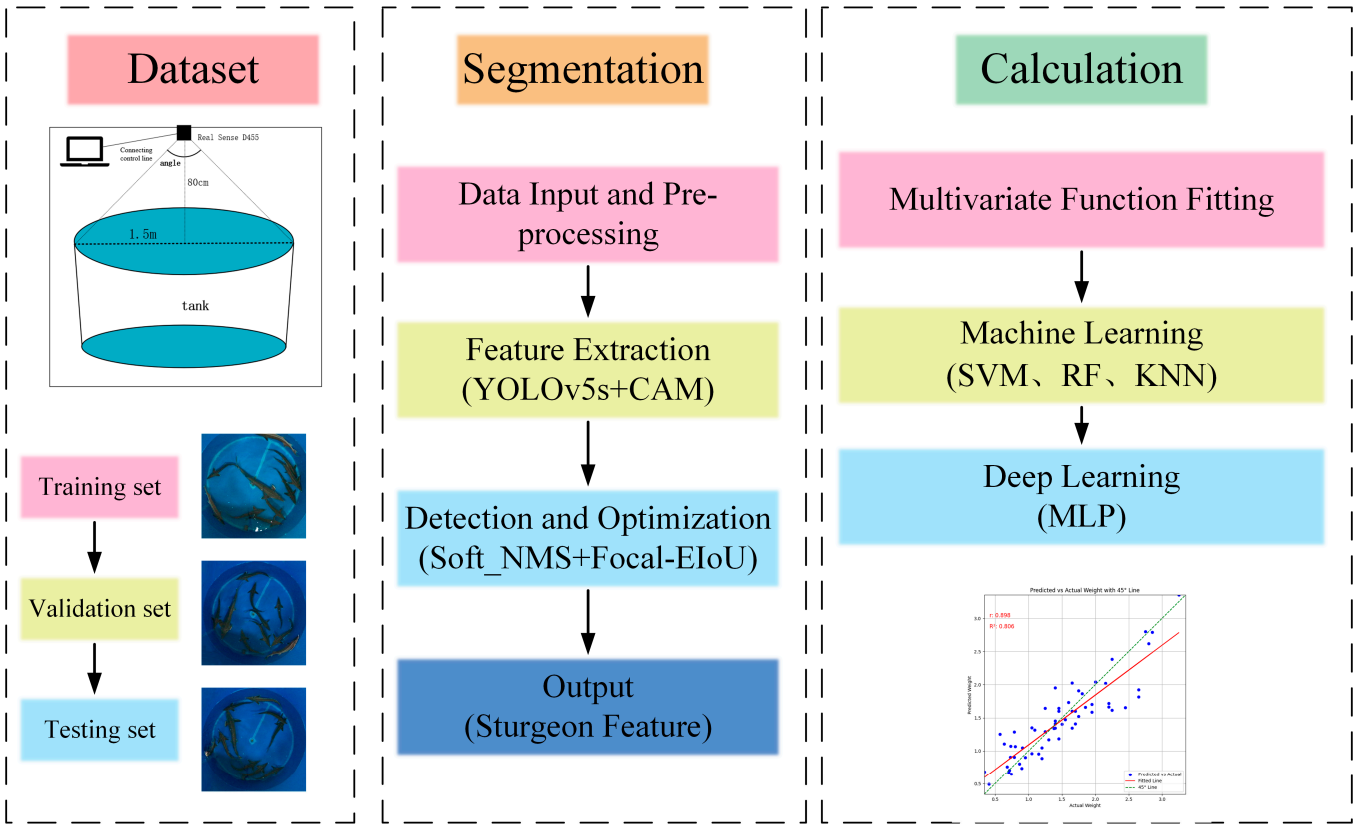

Figure 1. The overall architecture of non-contact sturgeon weight measurement system.

2.2. Data Collection and Dataset Creation

The collection of pictures and weighing of sturgeons were conducted at the Fisheries Science Institute, Beijing Academy of Agriculture and Forestry Science between April 2022 and October 2023 [15,16]. The picture data acquisition was completed using an Intel Real Sense D455 Stereo Depth Camera. To minimize the effects of glare and reflections caused by the water surface, a 30mm MCCPL polarizing filter (NiSi, Zhuhai, China) was affixed in front of the RealSense D455 camera. Pictures of each specimen were taken at 90 frames per second, and the auto exposure mode employed the color image sensor and the infrared sensor. The field of view for the color image was 86° × 57° (H × V) with a resolution of 640 × 480, whereas the field of view for the depth image was 87° × 58° with a resolution of 640 × 480 [17]. Figure 2 shows that the camera was mounted 0.8–1.2 m above the tank, vertically aligned with the water surface to capture images at varying distances [18]. Camera parameter calibration was autonomously performed on site using Intel®Depth Quality Tool software v2.50.0 [17]. Images of sturgeons in various swimming postures, group sizes, body sizes, lighting conditions, and distances were collected to improve the model's generalization.

Figure 2. Sturgeon image data acquisition device.

Two datasets were established: the sturgeon dataset and the occlusion dataset. The sturgeon dataset was used for model pre-training to obtain the pre-trained weight file, whereas the occlusion dataset was used for model training. The number of images in the dataset is shown in Table 1. The sturgeon dataset included 100 sturgeons, representing five species: *Acipenser baerii*, *Acipenser schrenckii*, *Acipenser ruthenus*, *Acipenser gueldenstaedtii*, and hybrid sturgeons. A total of 4000 images were captured, and 40 images were taken per sturgeon [17]. To minimize redundancy and avoid overfitting due to high similarity between frames, the structural similarity index (SSIM) algorithm was applied, and the sturgeon dataset was reduced to 1974 images. Data labeling was conducted on the obtained images using LabelImg software 1.8.6, and the corresponding TXT files were generated [18]. Training deep convolutional neural networks requires a large number of data. Extremely limited numbers of data can result in underfitting or overfitting of deep convolutional neural networks. To enhance the diversity of the sturgeon images, various transformations were applied using Python scripts with the imgaug library for data augmentation. This approach augments dataset variability by simulating diverse conditions and mitigates overfitting risks, ultimately enhancing the model's generalization ability and accurate recognition of sturgeons in different aquaculture environments. The methods used for image enhancement are shown in Figure 3 [18]. The sturgeon dataset contained 3948 sturgeon images after enhancement. For the occlusion dataset, 40 sturgeons, including *Acipenser gueldenstaedtii* and *Acipenser ruthenus*, were grouped and placed in a round fish tank. Twenty images were captured per group, resulting in 800 images. The SSIM algorithm was then used to select 420 occlusion images, which were further augmented using the imgaug library. Finally, the occlusion dataset contained 1020 images for segmentation experiments. The sturgeon dataset and the occlusion dataset were divided into training, validation, and test sets in an 8:1:1 ratio to ensure a balanced evaluation for training deep convolutional neural networks [15].

The real weight data of the sturgeons were measured using the weighing scale to provide support. Each sturgeon was weighed five times, and its average weight was taken as the real weight to reduce the random error.

Table 1. The number of dataset images.

Image Name	Image Dataset	Sturgeon Species	Number	Original Image	Augmentation	Total
Sturgeon Dataset	Training set	*Acipenser baerii* *Acipenser schrenckii* *Acipenser ruthenus* *Acipenser gueldenstaedti* *hybrid*	20 20 32 8 20	1580	1580	3160
	Validation set	*Acipenser baerii* *Acipenser schrenckii* *Acipenser ruthenus* *Acipenser gueldenstaedti* *hybrid*	20 20 32 8 20	197	197	394
	Testing set	*Acipenser baerii* *Acipenser schrenckii* *Acipenser ruthenus* *Acipenser gueldenstaedti* *hybrid*	20 20 32 8 20	197	197	394
Occlusion Dataset	Training set	*Acipenser gueldenstaedti* *Acipenser ruthenus*	8 32	336	480	816
	Validation set	*Acipenser gueldenstaedti* *Acipenser ruthenus*	8 32	42	60	102
	Testing set	*Acipenser gueldenstaedti* *Acipenser ruthenus*	8 32	42	60	102

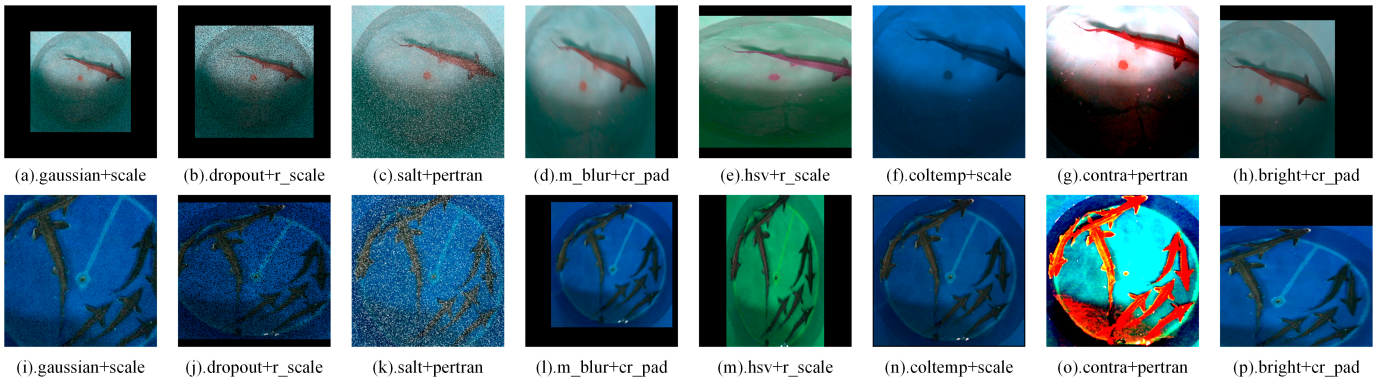

Figure 3. Image enhancement effect diagram. (**a**–**h**) schematic diagrams of single sturgeon amplification method; (**i**–**p**) schematic diagrams of occlusion sturgeon amplification method.

2.3. Experimental Platform and Parameters Setting

The target detection model was trained on the Windows 11 operating system, the CPU was anRyzen 7 6800H with Radeon Graphics 3.20 GHz (AMD, Santa Clara, CA, USA), and the GPU was an GeForce RTX3060 (NVIDIA, Santa Clara, CA, USA). The version of CUDA was 12.4, and the version of Cudnn was 8.9.7. Python 3.9 was used as the main programming language in the experiment [18].

In the model training, 480 × 640 of RGB images were input, the initialize learning rate of the network, momentum factor training, and weight attenuation were set at 0.01, 0.937, and 0.0005, respectively, SGD was chosen as the optimizer, and the hyperparameter was hyp. scratch-low [15].

2.4. Improved YOLOv5 Algorithm

2.4.1. YOLOv5 Algorithm

The YOLOv5 algorithm is an upgraded version of the YOLOv4 algorithm [18], and it offers 10 different versions with varying numbers of convolutional layers for optimal application scenarios [19]. The model size gradually increases as the number of convolutional layers increases, while detection performance improves and detection speed decreases. This study focuses on the YOLOv5s 6.0 model due to its balanced size and accuracy [20] and it consists of four components: input layers, Backbone network, Neck network, and Head network [21]. The YOLOv5 model's input layers are responsible for receiving and preparing the input image, which involves enhancing the data with the Mosaic technique, scaling the image adaptively, and calculating anchor boxes dynamically [19]. The Backbone component serves as a feature extractor in the network. It extracts features from the input image using convolutional structures, C3 modules, and spatial space pyramid pooling (SPPF) modules [20]. The convolutional module consists of convolution operations, batch normalization, and SiLU activation. The C3 module is based on the cross stage partial network concept and includes three standard convolutional layers along with multiple bottleneck modules [22]. To expand the receptive field's feature map, the SPPF module utilizes serial max-pooling for multiscale fusion [23]. The Neck network in YOLOv5 combines backbone features and enhances the model's capability to express features effectively [19]. It is a combination of a feature pyramid network (FPN) and path aggregation network (PAN), merging shallow graphical features with deep semantic features [21]. The responsibility for predicting bounding boxes and class probabilities for objects in an input image lies within the domain of the Head component [19]. For loss calculation at the output layers of the YOLOv5 model, three components contribute to it [21]. The complete intersection over union (CIoU) loss function is employed to calculate boundary regression loss, weighted non-maximum suppression (NMS) is performed on CIoU to efficiently select the optimal bounding box, and confidence prediction loss is calculated using Binary Cross Entropy with the Logits loss function while class prediction loss uses the Binary Cross Entropy loss function [18].

2.4.2. Context Augmentation Module

Small object information is gradually lost during forward propagation, which leads to the low detection accuracy and high miss rate of the network for sturgeon targets, so the CAM is incorporated into the PAN structure of the Neck network in this paper [24].

The contextual information from various receptive fields is obtained by CAM through dilated convolution with different convolution rates, and then injected into the FPN in a top-down manner. The input [bs, C, H, W] undergoes convolution operations with dilation rates of 1, 3, and 5. Here, bs represents the batch size, C represents the number of channels, H represents the height of the feature map, and W represents its width. To ensure that detailed features are captured without introducing excessive parameters due to small input size in this module, a kernel size of 3×3 is used instead of larger convolutions. Additionally, to maintain parameter efficiency while preserving information richness during compression and expansion stages within this process flow, the number of convolution kernels is chosen as C/4. Initially compressing the number of channels to one-fourth of the input's value allows for subsequent expansion back to C using a 1×1 convolution operation. This results in three outputs with identical sizes but distinct receptive fields [25]. Finally, these obtained features are fused together using adaptive fusion technology suitable for detecting medium and large targets. Figure 4 illustrates both the CAM structure schematic and fusion method employed in this study.

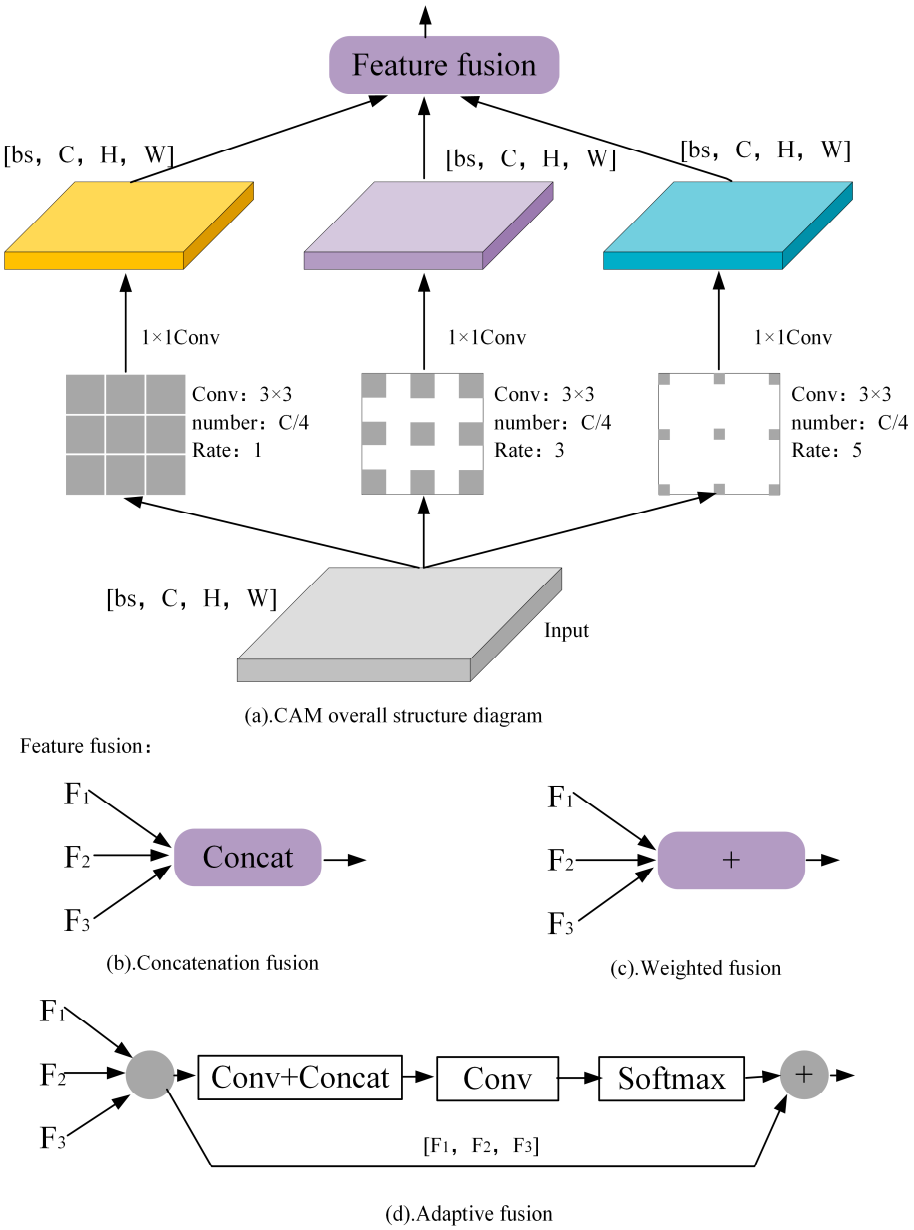

Figure 4. The structure of CAM. (**a**) CAM overall structure diagram; (**b**–**d**) three different feature fusion methods. Adaptive fusion is what we use.

2.4.3. Focal-Efficient Intersection over Union Loss

The conventional YOLOv5 employs the CIoU loss function for bounding box regression, which exhibits superior performance compared with IoU, GIoU, and DIoU [26]. However, the aspect ratio representation in the CIoU loss function is relative and does not address the issue of balancing between high-quality and low-quality samples [27]. To tackle this challenge, this study adopts the efficient intersection over union (EIoU) loss function instead of CIoU to calculate width and height differences. Additionally, we introduce Focal loss to prioritize regression on high-quality anchor boxes [28,29]. The formula for Focal-EIoU is as shown in Equations (1) and (2):

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(\omega, \omega^{gt})}{C_\omega^2} + \frac{\rho^2(h, h^{gt})}{C_h^2} \quad (1)$$

$$L_{Focal-EIoU} = IoU^\gamma L_{EIoU} \quad (2)$$

In Equation (1), L_{EIoU} comprises three components: the IoU loss, L_{IoU}, the distance loss, L_{dis}, and the aspect ratio loss, L_{asp}, and IoU is a metric that quanlifies the overlap between the predicted and ground truth bounding boxes. L_{dis} penalizes the difference in the center points (b and b^{gt}) of the predicted and ground truth boxes, normalized by the diagonal, c, of the smallest enclosing box. L_{asp} addresses differences in the width, ω, and height, h, of the predicted and ground truth boxes, using constants C_w and C_h for normalization. In Equation (2), γ is a parameter controlling the degree of outlier suppression.

2.4.4. Soft Non-Maximum Suppression

In the prediction stage, the conventional NMS algorithm is widely used to address the issue of multiple repeated prediction boxes around the object [30]. By employing a manually set threshold, only candidate boxes with high confidence are retained while those with low confidence are discarded. Due to the close proximity of objects in the sturgeon dataset, applying the NMS algorithm leads to forced elimination of overlapping feasible detection boxes, resulting in missed detections during detection.

In this paper, the Soft-NMS algorithm was employed as a replacement for the original NMS algorithm [31,32]. The Soft-NMS algorithm can effectively retain the object detection boxes with higher value by the Gaussian penalty function to reduce the t_i score gradually instead of directly suppressing the adjacent object boxes with lower scores [30]. The Soft-NMS algorithm is computed according to Equation (3):

$$s_i = \begin{cases} s_i(1 - IoU(M, t_i)), IoU(M, t_i) \geq N_t \\ s_i, IoU(M, t_i) < N_t \end{cases} \quad (3)$$

Equation (3) redefines the scores of the boxes. Thus, the detection boxes with a large overlap with the maximum detection boxes are assigned a very low score [25]. However, the original detection score of detection boxes with only a small overlap with the maximum detection boxes is not considerably changed. In Equation (3), M is the bounding box with the highest score, t_i is the i-th detection's bounding box and s_i is the original score of the i-th detection, IoU (M, t_i) is a metric that quantifies M and t_i, and N_t is the threshold for adjustment, used to determine whether the score needs to be modified.

The overall network structure of the improved YOLOv5s algorithm is shown in Figure 5.

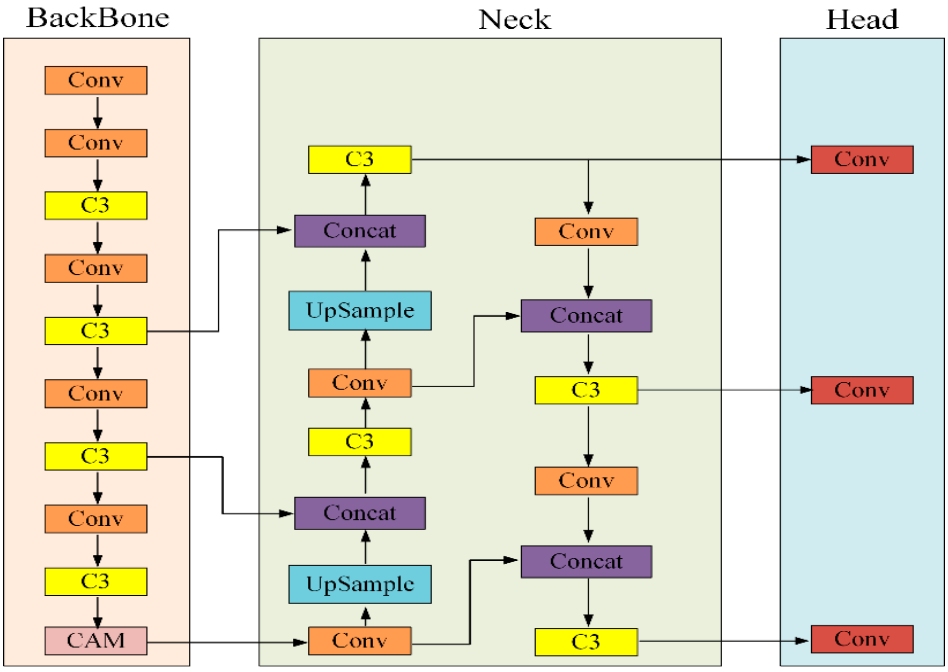

Figure 5. The structure diagram of improved YOLOv5s algorithm.

3. Results and Discussion

3.1. Evaluation Metrics

To evaluate the accuracy of the model for target detection, we assessed various metrics: precision (P), recall (R), F1-score, and mean average precision (mAP), along with model speed measured by number of layers (layers), number of parameters (Parameters), and giga floating-point operations per second (GFLOPs) [25]. P verifies the accuracy of the prediction results by calculating the ratio of correctly identified positive samples (True Positives) to the total predicted positive samples, which include both correctly identified positives (True Positives) and incorrectly identified negatives (False Positives) [26,29]. Meanwhile, R assesses the comprehensiveness of predictions by measuring the proportion of correctly identified positive samples (True Positives) out of all actual positive samples, including those incorrectly classified as negative (False Negatives) [27]. The F1-score serves as the reconciled average of P and R [16,29]. The average precision (AP) is defined as the area enclosed by the precision–recall curve for a specific category in the training results, and the mAP is the average of all APs across different categories [18,27]. Specifically, $mAP_{0.5}$ represents the average detection precision when the intersection over union (IoU) threshold is 0.5, and $mAP_{0.5-0.95}$ represents the average detection precision calculated across IoU thresholds from 0.50 to 0.95 in increments of 0.05.

3.2. Ablation Experiment

The Ablation Study is a widely adopted experimental methodology, particularly in the fields of machine learning, deep learning, and system design. It involves systematically eliminating or modifying specific components of a model or system and observing the resulting performance changes to determine the contributions of these components towards the overall effectiveness of the system. Specifically, it assists researchers in understanding which modules, features, or design decisions are essential for achieving desired outcomes while identifying those with relatively lesser impact [33]. To demonstrate the effect of various enhancement methods on model performance visually, ablation experiments were conducted using YOLOv5s as the baseline model. The focus was on three specific factors: Soft-NMS, CAM, and Focal-EIoU loss. To evaluate the effectiveness of these proposed methods thoroughly, quantitative ablation experiments were performed on the sturgeon dataset. The model underwent 100 epochs of training on our experimental platform to ensure robust, reliable results [30]. The experimental results in Table 2 comprehensively compare the model's performance with and without the inclusion of each improvement to pinpoint the contribution of each enhancement to the overall model performance.

Table 2. Improved mechanism ablation test results.

No.	Network Model	Layers	Parameters/$\times 10^6 M$	Computation/GFLOPs	$mAP_{0.5-0.95}$/%
1	YOLOv5s	157	7.02	15.8	94.4
2	YOLOv5s+Soft-NMS	157	7.02	15.8	95.2
3	YOLOv5s+Soft-NMS+CAM	164	14.23	21.6	95.5
4	YOLOv5s+Soft-NMS+CAM+Focal-EIoU	164	14.23	21.6	95.6

Comparing with experiments 1 and 2, a remarkable improvement was noted after using Soft-NMS: $mAP_{0.5-0.95}$ was improved by 0.8% and the model size remained the same [26]. The sturgeon dataset had a large amount of occlusion, and traditional NMS may miss detections after reaching the set threshold. Soft-NMS replaced the original high score with a low score and then recalculated the score of the current detection box, which maximized the retention of heavily occluded targets. The improved method had remarkable improvements for data with severe occlusion. The comparison between experiments 2 and 3 shows that CAM greatly improved the accuracy of the network [27], and the accuracy increased by 0.3%, although the parameter quantity increased. As the CAM module reduced the information loss of small objects during forward propagation, the model's

detection accuracy of the sturgeon improved. Compared with experiments 3 and 4, using Focal-EIoU instead of the CIoU loss function, $mAP_{0.5-0.95}$ was improved by 0.1% compared with CIoU, and no additional parameters were introduced. Focal-EIoU reduced the overall loss, sped up model convergence, and focused more on high-quality anchor boxes.

Overall, after adding the CAM module to the neck of the model and introducing the Focal-EIoU loss and Soft-NMS algorithm, the final improved YOLOv5s model was obtained, with an $mAP_{0.5-0.95}$ value of 95.6%, which is 1.2% higher than the baseline model [34]. However, this enhancement increased GFLOPs and indicated a higher computational cost. The overall increase in GFLOPs is mainly due to the need to add more convolution layers by using CAM. These additional convolution layers increase the calculation load and inference time of the model, which leads to a decrease in the running speed [27]. These algorithms were indispensable for the improvement of overall accuracy, and increased the detection accuracy of the entire model. Therefore, the ablation experiment shows that the model's overall performance improved [30].

3.3. Detection Performance of Different Algorithms for Different Occlusion Conditions

In intensive aquaculture, large quantities of sturgeons are often in a culture tank, which leads to numerous inevitable occlusion problems [35]. These occlusions increase the difficulty of sturgeon recognition and reduce the accuracy of identification [34]. Through long-term observation of sturgeon occlusion, occlusion was categorized in this paper into three types, namely, I-shaped occlusion, II-shaped occlusion, and III-shaped occlusion, as illustrated in Figure 6. The snout and centroid of the sturgeons were selected, a vector was drawn from the snout to the centroid, and the occlusion was classified based on the angle between two vectors from different sturgeons. To characterize the occlusion types intuitively, a factor called the "occlusion angle" was proposed [36]. The occlusion angle could be obtained by the definition of scalar product as follows:

$$occlusion\ angle = \cos^{-1} \frac{\boldsymbol{v_1} \cdot \boldsymbol{v_2}}{\| \boldsymbol{v_1} \| \cdot \| \boldsymbol{v_2} \|} \tag{4}$$

where the occlusion angle is the angle between two vectors, $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$; here $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ represent the instantaneous vectors of fish 1 and fish 2, respectively. For example, the instantaneous vector of fish 1 is given as

$$\boldsymbol{v_1} = (x_{i+1} - x_i) \cdot \boldsymbol{i} + (y_{i+1} - y_i) \cdot \boldsymbol{j} \tag{5}$$

where $\boldsymbol{i}$, $\boldsymbol{j}$ are the unit vectors along the x- and y-axes. x_{i+1} and y_{i+1} are the x, y coordinates of the centroids, and x_i and y_i is the x, y coordinates of the snout.

When the occlusion angle is equal to 0, this situation is classified as a I-shaped occlusion. When the occlusion angle is between 0 and 90°, it is classified as a II-shaped occlusion. When the occlusion angle is greater than 90 degrees, this condition is classified as a III-shaped occlusion.

Occlusion can remarkably influence the sturgeon image segmentation task. First, occlusion hides parts of the sturgeon's body and prevents the model from accurately capturing certain targets or features, which negatively affects the segmentation performance, especially when the occluded region is the primary area to be segmented. Second, when the sturgeon is partially obscured, the target's boundary becomes blurred, and the difficulty of the segmentation task is increased. This blurring can lead to inaccuracies in determining the boundary position and imprecise, fuzzy segmentation outcomes. Third, if the occluded part overlaps with targets from other categories, the model might mistakenly assign the occluded region to these other categories, leading to false segmentation. Fourth, occlusion causes a loss of contextual information, which can impair the model's overall understanding of the image. This lack of context hinders the model from inferring the category or attribute of the occluded part accurately. Finally, occlusion increases the complexity of the image segmentation task and causes a decline in model performance. The model must be

adequately robust to handle occlusions and segment the entire target accurately, not just the visible parts. In summary, occlusion poses a substantial challenge to image segmentation, complicates the task, and demands greater robustness and precision from the model.

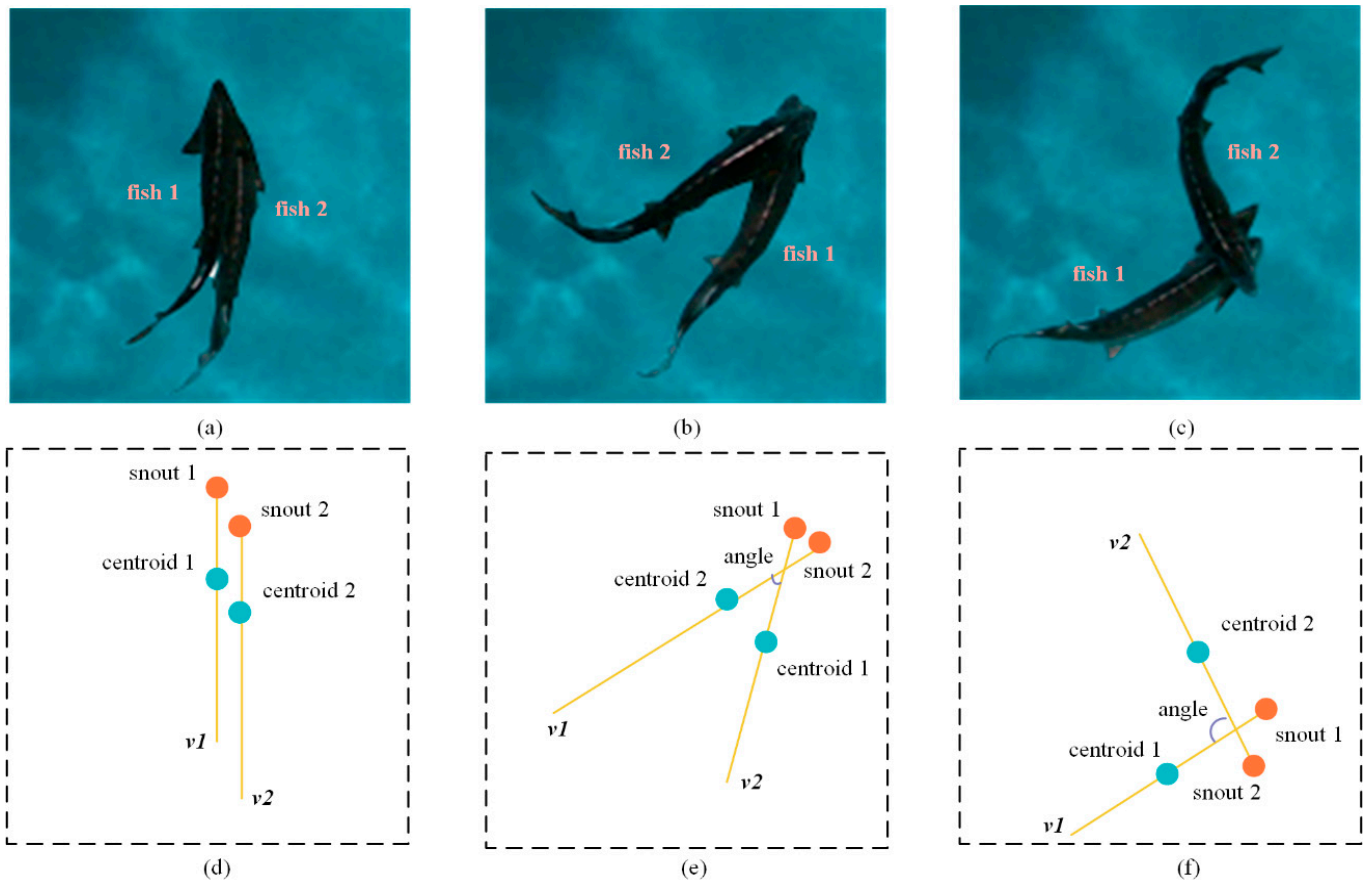

Figure 6. The schematic diagram of occlusion sturgeon classification. The orange dots indicate the sturgeon's snout, the blue dots indicate the sturgeon's centroid, and the yellow line represents the vector corresponding to the sturgeon. The angle between two vectors is the occlusion angle that we define. (**a**,**d**) the I-shaped occlusion; (**b**,**e**) the II-shaped occlusion; (**c**,**f**) the III-shaped occlusion.

To evaluate the segmentation accuracy of the model across the different types of occlusion, sturgeon images corresponding to each occlusion type were selected to construct various test sets [33]. Additionally, a comprehensive test set containing various sturgeon occlusion images was compiled. Tests were conducted using all test sets to determine the model's segmentation accuracy. The specific results were evaluated using $mAP_{0.5-0.95}$, as shown in Table 3 [37].

According to $mAP_{0.5-0.95}$ of the segmentation results in Table 2, the improved YOLOv5s model achieved the top two results across four occlusion datasets and attained the highest segmentation accuracies of 93.0% and 89.8% in the II-shaped occlusion test dataset and the comprehensive test dataset, respectively. Additionally, it secured second place in the I-shaped occlusion test dataset and III-shaped occlusion test dataset and reached segmentation accuracies of 93.0% and 95.5%, respectively. These conclusions strongly indicate that our improved model is the most effective segmentation model for occlusion sturgeon classification [34].

Table 3. Different occlusion conditions of test results.

No.	Network Model	I-Type Block	II-Type Block	III-Type Block	Comprehensive Test
		$mAP_{0.5-0.95}$/%	$mAP_{0.5-0.95}$/%	$mAP_{0.5-0.95}$/%	$mAP_{0.5-0.95}$/%
1	YOLOv5s	91.3	93.7	94.0	88.0
2	YOLOv5s+CAM	91.9	94.2	94.7	87.9
3	YOLOv5s+Soft_NMS	93.3	94.2	95.0	89.5
4	YOLOv5s+Focal-EIoU	91.7	93.5	94.6	88.3
5	YOLOv5s+CAM+Soft_NMS	92.7	94.8	96.1	89.5
6	YOLOv5s+CAM+Focal-EIoU	91.8	94.6	95.2	88.3
7	YOLOv5s+Focal-EIoU+Soft_NMS	92.8	93.8	95.1	89.7
8	YOLOv5s+CAM+Focal-EIoU+Soft_NMS	93.0	94.9	95.5	89.8

3.4. Fish Mass Estimation

In weight prediction, three approaches are commonly used for weight estimation: fitting methods, machine learning, and deep learning. Fitting methods are typically based on statistical regression analysis, which describes the relationship between input variables and output variables through mathematical functions. They assume a certain relationship between weight and specific input variables (such as body length and area) and use the least square method to fit the relationship between body length, area, and weight. By contrast, machine learning methods automatically learn the complex relationships between inputs and outputs from the data by constructing models. Common machine learning algorithms include SVM, RF, and KNN. These methods do not require prior assumptions about the relationships between variables, so they are more suitable for handling complex pattern recognition tasks. Deep learning is a subfield of machine learning that employs multilayer neural networks to process large-scale, high-dimensional data. Deep learning can automatically extract high-level representations of input features and demonstrates exceptional performance in tackling complex nonlinear problems. Each of these approaches has its strengths and weaknesses. Therefore, the three types of weight prediction models were validated and compared in this paper to determine the optimal weight prediction model for sturgeons.

Multivariate fitting is one of the most commonly used fitting methods and is well suited for modeling the relationship between multiple independent variables and a single dependent variable. This approach enables a more accurate description and prediction of the interactions between variables in complex systems. Therefore, multivariate fitting was utilized to develop a sturgeon weight prediction model. From the sturgeon dataset, as described in Section 2.2, 61 sturgeons of various species were selected, and their body length and area data were extracted. This information was then combined with the actual measured weight data to create a new dataset, referred to as the Weight Prediction Dataset. The least square method for regression analysis was applied, the prediction model by this dataset was developed, and the equation is as follows:

$$W = 1.710L + 11.354A^{1.059} - 0.392 \quad (6)$$

where W represents the weight of the sturgeon, L denotes the body length, and A is the 2D area derived from the model segmentation. The results of this model indicate a correlation coefficient of 0.888, an R^2 of 0.789, and a mean squared error (MSE) of 0.118. In a related study, Naruephorn et al. collected fish images in turbid water and conducted a multivariate fitting analysis to predict fish weight, resulting in a weight prediction model with an R^2 of 0.7 [38].

The SVM is a widely used supervised learning algorithm commonly applied to regression tasks. The core idea of SVM regression is to find the optimal hyperplane that best fits the data, with the key advantage of not requiring any assumptions about the data distribution. The SVM offers high flexibility and strong computational capabilities.

Mohammadmehdi et al. used an SVM algorithm to develop a fish weight prediction model, which achieved an R^2 of 0.872 and an RMSE of 0.1304 [3]. Based on these advantages and because researchers have used the SVM for fish weight prediction, SVM regression was employed in this paper to build a sturgeon weight prediction model. To ensure comparability, the SVM model was evaluated on the Weight Prediction Dataset, with a 4:1 split between the training and test sets. The SVM model used a radial basis function kernel with a regularization parameter set to 1 and an epsilon of 0.1 to allow errors within this range without penalty. The final model achieved a correlation coefficient of 0.822, an R^2 of 0.676, and an MSE of 0.168.

RF is an ensemble learning method that constructs multiple decision trees and aggregates their predictions to enhance accuracy and stability. Its key advantage lies in effectively handling large, high-dimensional datasets while mitigating the risk of overfitting. In this study, RF algorithms were employed to develop a sturgeon weight prediction model, ensuring comparability by evaluating it on the Weight Prediction Dataset with a 4:1 training–test split ratio. The RF model was configured with two decision trees, yielding a final model that achieved a correlation coefficient of 0.723, R^2 of 0.523, and MSE of 0.235.

The KNN algorithm is a straightforward and non-parametric method that classifies data points based on the majority class of their nearest neighbors. This approach proves to be particularly effective for smaller datasets, where pattern recognition heavily relies on local information. To ensure comparability, we evaluated the performance of the KNN model on the Weight Prediction Dataset by dividing it into training and testing sets at a ratio of 4:1. In our experiments, we set the 'n_neighbors' parameter to 10. The final model achieved notable results with a correlation coefficient of 0.822, an R^2 value of 0.676, and an MSE value of 0.190.

The MLP is a common deep learning model that uses multiple layers of neural networks to map input features such as length and area to predicted values of weight. The number of layers and neurons in an MLP can be flexibly adjusted to accommodate the complexity of different datasets. In this paper, an MLP-based model was developed to predict sturgeon weight by exploring the relationship between body length, area, and weight. The model was trained using the Weight Prediction Dataset, with a 4:1 split between the training and test sets. The MLP architecture consisted of four layers, with ReLU activation functions, and a single neuron in the output layer for the weight prediction task. The optimizer used was Adam, and the loss function was MSE. The model was trained for 100 epochs. With the strong nonlinear modeling capabilities of the deep learning framework, the complex effects of the input variables on the weight were successfully captured. The correlation coefficient of 0.898 and an R^2 of 0.806 indicate the substantial explanatory power of the model, while an MSE of 0.148 reflects a low prediction error, highlighting the model's accuracy.

In this paper, five different approaches for predicting sturgeon weight were implemented: multivariate fitting, SVM, KNN, RF, and MLP. All models were trained and tested on the same dataset. The performance of the model was evaluated using the correlation coefficient, determination coefficient, and MSE. The experimental results are shown in Table 4.

Table 4. Performance of sturgeon weight estimation with fitting, SVM, KNN, RF, and MLP methods.

No.	Model	r	R^2	MSE
1	Fitting	0.888	0.789	0.118
2	SVM	0.822	0.676	0.168
3	KNN	0.822	0.676	0.190
4	RF	0.723	0.523	0.235
5	MLP	0.898	0.806	0.148

The MLP model achieved the highest correlation coefficient of 0.898 and demonstrated superior predictive ability with an MSE of 0.148, indicating a low prediction error. Therefore,

the MLP algorithm is the most effective approach for sturgeon weight inversion. The results of the weight prediction model based on the MLP are shown in Figure 7.

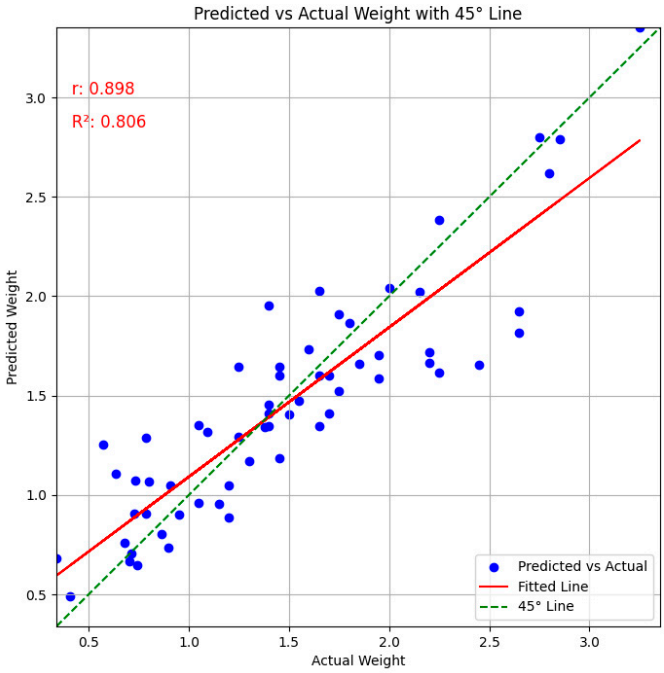

Figure 7. Scatterplots of measured weight versus predicted obtained by MLP.

4. Conclusions

This paper focused on sturgeons in an intensive culture environment to meet the technical requirements for in situ weight measurement. Based on the YOLOv5s, the model was enhanced by replacing the SPPF module in the Backbone network with the CAM module, adding the Soft-NMS algorithm to enhance the original NMS algorithm, and displacing the CIoU loss function to the Focal-EIoU loss function to improve feature extraction [26]. Ablation tests on the improved YOLOv5s model demonstrated a 1.2% increase in average precision, reaching 95.6%, thereby enhancing robustness in detecting individual sturgeons. In occluded conditions, the experimental results demonstrated that the proposed method can achieve an estimation accuracy of more than 93% [39]. Additionally, the MLP model of sturgeon weight estimation displayed a correlation coefficient of 89.8% and remarkably outperformed the multivariate fitting model and the SVM model. These experiments confirm that the improved YOLOv5s and MLP method substantially improve the accuracy of sturgeon weight measurement.

However, the existing system has two limitations: (i) The dataset used consisted solely of manually captured images, which resulted in considerable variations in angles, scenes, and conditions, and (ii) the camera frame rate was not high enough to capture fast-swimming fish, and the lighting conditions affected the actual measurement [40]. Consequently, this dataset and camera may not be ideal. Despite these issues and current technological and algorithmic limitations, fully automated systems for species identification of sturgeons and phenotypic measurement of fish in aquaculture remain desirable and achievable [32].

Author Contributions: Conceptualization, K.L., S.Z., H.L. and R.X.; methodology, K.L., S.Z. and R.X.; software, K.L. and J.H.; validation, J.H. and K.L.; formal analysis, K.L.; investigation, K.L.; resources, K.L.; data curation, J.H. and K.L.; writing—original draft preparation, J.H.; writing—review and editing, K.L., S.Z., R.Z. and J.H.; visualization, J.H. and K.L.; funding acquisition, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing Natural Science Foundation, grant number 6232013, National Natural Science Foundation of China, grant number 32202996, Beijing Academy of Agriculture and Forestry Science Youth Science Foundation, grant number QNJJ202401, and Key Laboratory of Equipment and Informatization in Environment Controlled Agriculture, Ministry of Agriculture and Rural Affairs, China, grant number 2011NYZD2302.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to this study exclusively employing non-invasive monitoring techniques on the experimental animals without any involvement of animal ethical procedures. All animal handling and manipulation procedures were based on the standards of the Chinese Council on Animal Care.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

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

References

1. Wang, W.; Gao, L.; Liu, W.; Tian, Z.; Wang, X.; Hu, H. Regulation of antioxidant defense in response to heat stress in Siberian sturgeon (*Acipenser baerii*). *Aquaculture* **2023**, *572*, 739551. [CrossRef]
2. Zhang, L.; Wang, J.; Duan, Q. Estimation for fish mass using image analysis and neural network. *Comput. Electron. Agric.* **2020**, *173*, 105439. [CrossRef]
3. Saberioon, M.; Císař, P. Automated within tank fish mass estimation using infrared reflection system. *Comput. Electron. Agric.* **2018**, *150*, 484–492. [CrossRef]
4. Njane, S.N.; Shinohara, Y.; Kondo, N.; Ogawa, Y.; Suzuki, T.; Nishizu, T. Underwater fish volume estimation using closed and open cavity Helmholtz resonators. *Eng. Agric. Environ. Food* **2019**, *12*, 81–88. [CrossRef]
5. Li, D.; Hao, Y.; Duan, Y. Nonintrusive methods for biomass estimation in aquaculture with emphasis on fish: A review. *Rev. Aquac.* **2019**, *12*, 1390–1411. [CrossRef]
6. Arnal, B.J.G. A review on the use of computer vision and artificial intelligence for fish recognition, monitoring, and management. *Fishes* **2022**, *7*, 335. [CrossRef]
7. Shi, C.; Zhao, R.; Liu, C.; Li, D. Underwater fish mass estimation using pattern matching based on binocular system. *Aquac. Eng.* **2022**, *99*, 102285. [CrossRef]
8. Tonachella, N.; Martini, A.; Martinoli, M.; Pulcini, D.; Romano, A.; Capoccioni, F. An affordable and easy-to-use tool for automatic fish length and weight estimation in mariculture. *Sci. Rep.* **2022**, *12*, 15642. [CrossRef] [PubMed]
9. Mathiassen, J.R.; Misimi, E.; Toldnes, B.; Bondo, M.; Ostvik, S.O. High-speed weight estimation of whole herring (*Clupea harengus*) using 3D machine vision. *J. Food Sci.* **2011**, *76*, E458–E464. [CrossRef]
10. Hao, Y.; Yin, H.; Li, D. A novel method of fish tail fin removal for mass estimation using computer vision. *Comput. Electron. Agric.* **2022**, *193*, 106601. [CrossRef]
11. Yu, X.; Wang, Y.; Liu, J.; Wang, J.; An, D.; Wei, Y. Non-contact weight estimation system for fish based on instance segmentation. *Expert Syst. Appl.* **2022**, *210*, 118403. [CrossRef]
12. Zhang, T.Y.; Yang, Y.Q.; Liu, Y.Y.; Liu, C.L.; Zhao, R.; Li, D.L.; Shi, C. Fully automatic system for fish biomass estimation based on deep neural network. *Ecol. Inform.* **2024**, *79*, 102399. [CrossRef]
13. Yang, Y.; Xue, B.; Jesson, L.; Wylie, M.; Zhang, M.; Wellenreuther, M. Deep convolutional neural networks for fish weight prediction from images. In Proceedings of the 2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ), Tauranga, New Zealand, 9–10 December 2021; pp. 1–6. [CrossRef]
14. Billard, R.; Lecointre, G. Biology and conservation of sturgeon and paddlefish. *Rev. Fish Biol. Fish.* **2000**, *10*, 355–392. [CrossRef]
15. Zhu, A.; Zhang, R.; Zhang, L.; Yi, T.; Wang, L.; Zhang, D.; Chen, L. YOLOv5s-CEDB: A robust and efficiency Camellia oleifera fruit detection algorithm in complex natural scenes. *Comput. Electron. Agric.* **2024**, *221*, 108984. [CrossRef]
16. Wang, Z.; Jin, L.; Wang, S.; Xu, H. Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. *Postharvest Biol. Technol.* **2022**, *185*, 111808. [CrossRef]
17. Paudel, S.; de Sousa, R.V.; Sharma, S.R.; Brown-Brandl, T. Deep learning models to predict finishing pig weight using point clouds. *Animals* **2024**, *14*, 31. [CrossRef]
18. Xiao, F.; Wang, H.; Xu, Y.; Shi, Z. A lightweight detection method for blueberry fruit maturitybased on an improved YOLOv5 algorithm. *Agriculture* **2024**, *14*, 36. [CrossRef]
19. Arifando, R.; Eto, S.; Wada, C. Improved YOLOv5-based lightweight object detection algorithm for people with visual impairment to detect buses. *Appl. Sci.* **2023**, *13*, 5802. [CrossRef]
20. Zhao, Z.; Wang, J.; Zhao, H. Research on apple recognition algorithm in complex orchard environment based on deep learning. *Sensors* **2023**, *23*, 5425. [CrossRef]
21. Chen, X.M.; Chen, T.Z.; Meng, H.M.; Zhang, Z.Q.; Wang, D.H.; Sun, J.C.; Wang, J. An improved algorithm based on YOLOv5 for detecting Ambrosia trifida in UAV images. *Front. Plant Sci.* **2024**, *15*, 14. [CrossRef]
22. Zhang, L.; Li, J.; Zhang, F. An efficient forest fire target detection model based on improved YOLOv5. *Fire* **2023**, *6*, 291. [CrossRef]

23. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
24. Jinsheng, X.; Tao, Z.; Jian, Z.; Qiuping, L.; Liheng, Y. Small target detection network based on context augmentation and feature refinement. *Jisuanji Yanjiu Yu Fazhan/Comput. Res. Dev.* **2023**, *60*, 465–474. [CrossRef]
25. Gao, W.Q.; Gu, W.J.; Yin, Y.C.; Li, T.G.; Dong, P.L. ODCS-YOLO detection algorithm for rail surface defects based on omni-dimensional dynamic convolution and context augmentation module. *Meas. Sci. Technol.* **2024**, *35*, 13. [CrossRef]
26. Huang, B.; Liu, J.; Liu, X.; Liu, K.; Liao, X.; Li, K.; Wang, J. Improved YOLOv5 network for steel surface defect detection. *Metals* **2023**, *13*, 1439. [CrossRef]
27. Shi, T.; Ding, Y.; Zhu, W. YOLOv5s_2E: Improved YOLOv5s for aerial small target detection. *IEEE Access* **2023**, *11*, 80479–80490. [CrossRef]
28. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
29. Li, X.; Li, X.; Shen, Z.; Qian, G. Driver fatigue detection based on improved YOLOv7. *J. Real-Time Image Process.* **2024**, *21*, 75. [CrossRef]
30. Li, Y.; Li, S.; Du, H.; Chen, L.; Zhang, D.; Li, Y. YOLO-ACN: Focusing on small target and occluded object detection. *IEEE Access* **2020**, *8*, 227288–227303. [CrossRef]
31. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS—improving object detection with one line of code. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5562–5570. [CrossRef]
32. Liu, G.; Cai, S.; Wei, H.; Guo, H.; Ni, C.; Ni, Z. Identification methods for structural problems of bridges based on deep convolutional neural network. *Sens. Mater.* **2024**, *36*, 2033. [CrossRef]
33. Chen, S.; Zou, X.; Zhou, X.; Xiang, Y.; Wu, M. Study on fusion clustering and improved YOLOv5 algorithm based on multiple occlusion of Camellia oleifera fruit. *Comput. Electron. Agric.* **2023**, *206*, 107706. [CrossRef]
34. Lai, J.; Liang, Y.; Kuang, Y.; Xie, Z.; He, H.; Zhuo, Y.; Huang, Z.; Zhu, S.; Huang, Z. IO-YOLOv5: Improved pig detection under various illuminations and heavy occlusion. *Agriculture* **2023**, *13*, 1349. [CrossRef]
35. Chen, J.; Wu, J.; Wang, Z.; Qiang, H.; Cai, G.; Tan, C.; Zhao, C. Detecting ripe fruits under natural occlusion and illumination conditions. *Comput. Electron. Agric.* **2021**, *190*, 106450. [CrossRef]
36. Lin, K.; Zhang, S.Y.; Hu, J.J.; Lv, X.D.; Li, H.S. Quantitative comparison of 2D and 3D monitoring dimensions in fish behavior analysis. *J. Fish Biol.* **2024**, *104*, 929–938. [CrossRef] [PubMed]
37. Kim, S.; Hong, S.-J.; Ryu, J.; Kim, E.; Lee, C.-H.; Kim, G. Application of amodal segmentation on cucumber segmentation and occlusion recovery. *Comput. Electron. Agric.* **2023**, *210*, 107847. [CrossRef]
38. Tengtrairat, N.; Woo, W.L.; Parathai, P.; Rinchumphu, D.; Chaichana, C. Non-intrusive fish weight estimation in turbid water using deep learning and regression models. *Sensors* **2022**, *22*, 5161. [CrossRef]
39. Sthapit, P.; Kim, M.; Kim, K. A method to accurately estimate fish abundance in offshore cages. *Appl. Sci.* **2020**, *10*, 3720. [CrossRef]
40. Shafait, F.; Harvey, E.S.; Shortis, M.R.; Mian, A.; Ravanbakhsh, M.; Seager, J.W.; Culverhouse, P.F.; Cline, D.E.; Edgington, D.R. Towards automating underwater measurement of fish length: A comparison of semi-automatic and manual stereo–video measurements. *ICES J. Mar. Sci.* **2017**, *74*, 1690–1701. [CrossRef]