Journal of *Marine Science and Engineering*

MDPI

Technical Note

A New Workflow for Instance Segmentation of Fish with YOLO

Jiushuang Zhang [1,2] and Yong Wang [1,2,*]

[1] Institute for Ocean Engineering, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; zjs22@mails.tsinghua.edu.cn
[2] Shenzhen Key Laboratory of Advanced Technology for Marine Ecology, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China
* Correspondence: wangyong@sz.tsinghua.edu.cn

Abstract: The application of deep-learning technology for marine fishery resource investigation is still in its infancy stage. In this study, we applied YOLOv5 and YOLOv8 methods to identify and segment fish in the seabed. Our results show that both methods could achieve superior performance in the segmentation task of the DeepFish dataset. We also expanded the labeling of specific fish species classification tags on the basis of the original semantic segmentation dataset of DeepFish and completed the multi-class instance segmentation task of fish based on the newly labeled tags. Based on the above two achievements, we propose a general and flexible self-iterative fish identification and segmentation standard workflow that can effectively improve the efficiency of fish surveys.

Keywords: computer vision; deep learning; YOLO; DeepFish; automation

1. Introduction

1.1. Deep Learning for Fish Counting and Segmentation

Fish are important biological resources in global waters, and therefore, instant recognition and statistics in wild research fields of fish have far-reaching implications for ecological balance and the fishery industry. The application of computer vision-related technologies provides a new perspective for marine ecological studies. With the rapid growth of monitoring data and computing power, deep-learning technology has once again entered the field of computer vision.

Applying deep learning to a certain field actually faces many challenges. Firstly, this process requires a significant amount of raw data in the data-preparation stage. Taking computer vision as an example, it usually requires accumulating a large number of videos and images to complete the next step of the work. Then, this process requires significant manpower and material resources to complete the corresponding annotation work on the original data based on specific tasks, and the quality of annotation will directly determine the recognition effect. This necessitates more precise annotation for data of the same scale, with higher accuracy requirements and time cost for the annotation. Moreover, a decrease in data quality due to annotation errors is unavoidable. Subsequently, this process will consume a significant amount of computing power and time to complete training based on the annotated data. Although the emergence of GPUs has effectively shortened the time required for this step, it usually still takes several hours or even days to complete, and the specific values vary depending on the difficulty of the training task and the size of the data.

With the continuous development of modern technology, underwater vehicles, such as remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs), can dive to any depth of the seabed, cruise within a certain range, and collect a large number of relatively clear underwater images. This also lays a solid foundation for completing computer vision-related tasks based on fish images.

There are many specific task categories in computer vision, including classification, detection, and segmentation. The main focus of this article is on segmentation tasks.

check for updates

Citation: Zhang, J.; Wang, Y. A New Workflow for Instance Segmentation of Fish with YOLO. *J. Mar. Sci. Eng.* **2024**, *12*, 1010. https://doi.org/10.3390/jmse12061010

Academic Editors: Niamat Ullah Ibne Hossain and Chiranjibi Shah

Received: 31 May 2024
Revised: 14 June 2024
Accepted: 15 June 2024
Published: 18 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Different image segmentation tasks have different objectives and methods. Here are some main segmentation tasks and their differences:

1. Semantic Segmentation:

Objective: To identify the pixels of interest in the image and assign them to a semantic category in a specific business scenario, such as "fish", or more refined species, such as "tilapia" and "deep-sea angler fish".

2. Instance Segmentation:

Objective: On the basis of semantic segmentation, it is not only necessary to segment different object categories but also to distinguish different object instances of the same category (i.e., to uniquely identify each object, such as labeling several different tilapia in the image). This means that instance segmentation can be used simultaneously to complete counting tasks for different fish species.

3. Panopic Segmentation:

Objective: On the basis of instance segmentation, corresponding labels and unique instance identifiers for each pixel in the image can be determined. If it cannot be determined, an "unlabeled" category is provided.

These different segmentation tasks have different application scenarios and complexities, which render an increment of the complexity of the above tasks in sequence.

At present, there are some research works on fish counting and segmentation based on deep learning. A combined loss function based on the active contour theory and level set methods was proposed to refine the spatial segmentation resolution and quality [1]. A transformer-based model that uses self-supervision for high-quality fish segmentation was trained on videos without any annotations to perform fish segmentation in underwater videos taken in situ in the wild [2]. A method was then invented based on the model. Two novel architectures for the automatic and high-performance segmentation of fish populations were developed, in which EFS-Net (efficient fish segmentation network) and MFAS-Net (multi-level feature accumulation-based segmentation network) are the base and final networks, respectively [3]. Recently, two novel networks, namely PFFS-Net (parallel feature fusion-based segmentation network) and PIFS-Net (progressive information fusion-based segmentation network) were developed for pixel-wise fish segmentation. PFFS-Net is a base network that relies on parallel feature fusion to achieve a better segmentation performance. PIFS-Net is the final model of this work and depends on a progressive spatial feature fusion (SFF) mechanism to enhance segmentation accuracy [4]. An efficient deep-learning template for crowd counting aquatic animals was proposed by providing an open-source framework along with training data, in which deep learning with a density-based regression approach was integrated [5]. An algorithm for the super-resolution fusion of enhanced extraction features, named RMP-Net, was developed to perform semantic segmentation of seabed scenes, which solved the problems in underwater exploration that include the lack of ambient light on the seabed, the complicated seabed terrain, and the variety of creatures with different shapes and colors [6]. To address the problem that wireless communication is not feasible for big marine data due to the narrow-frequency bandwidth of acoustic waves and the ambient noise, an optimized deep learning for low-energy and real-time image processing was designated at the underwater edge [7].

1.2. Application Scenarios of Segmentation Technology in Ecology

The segmentation technology in computer vision provides powerful tools for the utilization and research of fish resources, specifically manifested in the following aspects:

1. Ecological monitoring and protection: By segmenting fish images, we can better understand the spatiotemporal distribution, quantity, and behavior of various fish species, which helps to protect endangered fish and maintain ecological balance.

2. Fisheries management: Fisheries are an important component of the global economy, and the accurate identification and measurement of caught fish contribute to sustainable fisheries management to ensure the sustainable development of fisheries resources.
3. Aquatic biology research: Segmented fish images provide rich reference data for aquatic biology research, which can help scientists and researchers gain a deeper understanding of fish population distribution, behavioral habits, habitat selection, and interactions with other organisms.

This paper aims to explore specific methods and practical applications of fish image segmentation to meet the above requirements. We will introduce relevant computer-vision technologies and emphasize their potential application value in the field of fish research. This research, therefore, aims to provide useful tools and assistance for ecologists, fishermen, and aquatic biologists that strongly support the protection and sustainable management of fish.

1.3. Datasets

At present, there are various publicly available datasets of fish or marine organisms for underwater computer-vision research in the academic community. We can obtain performance indicators of the same type by running different algorithms on these publicly available datasets and compare the different algorithms by evaluating the performance indicators. Due to the significant differences in annotation difficulty, most datasets only have the simplest classification annotation information or counting, positioning, and detection annotation information. Only a small number of datasets have segmentation-related annotation information, and the overall number of annotations is relatively small compared to other tasks. There are several datasets that exist for underwater image classification [8–15]. These datasets only classify objects under limited conditions, and most of them cannot solve the challenges encountered in object-detection tasks. In other words, these datasets can only classify images on cropped images, lacking descriptions and annotations of object size, shape, and other information.

2. Materials and Methods

2.1. Comparison between YOLOv5 and YOLOv8

YOLO, mentioned in the title, which stands for "You Only Look Once", is a state-of-the-art, real-time, object-detection system in computer vision. It was designed to quickly and accurately identify objects within images and videos. YOLO divides the image into a grid and predicts bounding boxes and probabilities for each grid cell, allowing for it to simultaneously detect multiple objects in a single pass. This approach reduces the need for computationally expensive region proposal networks, making YOLO exceptionally fast and efficient. As of December 2023, YOLO has iterated to its 8th version, with the 3rd and 5th versions widely used in a large number of AI applications. In addition to the official YOLO models, there are also many conceptual models derived from YOLO, such as YOLO9000, which emphasizes real-time detection capabilities for multiple objects. YOLO9000 is called a conceptual model in that it was first proposed to demonstrate the YOLO algorithm's ability to extend to detecting and recognizing a large number of object categories. As such, it is only a theoretical model used to discuss and explore the limits of algorithms. Correspondingly, unlike the actual released versions of YOLO, YOLO9000 does not have a publicly available code library or pre-trained models for researchers and developers to use, which brings difficulties for further in-depth research based on this version.

We used YOLOv5 and YOLOv8 as our basic frameworks to finish the instance segmentation workflow in this study.

The common features between YOLOv5 and YOLOv8 are as follows:

(1) For the backbone, CSP thought is adopted by both of them, and an SPPF module has been integrated into them.
(2) PAN (path aggregation network) thought is included in both of them.
(3) For classification, both of them apply BCE loss in the loss function.

The differences between YOLOv5 and YOLOv8 are as follows:

(1) For the backbone, a C2f module is integrated in YOLOv8 in contrast to a C3 module in YOLOv5.
(2) For the detection head, a coupled head and anchor-base are used in YOLOv5 in comparison to a decoupled head and anchor-free in YOLOv8.
(3) For the positive and negative sample assignment strategy, static assignment and TAL (task alignment learning) dynamic assignment strategies are adopted in YOLOv5 and YOLOv8, respectively.
(4) The PAN-FPN up-sampling CBS module in YOLOv5 is removed from YOLOv8.
(5) Object loss is removed from YOLOv8, whereas CIOU loss and DFL (dual focal loss) are included in YOLOv8.

It is indicated that YOLOv8 has better performance than YOLOv5 under the same experimental conditions in many scenes according to ultralytics' official benchmark report.

2.2. Preparation for Transfer Learning and Retraining

From a usability perspective, we use Python 3.10.11 as the programming language required for implementing the experiments, and Pytorch 2.0.1 as the deep-learning framework for implementing YOLOv5 and YOLOv8. We trained our model and performed corresponding inference and prediction work based on the model on the CentOS 7 operating system. We used YOLOv5s and YOLOv8s as our pre-trained models, in which "s" refers to the size. YOLOv5 and YOLOv8 both have different models with different sizes, including "n" (nano), "s" (small), "m" (medium), "l" (large), and "x" (extra-large). These models have different depths, widths, and channels, leading to differences in their performance. From the nano specification to the extra-large specification, the inference speed of the model will gradually slow down, followed by mounting accuracy. For a real-time scene, we could not use models with a large size on devices due to a lack of computer power. For example, according to ultralytics' official benchmark report, YOLOv8n can handle one image in 80.4 ms, outcompeting YOLOv8s that takes 128.4 ms, with a visualization of no lag when watching videos.

YOLOv5s and YOLOv8s were trained on the COCO instance segmentation dataset with 640 × 640 resolution images. We conducted the training based on these pre-trained models on our own dataset. It is also known as "transfer learning". The deep neutral network with trained weight has a strong ability for feature extraction, which is less time-costly than training the model from scratch.

For binary classification cases, we simply followed the division in the DeepFish semantic segmentation dataset. The dataset consists of a training set, a validation set, and a test set. There were 149, 161, and 96 images in the training set, the validation set, and the testing set, respectively. These datasets also include 161 (training), 65 (validation), and 59 (testing) images without fish. We chose the hyperparameters (Table 1) to perform training for YOLOv5 and YOLOv8.

To construct a new workflow for fish recognition, we modified the original DeepFish semantic segmentation dataset to distinguish different fish species. The original labeling method only marked the fish as white and the background as black. We set boundaries that are usually composed of a series of point-level annotations for each fish. When we converted the dataset in mask style into COCO instance segmentation dataset style, we could only mark the pixel with labeling fish as a fixed-type 0 as an example. Now, in order to distinguish fish species from the labeling, we need to mark fish of different species with different class labels. We finally labeled 19 classes in the training set and validation set (18 in the training set; one in the validation set). It satisfied our criteria well because there were always fish species present in our training set that, in reality, had to be recognized at least as class fish. Then, we could hand over the potential fish to manually conduct taxonomic classification. The labeling distribution is shown in Figure 1, including class and bounding box position.

Table 1. Hyperparameters for YOLOv5 and YOLOv8 training.

Model	Name	Value
YOLOv5	pre-trained model	YOLOv5s
	epochs	600
	early stop epochs	200
	batch	64
	image size	640 × 640
	workers	32
	optimizer	SGD
YOLOv8	pre-trained model	YOLOv8s
	epochs	600
	early stop epochs	200
	batch	64
	image size	640 × 640
	workers	32
	optimizer	SGD
	lr_0	0.01
	lr_f	0.01

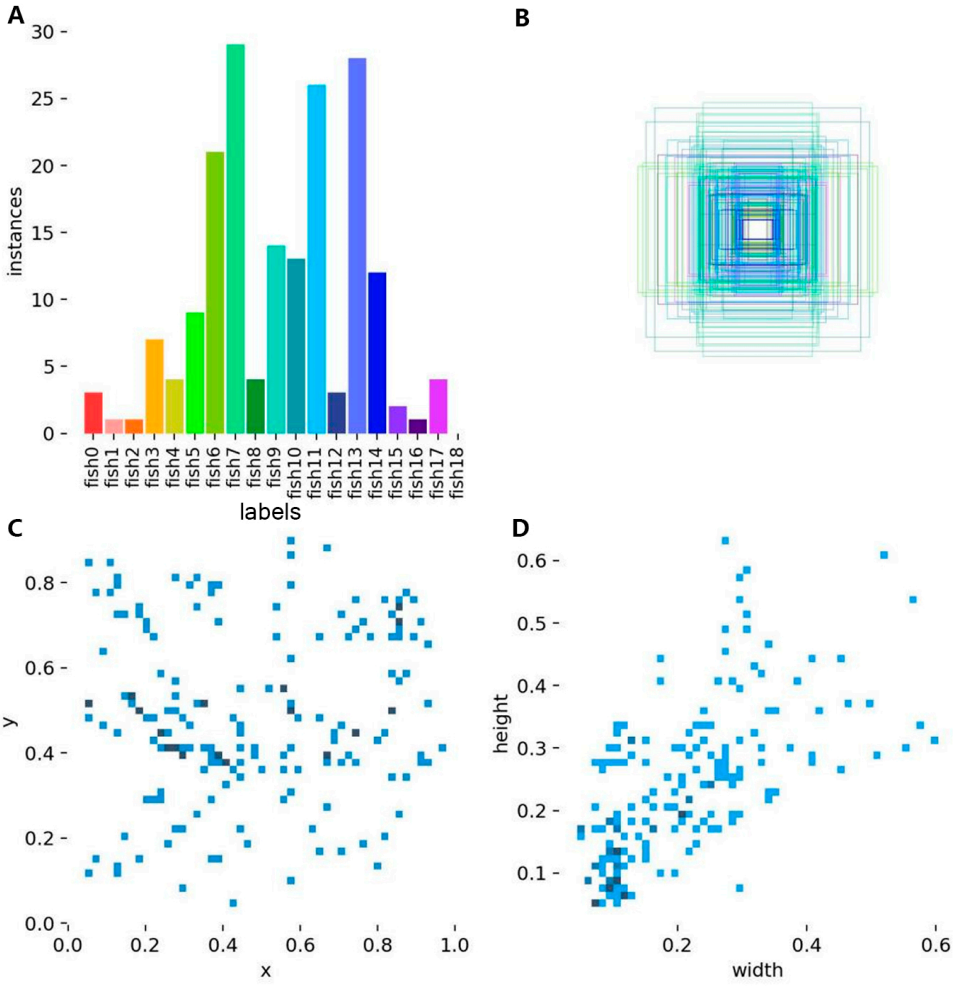


Figure 1. Distribution of DeepFish new multi-class segmentation labels. (**A**) The instance number for every fish class. (**B**) Labeling position drawn on one figure. (**C**,**D**) The relationship between the x-coordinate and y-coordinate of the upper-left corner and width and height of the annotated bounding box, respectively.

2.3. New Workflow Design

There are two specific approaches to transforming the DeepFish's binary segmentation problem into a more valuable multi-classification segmentation problem. The first approach is (1) the selection of as many potential fish candidates as possible based on the general features of the fish (such as body shape contours) through the previously trained network for fish recognition and (2) classification of these candidates into the subsequent classification network. The second approach is (1) the direct implementation of multi-classification and (2) determination of the high and low thresholds a and b. If it is higher than a, the classification result is considered completely reliable. If it is between a and b, it is considered as an unknown fish species and will be handed over to fish biologists for manual intervention and discrimination. Otherwise, it is not a fish. The second method was adopted in this study. The visualization of our new workflow is shown in Figure 2. The parts in the dashed box reflect the self-iterative nature of this workflow.

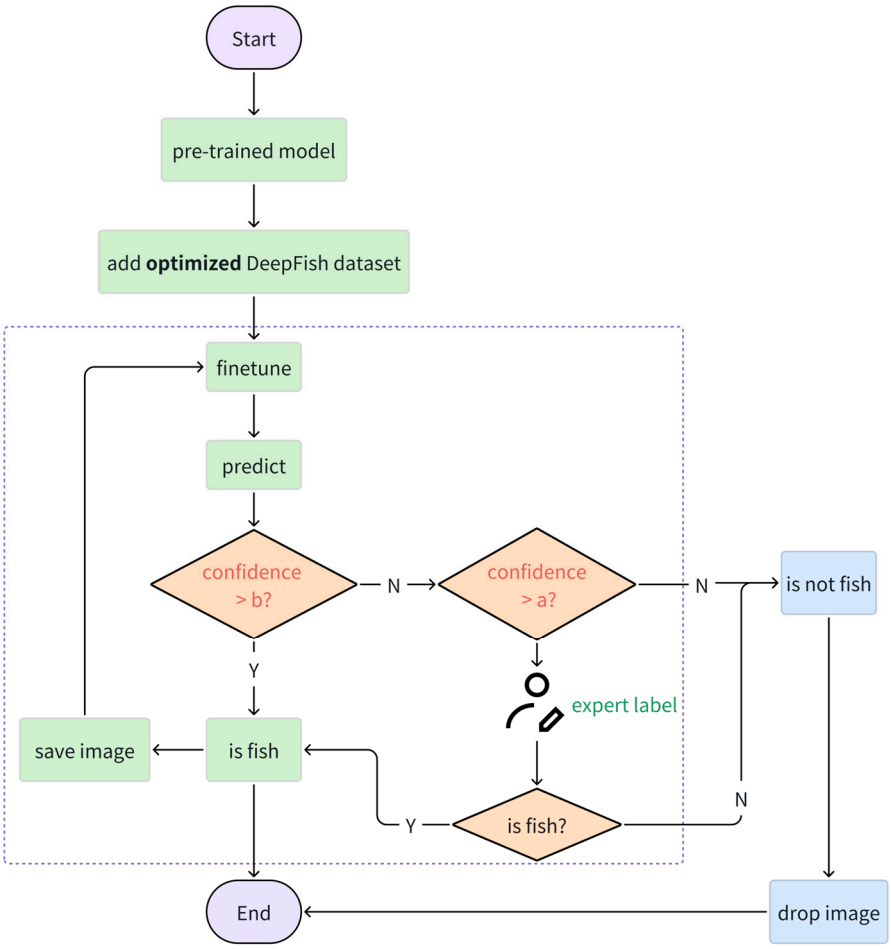


Figure 2. The visualization of the new workflow.

2.4. Metrics for Model Performance

In image segmentation, many standards are commonly used to measure the accuracy of algorithms. These standards are usually variations in pixel accuracy and IoU. Below, several commonly used precision standards for pixel-by-pixel labeling are introduced. For the sake of explanation, it is assumed that there are k + 1 classes, from L_0 to L_k, including one empty class or background. We assume p_{ij} as the number of pixels that are supposed to be classified as class i but are actually classified as class j. In other words, p_{ii} is denoted as the true positive (TP) number, while p_{ij} and p_{ji} refer to the false negative (FN) number and false positive (FP) number, respectively.

(1) Pixel Accuracy (PA): This is the simplest measure, which is the proportion of correctly labeled pixels to the total number of pixels.
(2) Mean Pixel Accuracy (MPA): It is a simple improvement of PA, which calculates the proportion of correctly classified pixels within each class and then calculates the average of all classes.
(3) Mean Intersection over Union (MIoU): The standard measure for semantic segmentation. It calculates the ratio of the intersection and union of two sets, which are the ground truth and predicted segmentation in semantic segmentation problems. This ratio can be transformed into a ratio of true, false negative, and false positive (union) to the sum of true positive (intersection). It calculates IoU on each class, followed by the calculation of an average.
(4) Frequency Weighted Intersection over Union (FWIoU): A method of improving MIoU by setting weights for each class based on its frequency of occurrence.

Among all the metrics mentioned above, MIoU has become the most commonly used due to its simplicity and strong representativeness as demonstrated by DeepFish [16].

For comparison, we used MIoU as our performance metric, the same as the DeepFish dataset [16]. The average value between the MIoU on the fish class and background class was the final MIoU metric in DeepFish.

We also used mAP as our performance metric. Two formulas, TP/(TP + FP) and TP/(TP + FN), were defined for precision and recall, respectively. There will always be a precision and a recall under a certain scene so that we can draw a curve for precision (y-axis) and recall (x-axis). AP calculates the area under a certain category of P–R curves, while mAP calculates the average area under all categories of P–R curves. There are 10 APs under different intersections to union ratios (IoU) [0.5:0.05:0.95], and the average AP under these thresholds is taken as the result, denoted as mAP@[0.5, 0.95].

3. Results

The training process for YOLOv5 and YOLOv8 is shown in Figures 3 and 4, in which the x-axis denotes the epoch number. For binary class semantic segmentation, the final best results were 0.8431 and 0.8577 under YOLOv5 and YOLOv8, respectively, for mask level mAP@[0.5, 0.95] on the validation set. The final best results under YOLOv5 and YOLOv8 were 0.9850 and 0.9949, respectively, for mask level mAP@0.5 on the validation set. For MIoU, the final best results were 0.9562 and 0.9549 under YOLOv5 and YOLOv8, respectively. The MIoU metric in DeepFish was 0.93. Therefore, we could obtain a better metric in MIoU using YOLOv5 and YOLOv8.

To verify that our training results did not overfit on the DeepFish dataset, we selected raw videos containing fish that we had previously collected in the deep sea using high-definition cameras mounted on ROVs for out-of-sample testing. The videos involved in this dataset were captured at a depth of 6000 m in the deep sea at 142.2491° E, 11.66513° N. For a video clip with a resolution of 1920 × 1080 (1080P) and a length of 7 min and 32 s, the trained YOLOv8 small-size optimal model was directly used for frame-by-frame prediction. On average, an image was processed for 77.9 milliseconds (with 5.2 milliseconds for preprocessing, 71.8 milliseconds for inference, and 0.9 milliseconds for post-processing), running on a server with 128 CPU cores, and the image size was scaled to 384 pixels in height and 640 pixels in width. Throughout the entire processing, the prediction time for a single image gradually decreased and stabilized. After the image was aligned and focused, the recognition rate and corresponding confidence of the fish were at a high level (Figure 5). This indicates that the existing models indeed have a strong generalization ability, as the data are significantly different from the DeepFish dataset used for training.

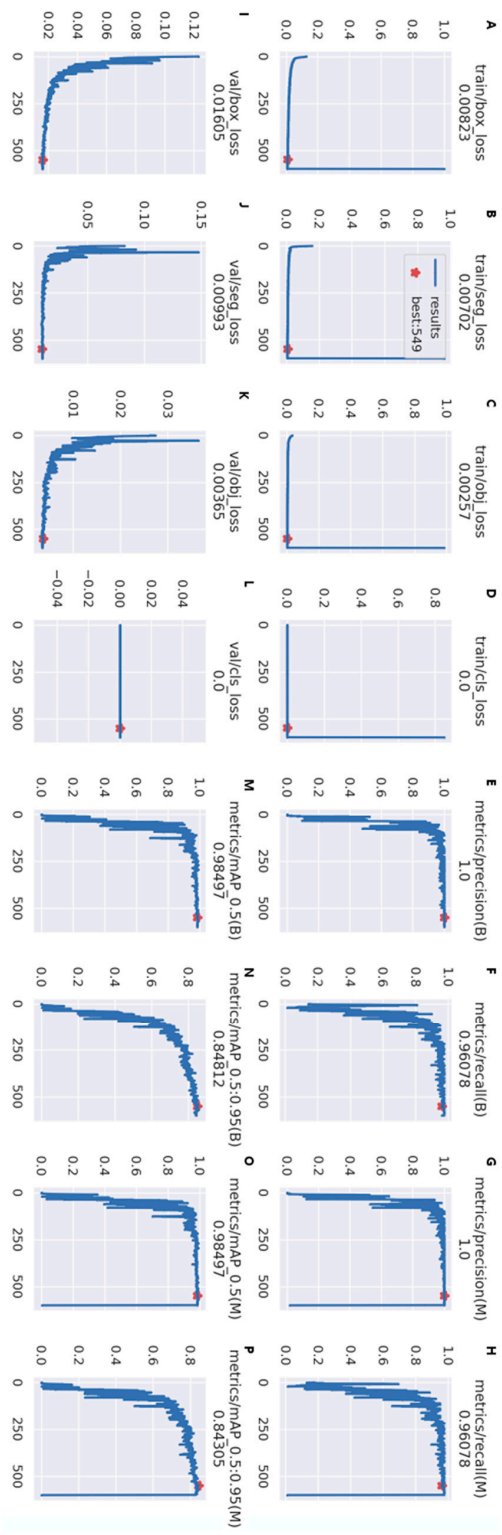


Figure 3. Whole training process for YOLOv5 binary class semantic segmentation on original DeepFish Seg dataset. (**A**) Bounding Box Loss curve in training stage. (**B**) Segmentation Loss curve in training stage. (**C**) Object Loss curve in training stage. (**D**) Classification Loss curve in training stage. (**E**) Bounding Box Precision curve. (**F**) Bounding Box Recall curve. (**G**) Mask Precision curve. (**H**) Mask Recall curve. (**I**) Bounding Box Loss curve in validation stage. (**J**) Segmentation Loss curve in validation stage. (**K**) Object Loss curve in validation stage. (**L**) Classification Loss curve in validation stage. (**M**) Bounding Box mAP50 curve. (**N**) Bounding Box mAP50-95 curve. (**O**) Mask mAP50 curve. (**P**) Mask mAP50-95 curve.

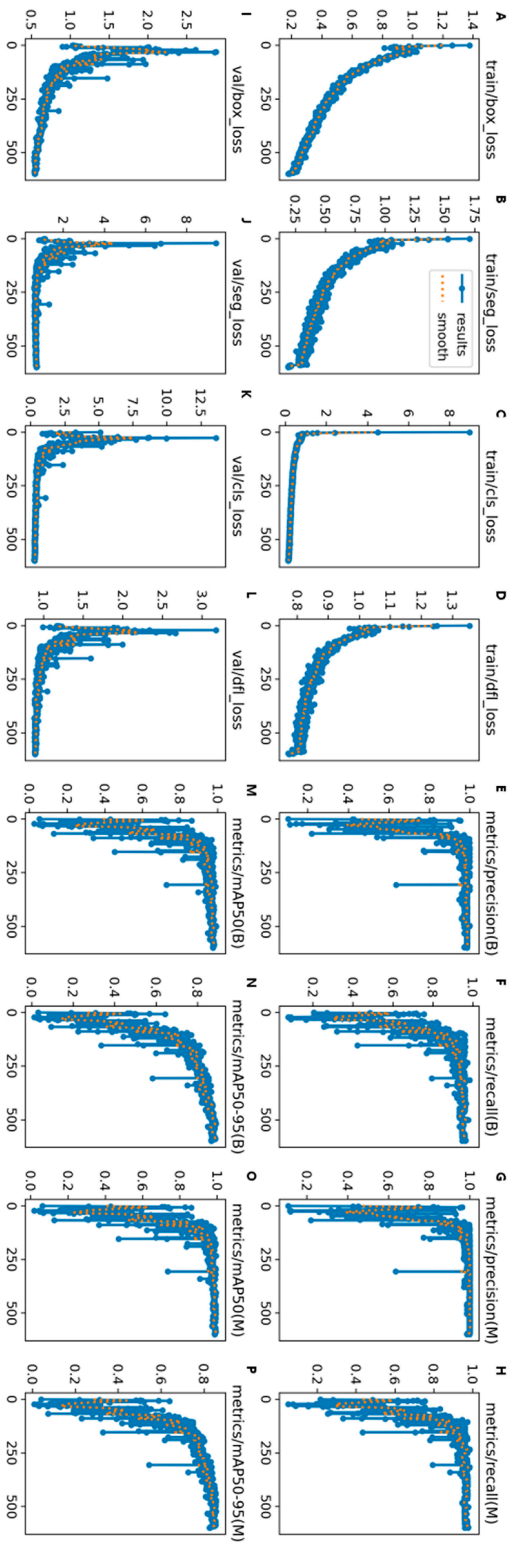


Figure 4. Whole training process for YOLOv8 binary class semantic segmentation on original DeepFish Seg dataset. (**A**) Bounding Box Loss curve in training stage. (**B**) Segmentation Loss curve in training stage. (**C**) Classification Loss curve in training stage. (**D**) Distribution Focal Loss curve in training stage. (**E**) Bounding Box Precision curve. (**F**) Bounding Box Recall curve. (**G**) Mask Precision curve. (**H**) Mask Recall curve. (**I**) Bounding Box Loss curve in validation stage. (**J**) Segmentation Loss curve in validation stage. (**K**) Classification Loss curve in validation stage. (**L**) Distribution Focal Loss curve in validation stage. (**M**) Bounding Box mAP50 curve. (**N**) Bounding Box mAP50-95 curve. (**O**) Mask mAP50 curve. (**P**) Mask mAP50-95 curve.

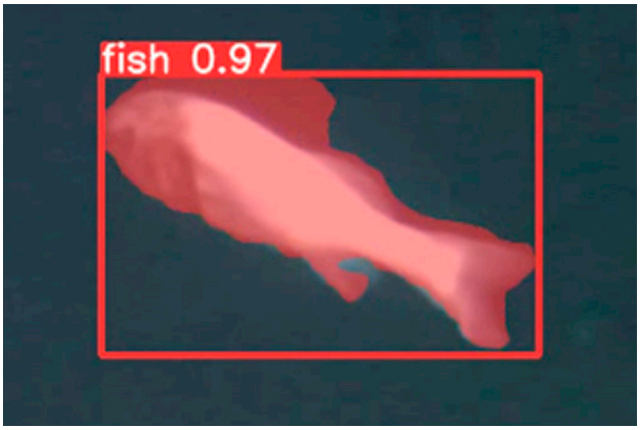


Figure 5. YOLOv8 application on deep-sea high-resolution video clip. The fish video clip was obtained by a camera mounted on an ROV at ~6000 m depth.

For multiple-class semantic segmentation, the final best results were 0.7359 and 0.9381 for mask levels mAP@0.5 and mAP@[0.5, 0.95] with 19 valid fish classes in total, respectively, under YOLOv8. Figure 6 shows the confusion matrix on the validation set.

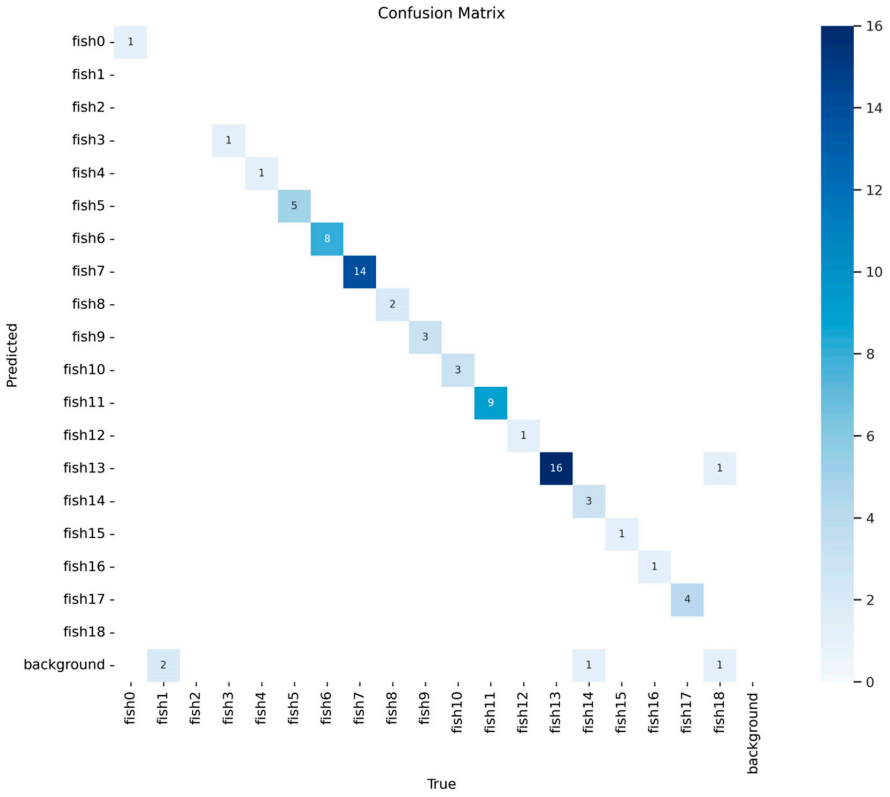


Figure 6. Confusion matrix on YOLOv8 multi-class semantic segmentation validation dataset newly labeled after training 600 epochs.

We chose a = 0.05 and b = 0.95417 as the two thresholds for the new workflow, respectively. Figures 7 and 8 show the corresponding confusion matrices.

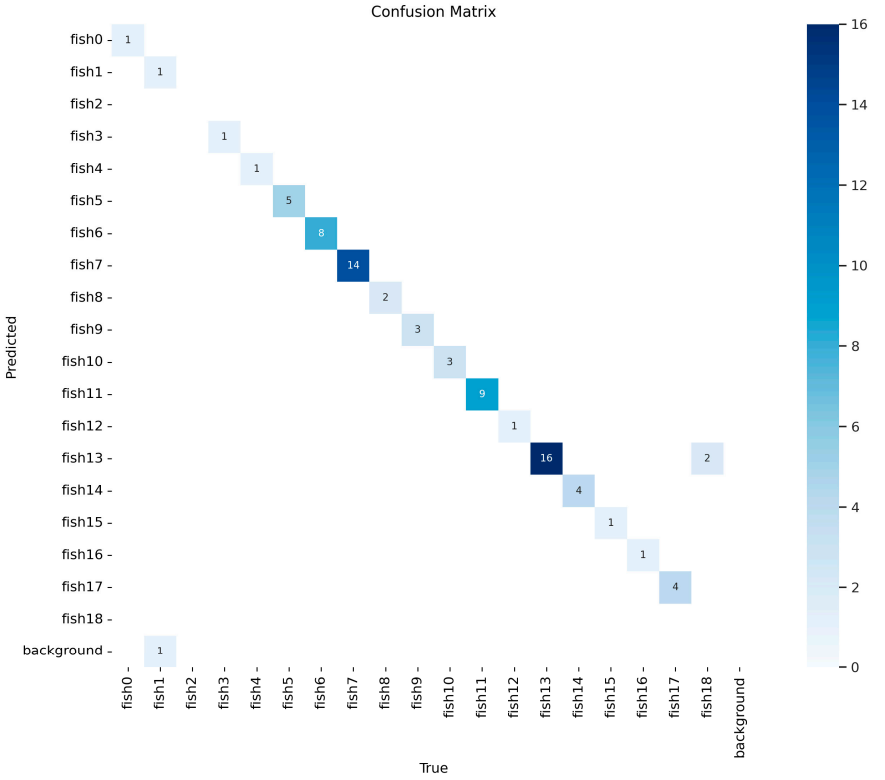


Figure 7. Confusion matrix on YOLOv8 multi-class semantic segmentation validation dataset newly labeled after training 600 epochs with confidence threshold = 0.05.

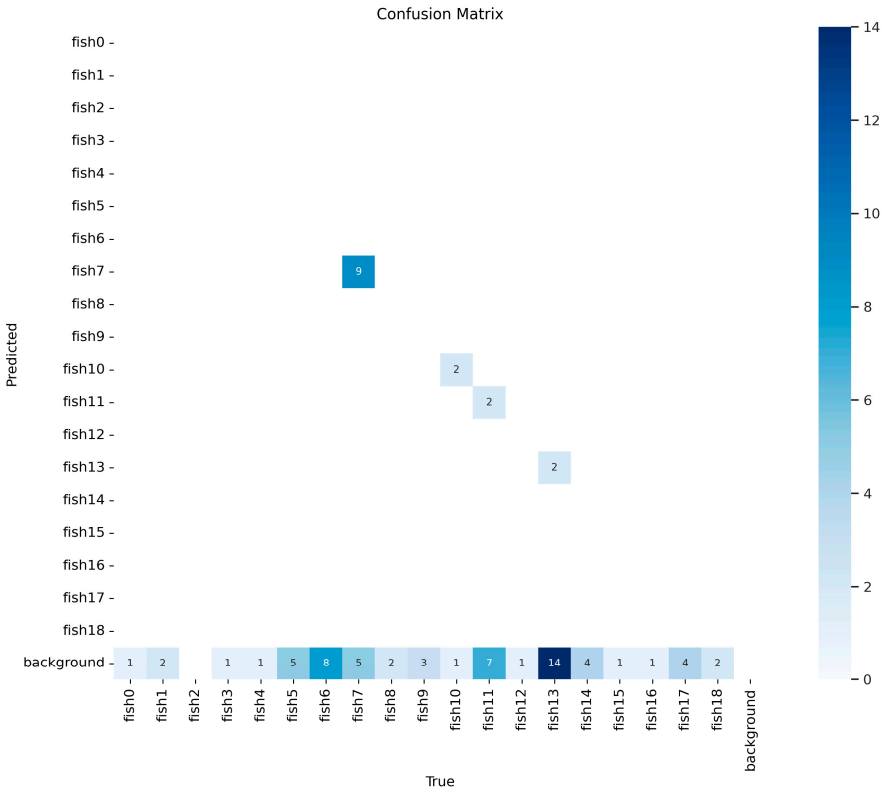


Figure 8. Confusion matrix on YOLOv8 multi-class semantic segmentation validation dataset newly labeled after training 600 epochs with confidence threshold = 0.95417.

4. Discussion

The results indicate that YOLOv5 and YOLOv8 have better performance than the original method in the DeepFish paper on the DeepFish binary class semantic segmentation dataset. And our new workflow for fish instance segmentation performed well on the extended DeepFish dataset based on the YOLOv8 fine-tuned model.

The innovation of this research lies in the application of advanced deep-learning techniques to traditional biologist fish recognition and analysis tasks [17], which can greatly reduce the cost of manual intervention to complete video annotation and classification. Due to insufficient training data for underwater fish, achieving the automation of the entire fish identification process will be a very difficult task. However, we have designed a new and self-iterative workflow that greatly simplifies and accelerates this process. Self-iterative is a process that can be continuously optimized as research progresses and the amount of original video data increases. It involves using data with a wider range of species and more refined annotations manually performed by biologists to continuously fine-tune the existing models and enhance their generalization and universality.

4.1. Limitation

The number of point-level labels in DeepFish is not sufficient. In detection tasks, YOLO-based deep-learning models usually require at least 2000 images per class to achieve good recognition results. This leads to training of the models on a small number of labels potentially falling into local optima during the convergence process, resulting in overfitting of the model to the current DeepFish dataset distribution. For example, if a type of fish in DeepFish is placed in another piece of water with different lighting angles and postures, it is likely that the currently trained model will not be able to recognize this type of fish well. In this case, we argue that the model probably does not have a strong ability to generalize out-of-real-world samples. For almost the same reason, we notice that mAP is higher in binary class segmentation and lower in multi-class segmentation tasks.

On the basis of the experimental results, we conclude that, regardless of bounding boxes or masks, the precision rate of the final training results is usually higher than the recall rate. The object marked by the model as a fish does have a ground truth value that is, indeed, a fish. However, there were still some fish that were affected by underwater image color differences, color degradation, and other factors. The object itself was mistakenly identified as the background due to a series of reasons such as almost blending with the background. Subsequently, targeted model tuning can be carried out for the small number of unrecognized cases.

It should be noted that point-level annotation and bounding box-level annotation belong to different task levels. The difficulty of point-level annotation is actually higher than that of bounding box-level annotation, and its fault tolerance is relatively lower. This results in low performance of the point-level segmentation mask of fish when completing preliminary fish recognition. As a contrast, the annotation of bounding boxes is relatively reliable. Therefore, we could not directly add point-level annotations to categories and used them as input for subsequent models to complete fine-tuning in this study.

4.2. Real-World Generalization

From the adoption of YOLOv8 on a private deep-sea high-resolution video, we can draw a conclusion that our fine-tuned model has high real-world generalization since the video is totally out of the training sample.

4.3. Comparison with Other Research

Compared with PIFS-Net, YOLOv5 and YOLOv8 have better MIoU on the DeepFish FishSeg task [4]. The MIoU of PIFS-Net is 0.9271, while those of YOLOv5 and YOLOv8 are 0.9562 and 0.9549, respectively.

4.4. Workflow Optimization

Firstly, we added additional annotations to the subset of segmentation tasks in the original DeepFish dataset, allowing for us to complete multi-class semantic segmentation tasks for multiple fish species using this dataset. Secondly, with the results obtained from training on a binary dataset using YOLOv5 and YOLOv8 combined with careful setting of reasonable thresholds, we can effectively complete the preliminary screening of fish and directly provide biologists with candidate video frames containing possible fish species for further artificial recognition and analysis. After completion of the manual annotation, we then input these annotated images with point-level labels as new categories or supplements to the existing categories into the model, continue to fine-tune, and finally complete a new self-iterative complete fish research workflow. With the appropriate threshold, we could achieve a high recall rate of $(72 - 1)/72 \approx 98.61\%$ and a normal precision rate of $15/(72 - 1) \approx 21.13\%$. The numerator of the recall rate represents the TP of fish, and the denominator represents the TP + FN of fish. The numerator of the precision rate represents the sum of TP of every category of fish, and the denominator represents the TP of fish. We could also observe that, due to inherent color differences in underwater image acquisition, natural protective colors of fish, and other factors, it is difficult for even humans to distinguish the presence of fish in the only image in the validation set that should have identified fish but did not (Figure 9). Therefore, compared to solutions that do not have automated means to identify fish, our new workflow will save biologists approximately 21.13% in identification and labeling costs for identifying marine fish.



Figure 9. The only image in the validation set that should have identified fish but did not.

5. Conclusions

YOLOv5 and YOLOv8 are reliable methods to deal with the fish instance segmentation task, as both of them can achieve better metrics than the original methods applied to deal with DeepFish. The original methods relied on a two-component approach: the first component is the ResNet-50 backbone, which extracted features from the input image, and the second component was either a feed-forward network that provided a scalar value for the entire image or an up-sampling path that generated a value for each pixel within the image. YOLOv5 and YOLOv8, building upon these foundations, have refined the process to deliver even more precise and efficient instance segmentation for fish detection. Compared to traditional annotation or detection systems, our new workflow is more intelligent and an interactive solution that integrates expert systems with rich prior knowledge and convolutional neural networks with strong feature-extraction capabilities. Our new workflow for fish instance segmentation has a great perspective in the future study of fishery survey and ecosystem monitoring, as it provides a new automation method for species recognition and statistics.

Author Contributions: J.Z. contributed to the development of the study design. J.Z. and Y.W. contributed to the writing of the manuscript. All authors contributed to the article and approved the submitted version. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 42376149.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This study is supported by the National Natural Science Foundation of China (No. 42376149) and Shenzhen Key Laboratory of Advanced Technology for Marine Ecology (ZDSYS20230626091459009).

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

References

1. Chicchon, M.; Bedon, H.; Del-Blanco, C.R.; Sipiran, I. Semantic Segmentation of Fish and Underwater Environments Using Deep Convolutional Neural Networks and Learned Active Contours. *IEEE Access* **2023**, *11*, 33652–33665. [CrossRef]
2. Saleh, A.; Sheaves, M.; Jerry, D.; Azghadi, M.R. Transformer-based Self-Supervised Fish Segmentation in Underwater Videos. *arXiv* **2022**, arXiv:2206.05390.
3. Haider, A.; Arsalan, M.; Choi, J.; Sultan, H.; Park, K.R. Robust segmentation of underwater fish based on multi-level feature accumulation. *Front. Mar. Sci.* **2022**, *9*, 1010565. [CrossRef]
4. Haider, A.; Arsalan, M.; Nam, S.H.; Sultan, H.; Park, K.R. Computer-aided fish assessment in an underwater marine environment using parallel and progressive spatial information fusion. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 211–226. [CrossRef]
5. Tarling, P.; Cantor, M.; Clapes, A.; Escalera, S. Deep learning with self-supervision and uncertainty regularization to count fish in underwater images. *PLoS ONE* **2022**, *17*, e0267759. [CrossRef] [PubMed]
6. Chen, J.; Tang, J.; Lin, S.; Liang, W.; Su, B.; Yan, J.; Zhou, D.; Wang, L.; Lai, Y.; Yang, B. RMP-Net: A structural reparameterization and subpixel super-resolution-based marine scene segmentation network. *Front. Mar. Sci.* **2022**, *9*, 1032287. [CrossRef]
7. Jahanbakht, M.; Xiang, W.; Waltham, N.J.; Azghadi, M.R. Distributed Deep Learning and Energy-Efficient Real-Time Image Processing at the Edge for Fish Segmentation in Underwater Videos. *IEEE Access* **2022**, *10*, 117796–117807. [CrossRef]
8. Holmberg, J.; Norman, B.; Arzoumanian, Z. Estimating Population Size, Structure, and Residency Time for Whale Sharks Rhincodon Typus Through Collaborative Photo-Identification. *Endangered Species Res.* **2009**, *7*, 39–53. [CrossRef]
9. Anantharajah, K.; Ge, Z.; McCool, C.; Denman, S.; Fookes, C.; Corke, P.; Tjondronegoro, D.; Sridharan, S. Local Inter-Session Variability Modelling for Object Classification. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, 24–26 March 2014; pp. 309–316.
10. Boom, B.J.; He, J.; Palazzo, S.; Huang, P.X.; Beyan, C.; Chou, H.M.; Lin, F.P.; Spampinato, C.; Fisher, R.B. A Research Tool for Long-Term and Continuous Analysis of Fish Assemblage in Coral-Reefs Using Underwater Camera Footage. *Ecol. Inf.* **2014**, *23*, 83–97. [CrossRef]
11. Kavasidis, I.; Palazzo, S.; Salvo, R.D.; Giordano, D.; Spampinato, C. An Innovative Web-Based Collaborative Platform for Video Annotation. *Multimedia Tools Appl.* **2014**, *70*, 413–432. [CrossRef]
12. Cutter, G.; Stierhoff, K.; Zeng, J. Automated Detection of Rockfish in Unconstrained Underwater Videos Using Haar Cascades and a New Image Dataset: Labeled Fishes in the Wild. In Proceedings of the 2015 IEEE Winter Applications and Computer Vision Workshops, Waikoloa, HI, USA, 6–9 January 2015; pp. 57–62.
13. Jäger, J.; Simon, M.; Denzler, J.; Wolff, V.; Fricke-Neuderth, K.; Kruschel, C. Croatian Fish Dataset: Fine-grained classification of fish species in their natural habitat. In Proceedings of the Machine Vision of Animals and their Behaviour (MVAB), Swansea, UK, 7–10 September 2015. [CrossRef]
14. Ditria, E.M.; Connolly, R.M.; Jinks, E.L.; Lopez-Marcano, S. Annotated Video Footage for Automated Identification and Counting of Fish in Unconstrained Seagrass Habitats. *Front. Mar. Sci.* **2021**, *8*, 629485. [CrossRef]
15. Lopez, S. slopezmarcano/automated-Fish-Detection-in-Low-Visibility: Automated Fish Detection in Low Visibility. 2021. Available online: https://zenodo.org/records/5238512 (accessed on 10 December 2023).
16. Saleh, A.; Laradji, I.H.; Konovalov, D.A.; Bradley, M.; Vazquez, D.; Sheaves, M. A Realistic Fish-Habitat Dataset to Evaluate Algorithms for Underwater Visual Analysis. *Sci. Rep.* **2020**, *10*, 14671. [CrossRef] [PubMed]
17. González-Sabbagh, S.; Robles-Kelly, A. A Survey on Underwater Computer Vision. *ACM Comput. Surv.* **2023**, *55*, 1–39. [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.