Journal of *Marine Science and Engineering*

MDPI

Article

Performance Comparison of Sea Cucumber Detection by the Yolov5 and DETR Approach

Xin Yuan [1], Shutong Fang [1], Ning Li [2,*], Qiansheng Ma [1], Ziheng Wang [1], Mingfeng Gao [3], Pingpeng Tang [1], Changli Yu [1], Yihan Wang [4,*] and José-Fernán Martínez Ortega [5]

[1] School of Ocean Engineering, Harbin Institute of Technology, Weihai 264200, China; xin.yuan@upm.es (X.Y.); shutongfang@stu.hit.edu.cn (S.F.); qianshengma@outlook.com (Q.M.); william_wangzh@outlook.com (Z.W.); tpp@hit.edu.cn (P.T.); yuchangli@hitwh.edu.cn (C.Y.)
[2] School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264200, China
[3] School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China; mingfeng.gao_bjtu@outlook.com
[4] College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China
[5] Department of Telematic Engineering and Electronic (DTE), School of Telecomunications Systems and Engineering (ETSIST), Techinical University of Madrid, 28031 Madrid, Spain; jf.martinez@upm.es
* Correspondence: li.ning@upm.es (N.L.); wangyihan@heu.ntesmail.com (Y.W.); Tel.: +86-158-9063-0793 (N.L.); +86-182-0106-9375 (Y.W.)

Abstract: Sea cucumber detection represents an important step in underwater environmental perception, which is an indispensable part of the intelligent subsea fishing system. However, water turbidity decreases the clarity of underwater images, presenting a challenge to vision-based underwater target detection. Therefore, accurate, real-time, and lightweight detection models are required. First of all, the development of subsea target detection is summarized in this present work. Object detection methods based on deep learning including YOLOv5 and DETR, which are, respectively, examples of one-stage and anchor-free object detection approaches, have been increasingly applied in underwater detection scenarios. Based on the state-of-the-art underwater sea cucumber detection methods and aiming to provide a reference for practical subsea identification, adjacent and overlapping sea cucumber detection based on YOLOv5 and DETR are investigated and compared in detail. For each approach, the detection experiment is carried out on the derived dataset, which consists of a wide variety of sea cucumber sample images. Experiments demonstrate that YOLOv5 surpasses DETR in low computing consumption and high precision, particularly in the detection of small and dense features. Nevertheless, DETR exhibits rapid development and holds promising prospects in underwater object detection applications, owing to its relatively simple architecture and ingenious attention mechanism.

Keywords: underwater target detection and recognition; YOLOv5; DETR; sea cucumber fishing

Citation: Yuan, X.; Fang, S.; Li, N.; Ma, Q.; Wang, Z.; Gao, M.; Tang, P.; Yu, C.; Wang, Y.; Martínez Ortega, J.-F. Performance Comparison of Sea Cucumber Detection by the Yolov5 and DETR Approach. *J. Mar. Sci. Eng.* **2023**, *11*, 2043. https://doi.org/10.3390/jmse11112043

Received: 22 September 2023
Revised: 18 October 2023
Accepted: 21 October 2023
Published: 25 October 2023

1. Introduction

Currently, with the development of automatic intelligent aquaculture and fishing technology and the improvement of human beings' living standards, the demand for aquatic products is increasing gradually. One of these products is sea cucumbers, with their high nutritional value and memory-enhancing and anti-tumor effects. The sea cucumber aquaculture industry has become a major industry in certain coastal areas, bringing increases in the income of fishermen and also promoting the development of secondary and tertiary industries, such as processing and transportation. However, there are various problems that make the fishing of sea cucumbers troublesome. Sea cucumbers only have two dormant periods in a year; thus, fishing operations can only be performed in spring and autumn. Additionally, due to the presence of abundant reefs in the living environment of sea cucumbers, it is unsuitable to use fishing nets for their capture. The fishing operations

of sea cucumbers in most marine pastures are conducted by professional staff, who are supposed to put on oxygen masks and dive into the seabed. This traditional method of artificial fishing requires high levels of skills and includes the risk of various occupational diseases due to the low temperatures of seawater in spring and autumn, the frequent changes in water pressure during diving and surfacing, and the complex seabed working environment. Therefore, replacing manual work with intelligent underwater robots that can capture sea cucumbers automatically has become the development trend [1].

Perceiving the underwater environment is an integral part of intelligent underwater fishing robot systems, such as Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) [2]. The system perceives information on the subsea environment through acoustic or optical sensors and then takes corresponding actions based on the surrounding area. Therefore, the autonomous detection of underwater sea cucumbers is a necessary step for subsea robots to localize and capture sea cucumbers automatically. High resolution and rich information make underwater optical imaging the most intuitive and commonly used method for data acquisition. Nevertheless, the turbidity and poor light transmittance of water can cause a crucial decline in the clarity of underwater imageries, presenting difficulties in the application of vision-based underwater target detection [3]. Turbidity is often encountered in sea cucumbers' complex living environment. The direction of light transmission is affected by the scattering and absorption of water and various organic and inorganic suspended particles like fish and sediment, which results in image distortion, including blurred target features, color variance, and severe distortion. In addition, the low light conditions lead to the retrieval of limited effective target light information. Thus, current research works on underwater vision are mainly focusing on scenarios with good water conditions. Moreover, marine animals like sea cucumbers are usually small in size, making it difficult to detect and recognize them. Therefore, research on feature detection and target recognition in complex and changeable underwater areas is challenging but essential.

Target detection in underwater areas needs to consider image restoration and enhancement. Compared with that in an atmospheric environment, traditional detection approaches and deep-learning-based methods are mainly taken into account. In the classic methods, the regions of interest are first selected through sliding windows, and features within them are extracted through conventional algorithms including Scale-Invariant Feature Transform (SIFT) [4], Histograms of Oriented Gradients (HOG) [5], etc. Then, machine-learning algorithms such as Support Vector Machine (SVM) [6] are applied to classify the extracted features and determine whether the region contains targets [7]. Deep-learning-based approaches study image sets by training neural networks and establishing logical relationships to enhance the clarity of the image and extract target features for intelligent recognition [8]. Other methods for underwater target detection are also studied, including sonar imaging [9], laser imaging, and polarization imaging [10,11].

The main contributions of this present work are summarized as follows:

1. The state-of-the-art underwater sea cucumber detection methods are summarized, including traditional methods, one-stage methods based on deep learning such as You Only Look Once (YOLO) series algorithms and Single Shot MultiBox Detector (SSD), two-stage methods based on deep learning such as R-CNN series algorithms, anchor-free approaches such as DEtection TRansformer (DETR), and other methods.
2. For the detection of sea cucumbers, fundamentals of YOLOv5 and DETR are first introduced. Then, in the training process, the test results of YOLOv5 and DETR and a performance comparison of these two approaches are presented, proving the excellent performance of YOLOv5 and DETR in underwater adjacent and overlapping sea cucumber detection.
3. Relevant research methods and the latest achievements on underwater target detection are systematically collated, and then experiments based on YOLO and DETR are carried out on the derived sea cucumber datasets.

The rest of the manuscript is organized as follows: Section 2 briefly describes the research related to underwater target detection and its recent developments. The fundamentals of YOLOv5 and DETR are described in Section 3. Section 4 demonstrates a detection performance comparation of YOLO and DETR on the sea cucumber dataset. Finally, in Section 5, conclusions are drawn, and current problems are discussed to provide a reference for future work.

2. Related Works

With the development of underwater image-processing and target detection techniques, many conventional algorithms and frameworks have been developed. Conventional target detection approaches extract the features of target zones manually, which is time-consuming and has poor robustness. For most of the conventional methods, the candidate regions are first selected through different sizes of sliding windows. Then, features in these windows are extracted. Finally, machine-learning algorithms are applied for recognition. Classic algorithms such as HOG [5] and Deformable Part Model (DPM) [12] have some limitations. The region selection strategy is not targeted, which leads to high computation consumption and window redundancy. Additionally, artificially designed methods are not as robust as required considering feature diversity [13].

With the emergence of the deep convolutional neural network, great breakthroughs have been achieved in object detection algorithms. Generally, approaches based on deep learning outperform those traditional approaches, most of which demand manual intervention. Thus, deep-learning methods are more suitable to deploy on underwater robots. Existing target detection algorithms are mainly divided into two categories: region proposal-based ones, also known as two-stage algorithms, and regression-based ones, also referred to as one-stage algorithms [14]. Two-stage algorithms first extract the proposed regions from the images and then classify and regress them to obtain the detection result, mainly including the use of algorithms based on Region Convolutional Neural Networks (RCNNs) [15], Fast RCNN [16], and Faster RCNN [17]. There are other two-stage networks that have been improved upon based on the above algorithms, such as Region-based Fully Convolutional Networks (R-FCNs) [18], Mask R-CNN [19], and Cascade R-CNN [20]. Two-stage algorithms can obtain more accurate detection results, but the processing time increases accordingly. Single-stage algorithms improve the detection speed by detecting and localizing the targets directly from the whole image. The main representatives of single-stage algorithms are the SSD [21] algorithm and YOLO algorithms (YOLO [22], YOLOv2 [23], YOLOv3 [24], YOLOv4 [25], YOLOv5 [26]). With continuous upgrades and innovations, the current single-stage target detection algorithms can take the precision of detection into account and guarantee the processing time as well.

Deep-learning-based approaches demonstrate good performance, but they also have some limitations, since the accuracy of detection is influenced by the image quality, and deep-learning approaches are only applicable in waters like those in the training set images. Therefore, it is necessary to combine good image restoration methods and deep-learning algorithms to make underwater target detection more effective. Thomas et al. [27] created a fully connected convolution neural network for underwater image defogging. By using the depth frame of the encoder-decoder to integrate low- and high-level features, the network was able to effectively restore blurred imageries. A method of recovered images was proposed by Martin et al. [28]. It combined image enhancement, image recovery, and a convolutional neural network. To address the issue of the maximum number of green pixels in underwater images, they proposed the Under Dark Channel Prior method (UDCP)-based Energy Transmission Restoration (UD-ETR) method to process green-channel images and obtain the recovered images. The Sample-Weighted Hyper Network (SWIPENet) was proposed by Chen et al. in 2020 [29] to cope with the blurring of underwater images in the context of severe noise interference, the architecture of which comprises many semantic rich and high-resolution hyper feature maps inspired by the Deconvolutional Single Shot Detector (DSSD) [30]. In [31], Dana et al. introduced an innovative approach for enhancing

the colors of single underwater images. They employed a physical image formation model, distinguishing themselves from previous research. Various Jerlov water types were used to estimate transmission values through a haze-lines model. Ultimately, color corrections were applied using the same physical image formation model, and the optimal outcome was elected from the diverse water types considered.

Weibiao Qiao et al. [32] introduced a novel method for the real-time and precise classification of underwater targets in 2021. They employed Local Wavelet Acoustic Patterns (LWAP) in conjunction with multi-layer perceptron (MLP) neural networks to tackle the challenges associated with underwater passive target classification, addressing issues related to heterogeneity and classification difficulty. A lightweight deep neural network was introduced in [33], aiming to simultaneously learn color conversion and target detection from subsea imageries. To mitigate color distortion, an initial step involves employing an image color conversion module to transform color images into grayscale ones. Subsequently, object detection is carried out on the converted grayscale images. This joint learning process involves optimizing a combined loss function. Xuelong Hu et al. incorporated the Pyramid Attention Network (PAN) into Feature Pyramid Networks (FPN) [34] in [35], augmenting it to produce a diverse multi-scale feature architecture. This enhanced feature structure was subsequently employed in an uneaten feed pellet detection model tailored for aquaculture applications. Experiments demonstrate a substantial increase in mean average precision (mAP) by 27.21% when compared with the baseline YOLO-v4 method [25]. To address the challenge of a constrained dataset, Lingcai Zeng et al. [36] introduced an innovative approach by incorporating an Adversarial Occlusion Network (AON) into the conventional Faster R-CNN detection algorithm. This methodology proved effective in augmenting the training data and enhancing the detection capabilities of the network. Taking inspiration from the shortcut connections observed in residual neural networks [37], Fang Peng et al. introduced a novel approach, the Shortcut Feature Pyramid Network (S-FPN) in [38], the primary aim of which is to enhance an existing strategy for multi-scale feature fusion, particularly for holothurian detection.

Through the exploration of enhancement strategies for simulated overlapping, occlusion, and blurred objects, Weihong Lin et al. devised a practical generalized model in [39], aimed at addressing challenges related to the overlapping, occlusion, and blurring of underwater targets. The Super-Resolution Convolutional Neural Network (SRCNN) is a super-resolution technique that relies on pure convolutional layers [40]. In the context of underwater imaging in low-light conditions, SRCNN has been utilized to enhance the quality of captured images [41]. To derive the low-resolution components, the raw data underwent iterative processing involving total variation regularization [42]. An approach based on YOLO was introduced in [43], aiming to safeguard rare and endangered species or eradicate invasive exotic species. It was designed to effectively classify objects and count their quantities in consecutive underwater video frames. This involved aggregating object classification outcomes from preceding frames to the current frame. In [44], Minghua Zhang et al. introduced a Multi-scale Attentional Feature Fusion Module (AFFM) designed to blend semantic and scale-inconsistent features across various convolution layers. In [45,46], an innovate method merging multi-scale features across different channels was introduced, which is achieved through various kernel sizes or intricate connections within a single block. This approach enhances the neural network's capacity for representational learning by emphasizing multi-scale feature fusion within a single block, as opposed to feature fusion through multiple stages of the backbone network.

Li et al. in [47] introduced the GBH-YOLOv5 model designed for detecting defects in photovoltaic panels. This model incorporates the BottleneckCSP module, which integrates a small target prediction head to improve the detection of smaller features and utilizes the Ghost convolution to optimize inference speed. In the detection of unique fish, Li et al. proposed the CME-YOLOv5 network model in [48], achieving 92.3% mAP. Wen et al. introduced the YOLOv5s-CA with 80.9% mAP in [49], a model that improves upon the initial C3 module by integrating a higher quantity of Bottleneck modules. Additionally,

it sequentially integrates attention-based CA and SE modules to further enhance the YOLOv5s model. In [50], Yu et al. introduced an innovative approach called the Multi-Attention Path Aggregation Network (APAN) with 79.6% mAP. This method incorporates a multi-attention mechanism to enhance the precision of detecting multiple underwater objects. Despite the various improved algorithms proposed by researchers, additional research and refinement are needed to evaluate their suitability for intricate and dynamic underwater environments.

Other methods for underwater target detection are also being explored, such as sonar imaging, laser imaging, polarization imaging, and electronic communications. In forward-looking sonar imaging, the geometry, grayscale, and statistical features of some interferences are like those of the targets, which brings about difficulty for target detection. Thus, an underwater linear target detection technology was proposed by Liu [51], combining Hough transform and threshold segmentation, which can effectively extract linear objects. The research in [52] introduces a tracking filter designed to combine Ultrashort-base Line (USBL) measurements and acoustic image measurements. This approach could achieve dependable underwater target tracking.

3. Theoretical Background

3.1. Fundamentals of YOLOv5

YOLO series algorithms are typical representatives of the one-stage methods, also known as regression-based object detection algorithms. The latest version, YOLOv7 [53], adds the ability to predict the attitude of key points; however, there is no attitude data on the tested dataset, and the attitude is also independent of the experimental task. Further, there are more cases applying YOLOv5. Therefore, YOLOv5 was selected for the experiment. It was proposed by Glenn Jocher in 2020 [20]. It outperforms the preceding YOLO series algorithms in flexibility and speed and proves its strong superiority in terms of rapid deployment of the model.

3.1.1. Model Structure

The central concept of YOLOv5 remains the same as that of previous versions, which is to first divide images into regions and then to predict bounding boxes and probabilities for individual area. The results are then refined based on anchor boxes, and Non-Maximum Suppression (NMS) is adopted to discard overlapping detections. There are four versions of YOLOv5: YOLOv5x, YOLO5l, YOLO5m, and YOLO5s, different in depth and width, of which YOLOv5s is the smallest in network depth and width. YOLOv5 mainly comprises four primary components: an input module, a backbone network for feature extraction, a neck network for feature fusion from various scales, and a head network for the prediction of detection results, as depicted in Figure 1.

Input images are pre-processed in the input module of YOLOv5. Preprocessing methods such as mosaic data augmentation, adaptive anchor box optimization, and adaptive image scaling are adopted at this stage to enhance the model's robustness. Mosaic data augmentation mixes four original images into one based on random scaling and clipping, enriching the dataset, and bolstering the detection performance of small objects such as sea cucumbers. Adaptive anchor box optimization is also a crucial step that is incorporated into the YOLOv5 code that enables the calculation of optimal anchor boxes in various training. The original anchor frames serve as the basis for the output prediction frames, which are subsequently contrasted with the ground truth to calculate the loss functions.

The backbone network handles object extraction and mainly consists of three structures: Focus, CSPDarknet53, and SPP. An improved network backbone leads to better feature extraction. The primary purpose of the Focus structure is to slice the input images and expand the channel count. This slicing effectively reduces the size of the feature map without any information loss. CSPDarknet53 is the main structure of YOLOv5, combining the main structure of YOLOv3 and Darknet53 with CSPNet. Figure 2 demonstrates the structure of BottleneckCSP in CSPNet. This structure enhances the learning ability of the

network and effectively decreases processing costs. By dividing the input matrix into two groups along the channel dimension, the number of parameters is significantly decreased, resulting in a lightweight network with high accuracy. SPP employs multi-scale feature fusion by utilizing 1×1, 5×5, 9×9, and 13×13 maximum pooling.

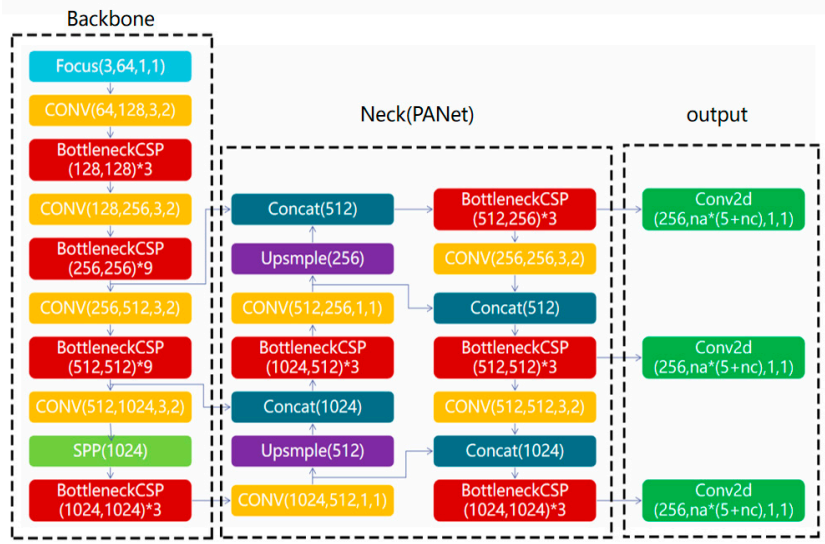


Figure 1. The main network structure of YOLOv5, including the backbone network, the neck network, and the head network (the output module).

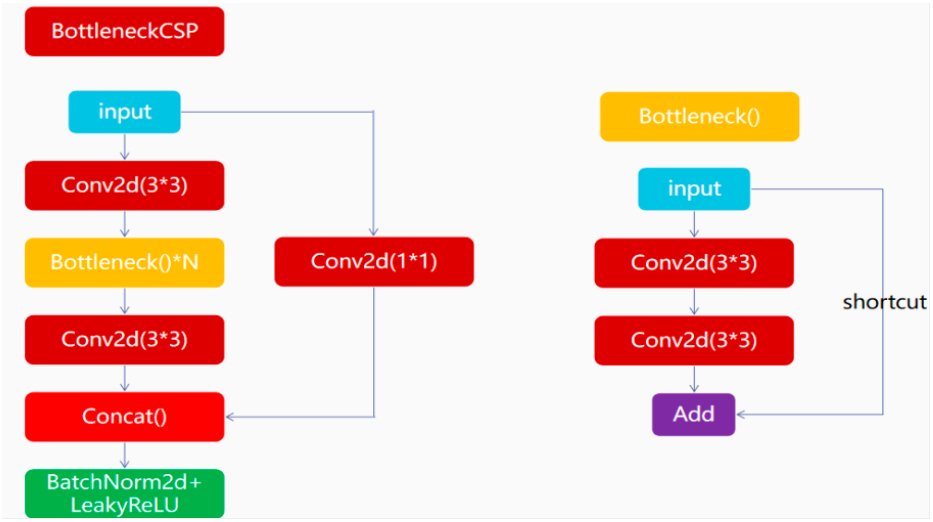


Figure 2. The main structure of BottleneckCSP.

The neck network adopts the structure of FPN connected with PAN, for further improving the feature fusion capability and the robustness of the network. In YOLOv5, in addition to the bottom-up transmission mode in the FPN structure used in YOLOv3, a top-down transmission mode is added. This allows features at different levels to be merged, enabling low-level feature maps with small receptive fields to be combined with high-level feature maps with large receptive fields, and vice versa. The head network completes the output of object detection results.

Additionally, compared to previous YOLO series algorithms, YOLOv5 has also made improvements in the activation function by replacing the Rectified Linear Unit (ReLU) activation equation in the network with the Mish activation function, as calculated in Equation (1). Though the ReLU function performs well in gradient descent, it is not smooth at $x = 0$ and can cause gradient vanishing. In contrast, the Mish function is smoother than

ReLU. To ensure gradient descent functionality, Mish is basically like ReLU in the positive interval, while still having a certain gradient size in a local part of the negative interval. This is beneficial for backpropagation and enhances the feature transmission ability of the network. Compared with ReLU, Mish does not suffer from the gradient vanishing problem when x is lower than 0, which can improve the network's execution power.

$$f(x) = x * tanh(\ln(1 + e^x)) \tag{1}$$

3.1.2. Loss Function

The loss function plays a pivotal role in deep learning, as it quantifies the discrepancy between the predicted and ground truth results of the learning mode. The selection of the loss function holds crucial significance in supervised learning. During the training process, the model parameters can be continuously adjusted to minimize the loss function, thereby enhancing the model's effectiveness. The loss function of YOLOv5 consists of three parts, which are bounding box loss, object loss, and classification loss, basically the same as previous YOLO algorithms. In YOLOv5, the bounding box loss is computed by using CIOU loss, and the object loss and classification loss are computed through BCE loss, as described in Equations (2) and (3):

$$\begin{aligned} L = {} & \lambda_{\text{coord}} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i) - (1 - CIOU) - \\ & \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} \left[\hat{C}_i log(C_i) + \left(1 - \hat{C}_i\right) log(1 - C_i) \right] - \\ & \lambda_{\text{noobj}} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{noobj} \left[\hat{C}_i log(C_i) + \left(1 - \hat{C}_i\right) log(1 - C_i) \right] \\ & - \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} \sum_{c\in classes} \left[\hat{p}_i(c) log(p_i(c)) + \left(1 - \hat{p}_i(c)\right) log(1 - p_i(c)) \right] \end{aligned} \tag{2}$$

where λ_{coord} refers to the positive sample weight coefficient, I_{ij}^{obj} denotes 1 for positive samples and 0 for negative ones, and w_i and h_i indicate the width and height of the bounding box, respectively.

$$CIOU = IOU - \frac{\rho^2\left(b, b^{gt}\right)}{c^2} - \beta v, \tag{3}$$

where c implies the diagonal length of the smallest closure area, which contains both the bounding box and ground truth, β measures the consistency of the aspect ratio, and v is the trade-off parameter.

According to the fundamentals of YOLOv5 and its improvements over the previous YOLO series algorithms, YOLOv5 can outperform two-stage detection algorithms based on a deep-learning approach in terms of real-time detection, and it has certain advantages in both accuracy and speed over conventional YOLO series algorithms. Combining the features of diverse sizes of receptor fields based on the feature pyramid structure and applying a large number of convolutional layers in its backbone network, YOLOv5 behaves better when identifying small targets and extracting subtler and deeper features from blurry images, which will be proved through the comparison in Section 4 in this manuscript. YOLOv5 is composed of YOLOv5s, YOLOv5m, and YOLOv5l based on the size of the network. The larger the size of YOLOv5l, the better the network detection performance, but meanwhile, it requires higher computer performance. When applying the trained model in the practical underwater project, considering the low computing performance of embedded computers, YOLOv5s, a lightweight version of YOLOv5, is adopted. In addition, an artificial intelligence embedded system developed by NVIDIA achieves excellent computing performance and low power consumption.

3.2. Fundamentals of DETR

DEtection Transformer (DETR) is a target detection algorithm rooted in the Transformer architecture, mainly relying on an attention mechanism for target detection, which differs from previous mainstream object detection algorithms. Both two-stage target detection algorithms exemplified by YOLO, represented by Faster R-CNN and one-stage target detection algorithms, are based on an anchor and need to undergo the Non-Maximum Suppression (NMS) operation after detection, but such complex operations decrease the overall detection speed. Nevertheless, the DETR model does not require anchors or the NMS process and can perform object detection tasks using only the Transformer framework. The main process of DETR is similar to that of Transformer. First, a convolutional neural network (CNN) is employed to extract image features, and the images are encoded into sequences. Then, position encoding is added to complete the entire encoding of the input information. The encoder-decoder model is the main structure of DETR, which mainly handles extracting global features. Due to the position encoding in the input sequence, the model can associate features with positions and 'understand' the input information, which then allows the model to determine the positions of different objects. The decoder structure mainly generates a bounding box to determine whether there are objects in the bounding box. Generally, 100 decoders are set to generate 100 bounding boxes. The main flowchart is demonstrated in Figure 3.

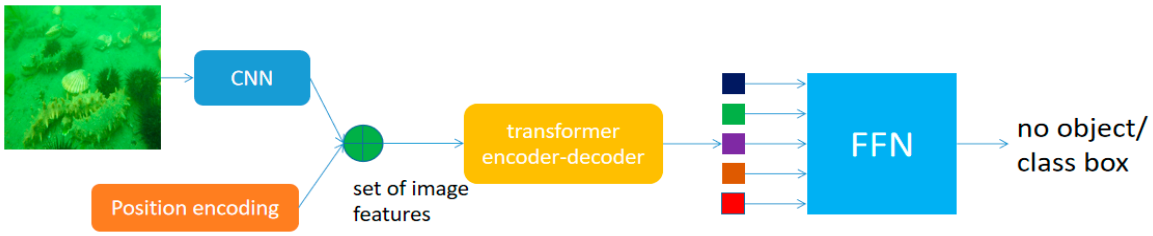


Figure 3. Main flowchart of the DETR model.

3.2.1. Position Encoding

When the model processes data, it first initializes the input image information into a sequence according to certain rules, and the image information is converted into a sequence through convolutional neural networks. Meanwhile, to reflect the positional relationship of different sequences in the entire input, positional encoding is introduced, which reflects the position of the input sequence through positional encoding, allowing the network model to have a certain 'understanding' of the overall information and positional relationship. Position encoding is embedded in the sequence during input initialization and output initialization, and the dimensions of position encoding are generally consistent with the sequence dimensions and can be added directly.

There are many ways to set up position encoding, and the most commonly used method is to apply sine or cosine function encoding, because sine and cosine functions can output different encodings for sequences at various positions and have strong generalization ability. The rules for positional encoding here are shown in Equations (4) and (5). The encoding should ensure that adjacent sequences have the same step size in the function. Among them, pos represents the position of the sequence, d_{model} denotes the dimension size of the encoded sequence, and i means the dimension of the encoded position in the sequence. During encoding, odd dimensions are encoded using a sine function, and even dimensions are encoded by a cosine function. Their frequency is $1/\left(10,000^{\frac{2i}{d_{model}}}\right)$, and the frequency declines as the dimension increases. This encoding method improves the generalization ability of encoding in different dimensions.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10,000^{\frac{2i}{d_{model}}}}\right), \tag{4}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10,000^{\frac{2i}{d_{model}}}}\right), \tag{5}$$

3.2.2. Encoder-Decoder Structure

The main structure of DETR is the encoder-decoder structure, which is depicted in Figure 4. The encoder primarily consists of a multi-head attention layer like the convolutional layer and a feed-forward network, and the decoder mainly contains a feed-forward network and a masked multi-head attention layer. Both internal structures contain operations similar to shortcuts. This connection method actually draws on the idea of ResNet; in this manner, the network's generalization ability is improved. After the shortcut is added, the regularization operation is added to make the training easier to converge, thus simplifying the training difficulty and improving the overall quality of the network training.

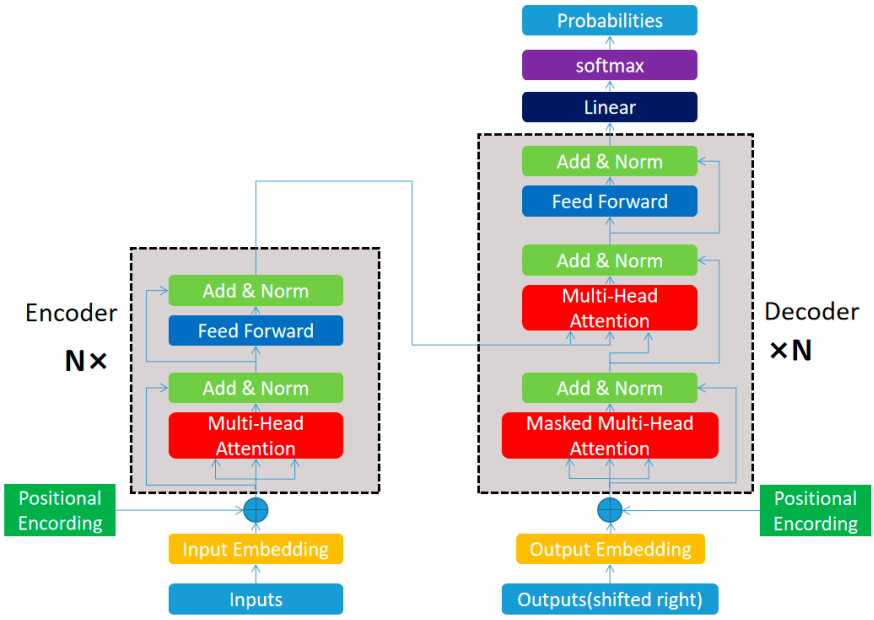


Figure 4. Structure of the encoder-decoder model.

The attention mechanism is achieved by setting three matrices. In an attention module, the model multiplies each sequence by three matrices to derive three vectors, namely, query, key, and value. The combination of these three vectors for each sequence is the query matrix Q, key matrix K, and value matrix V. The value matrix V is mainly used, and the conversion process is demonstrated in Equation (6). Among them, X is the matrix composed of input sequences, and W^Q, W^K, and W^V are the transformation matrices.

The value matrix V is mainly used in the three matrices. After each attention module, the value matrix is iterated once. The iterative equation is provided in Equation (7), and the results are passed to the following operation modules. From this, we can see that each value vector in the new value matrix is the result of the weighted addition of the original value vectors, which is the embodiment of the attention mechanism.

$$Q = X * W^Q,\ K = X * W^K,\ V = X * W^V, \tag{6}$$

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{7}$$

Of course, just one set of Q, K, and V is not enough. Transformer mainly uses a multi-head attention module, which sets multiple sets of W^Q,W^K, and W^V to obtain multiple sets of results. This idea is actually similar to the idea of a CNN setting multiple channels. After

obtaining multiple sets of iterated sequences, they are merged into a feed-forward network, which is equivalent to a fully connected layer and can condense features. Then, the same repeated process is employed to extract deeper global features.

The attention mechanism used by the decoder performs operations using the V of the decoder and the Q and K of the encoder, connecting the encoder and decoder to achieve overall operations. The other operations and structure of the decoder are roughly the same as those of the encoder. The main task of the encoder is to obtain the attention of each target, and it will distinguish different targets very clearly. Even if there are some occlusion phenomena, this will not have a significant impact. Waiting for the recognition task of the decoder, one advantage of using the encoder and CNN is that it can make the model clearer about the specific location of the predicted target. The decoder initializes 100 vectors, and it is responsible for predicting 100 coordinate boxes. The initialization form of the vectors is the sum of 0 and position encoding, which is equivalent to indirectly assigning each decoder its main area of responsibility, making the decoder sensitive to position first.

4. Methods

4.1. Dataset and Evaluation Metrics

The deep-learning-based underwater sea cucumber detection requires a large number of imageries for training. The experimental dataset of sea cucumbers is provided by Shandong Future Robot Co., Ltd. (Weihai, Shandong) [54]. It contains a wide variety of sea cucumber sample images. The targets in the dataset are labeled, and images that do not contain the detected targets are removed. A total of 3271 valid images are retained, the resolution of which is 416 × 416 pixels. With a total of more than 10,000 ground truth data for training, all images are processed and stored following the PASAL VOC dataset format.

One of the key points to underwater detection is the diversity of the underwater environments contained in the dataset, as well as the variety of feature poses and density. This dataset includes sea cucumber images under different conditions, for instance, above the water surface, in clear underwater areas, in turbid underwater areas, and in accumulation areas and sparse areas of sea cucumbers. Sample images, as illustrated in Figure 5, clearly demonstrate the problems faced in underwater detection: the turbidity of the subsea area and multi-object occlusion in the accumulation area. Diverse sample types enhance the model's robustness and provide it with strong adaptability to various special cases. In addition, a performance comparison of sea cucumber detection by YOLOv5 and DETR will be conducted based on these unique types of sample images.

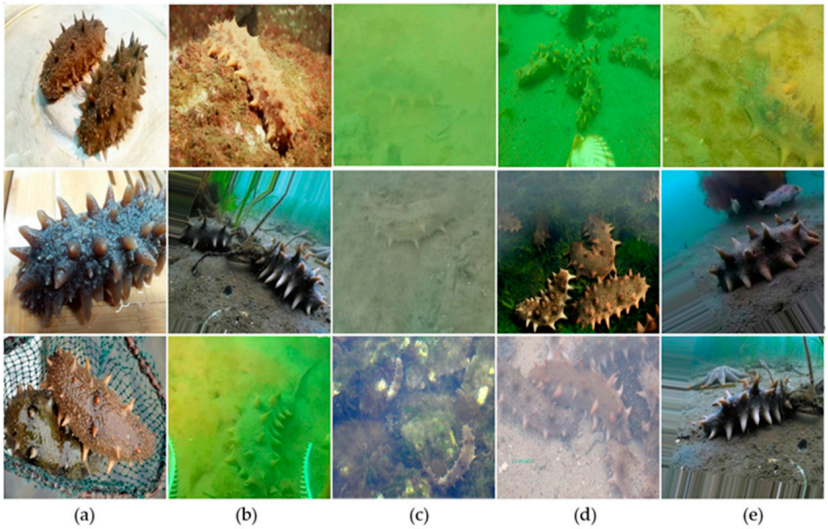


Figure 5. Some examples in the dataset: (**a**) images of sea cucumbers above the water surface; (**b**) images in clear underwater area; (**c**) images in turbid subsea region; (**d**) images with multiple targets in accumulation area of sea cucumbers; (**e**) images with few targets in sparse area of sea cucumbers.

The training and performance comparison of the YOLOv5 and DETR models for sea cucumber detection is carried out under the experimental environment configurations listed in Table 1.

Table 1. Computer configurations.

Experimental Platform	Configuration
CPU	Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz
GPU	NVIDIA GeForce RTX 2070 with Max-Q Design
RAM	DDR4 3000 MHz 16 G
Hard Drive	PCIE 3.0 NVME 512 G
OS	Windows 10
CUDA	11.6
Python	3.9

For the performance evaluation and comparation of target detection, the definitions of performance evaluation indicators need to be clarified. Precision and recall are the most frequently employed evaluation metrics. They are computed based on a confusion matrix, as described in Equations (8) and (9):

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

$$Recall = \frac{TP}{TP + FN}, \tag{9}$$

where TP, FP, and FN imply 'True Positive', 'False Positive', and 'False Negative', respectively. They are defined according to the Intersection over Union (IoU) between the predicted bounding box and the ground truth. If the IoU is bigger than the threshold, the bounding box is marked as TP, representing the number of correctly identified targets; otherwise, it is labeled as FP, implying the number of mistakenly identified objects.

Precision and recall are interactive. For the model with better performance, its precision remains high while the recall rate increases. Thus, by combining these two indicators, average precision (AP) and the mean AP of all categories (mAP) are defined; these are computed by Equations (10) and (11). In addition, the $F1$ score shown in Equation (12) is another commonly used metric for evaluating binary classification problems; it is the harmonic average of precision and recall mentioned above:

$$AP = \sum_{k=1}^{N} max_{\tilde{k} \geq k} P\left(\tilde{k}\right) \Delta R(k), \tag{10}$$

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP(c), \tag{11}$$

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{12}$$

where P, R, and C refer to precision, recall, and the number of target categories individually. The value of AP is also equal to the area under the precision–recall curve. mAP@0.5 denotes the mAP value calculated when the threshold of the IoU is 0.5. mAP@0.5:0.95 refers to the average value from mAP@0.5 to mAP@0.95, like mAP@0.5, mAP@0.55, ..., mAP@0.9, mAP@0.95. The higher the AP or mAP value is, the higher the accuracy that the model achieves.

4.2. Detection of Sea Cucumbers Based on YOLOv5

The processed sea cucumber dataset is randomly split into two sets to maintain the consistency of the data distribution: the training set and the validation set. A total of 3117 images belong to the training set, and 154 ones are in the validation set. The validation

set includes 34 images of sea cucumbers above the water surface, 61 figures in a clear underwater area, and 59 images in a turbid underwater area. The validation set can also be divided into 50 images with multiple targets in the accumulation area of sea cucumbers and 104 ones with few features.

The settings of hyperparameters of the network training are also of crucial significance. The model may eventually oscillate near the optimal solution if the learning rate is too high, and on the contrary, a small learning rate may require an increase in training epochs and result in resource waste. Regarding the selection of hyperparameters, based on the computer performance, take epoch as 98, bath_ Size as 32, other hyperparameters are consistent with the selection suggested by the YOLO's official recommendation. The specific settings of hyperparameters of the network training are displayed in Table 2.

Table 2. Hyperparameter settings of network training.

Training Epochs	Batch Size	Learning Rate (Initial)	Weight Decay
98	32	0.01	0.0005

The training consumes about 5 h, and the variation in the output loss values is demonstrated in Figure 6, including the trend of the bounding box loss, object loss, and classification loss with the training epochs, respectively. It can be found that the bounding box loss and target loss decrease rapidly in the first 10 rounds and gradually decline in the later rounds until stable. The classification loss equals 0 and remains constant during training, since there is only one type of target in the sea cucumber detection task.

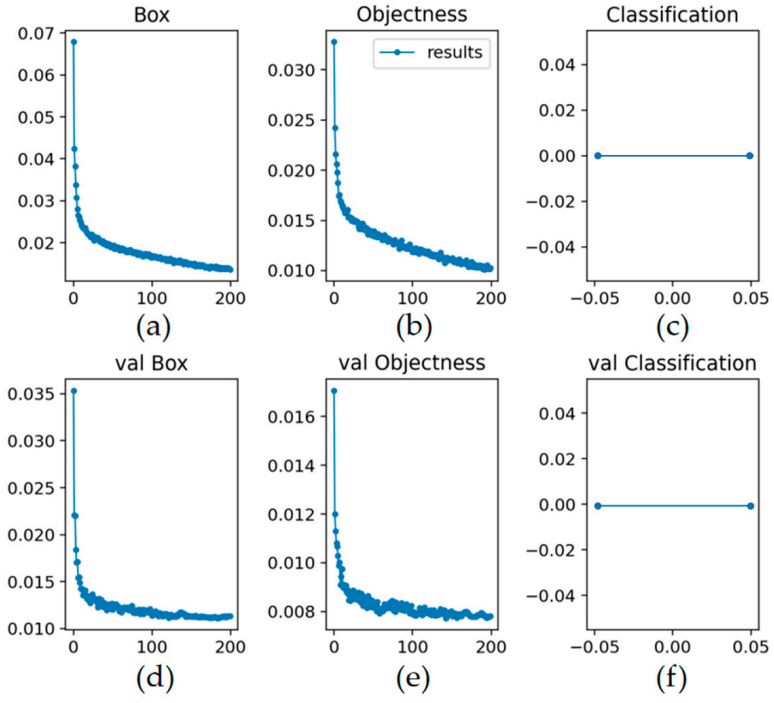


Figure 6. The variation in the loss values, including: (**a**) the bounding box loss of the training set; (**b**) the object loss of the training set; (**c**) the classification loss of the training set; (**d**) the bounding box loss of the validation set; (**e**) the object loss of the validation set; (**f**) the classification loss of the validation set.

The output values of evaluation metrics including precision, recall, and *mAP* during the training are presented in Figure 7. In this manuscript, the target precision of the detection performance is set as 95% based on the analysis of task requirements, which can ensure the overall efficiency without wasting too many computational resources. It can be

derived that the *mAP* value of the trained YOLOv5 model achieves 98.8%, exceeding the previous target precision.

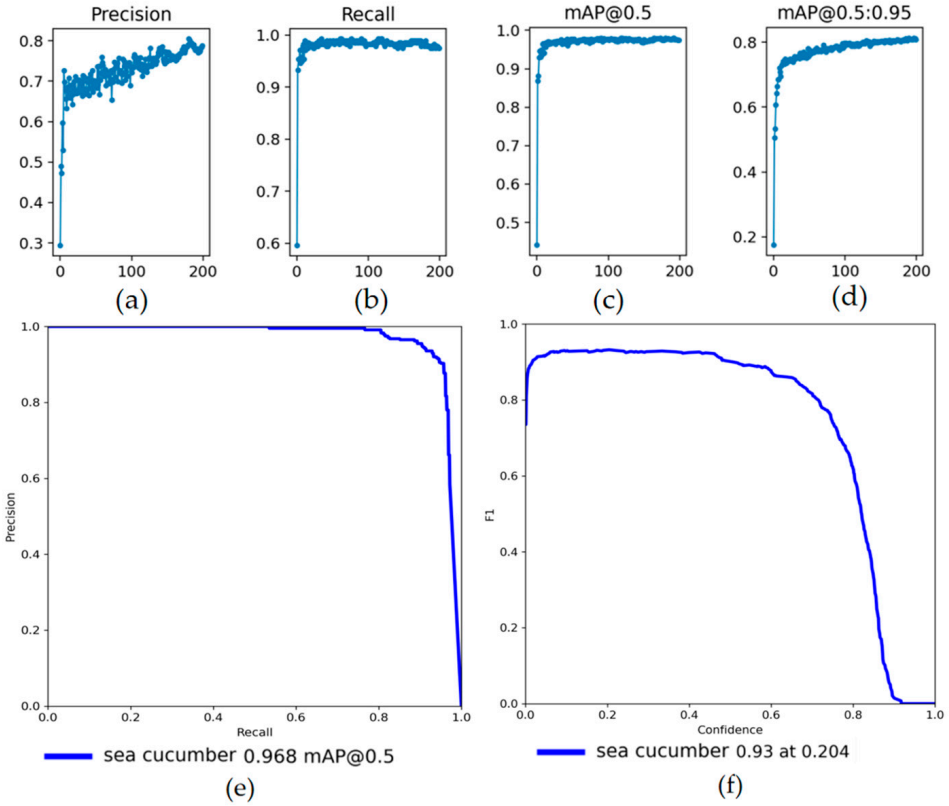


Figure 7. Output of evaluation metrics: (**a**) *precision*; (**b**) *recall*; (**c**) *MAP*@0.5; (**d**) *MAP*@0.5:0.95; (**e**) precision–recall curve; (**f**) *F*1 curve.

The precision–recall curve could then be plotted on account of the precision and recall values obtained above, which is conducive to the intuitive presentation of the current effectiveness of the model. Additionally, the *F*1 curve takes the recall rate as the horizontal coordinate and the *F*1 score as the perpendicular coordinate. The value of *F*1 is proportional to the effectiveness of the current model. When the *F*1 score reaches its maximum value, the corresponding threshold is often the optimal threshold for the model. The precision–recall curve and the *F*1 curve are also depicted in Figure 7. Some examples of the output detection performances are presented in Figure 8, and the result of running the model on the ROS of TX2 is demonstrated in Figure 9.

4.3. Detection of Sea Cucumbers Based on DETR

For the detection of sea cucumbers based on DETR, the dataset is processed and stored according to the COCO dataset format [55]. Since there are only a total of 3271 images in the dataset, and a large number of pictures are required for training, in this case, with limited data, most sample images here are the 3117 ones divided into the training set, and the remaining 154 representative ones are selected as the validation set. Additionally, in order to better compare the detection performance of YOLOv5 and DETR in diverse underwater conditions, the validation set is classified into three sorts according to clarity, so that the images are of sea cucumbers above the water surface, in a clear underwater area, and in a turbid underwater region—or it is divided into two categories based on the number of targets the images contain, with sea cucumbers within an accumulation area or a sparse zone. The variations in the output loss values during the training of DETR are demonstrated in Figure 10, including the trend of the bounding box loss and total loss.

From Figure 10, it can be seen that loss values are gradually declining, but there are significant oscillations in the mid-period of training. After training, the final *mAP*@0.5 value reaches 95.2%, and the final *mAP*@0.5:0.95 achieves 67.5%. The PR curve is drawn in Figure 10e.

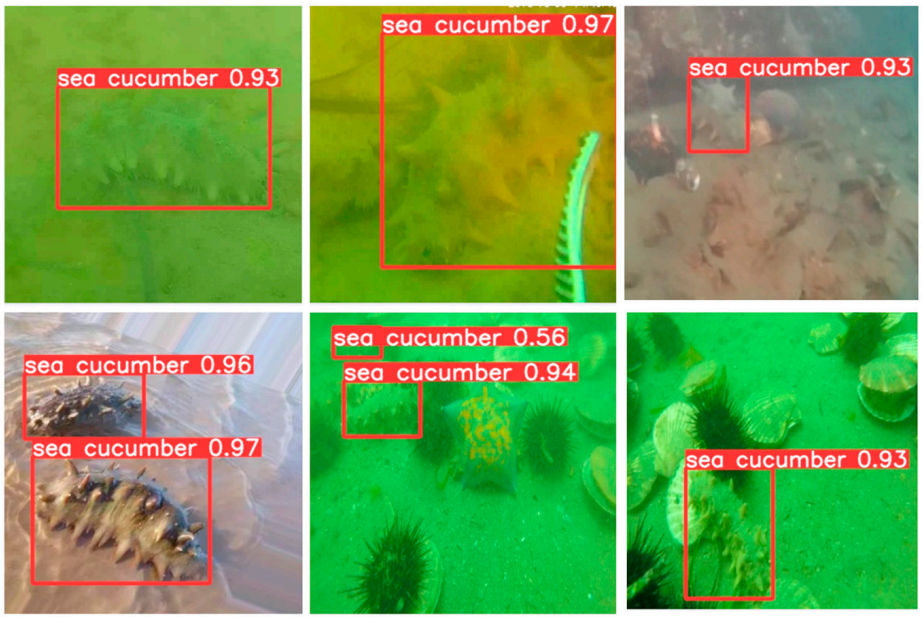


Figure 8. Examples of the detection output.

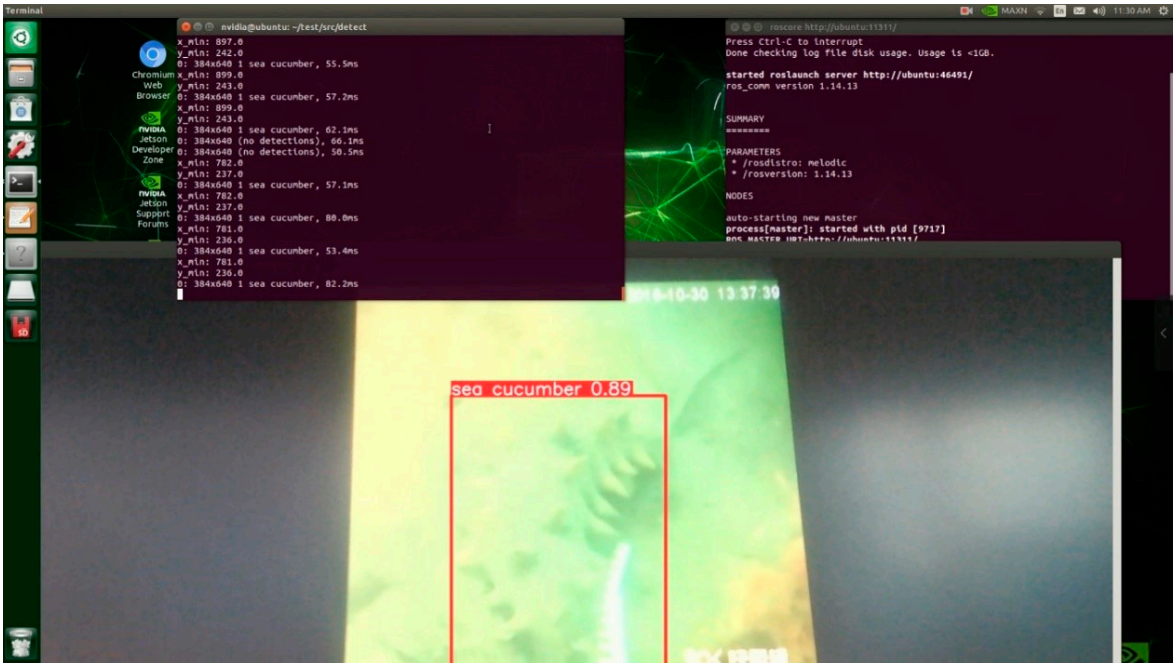


Figure 9. The result of running the model on ROS of TX2.

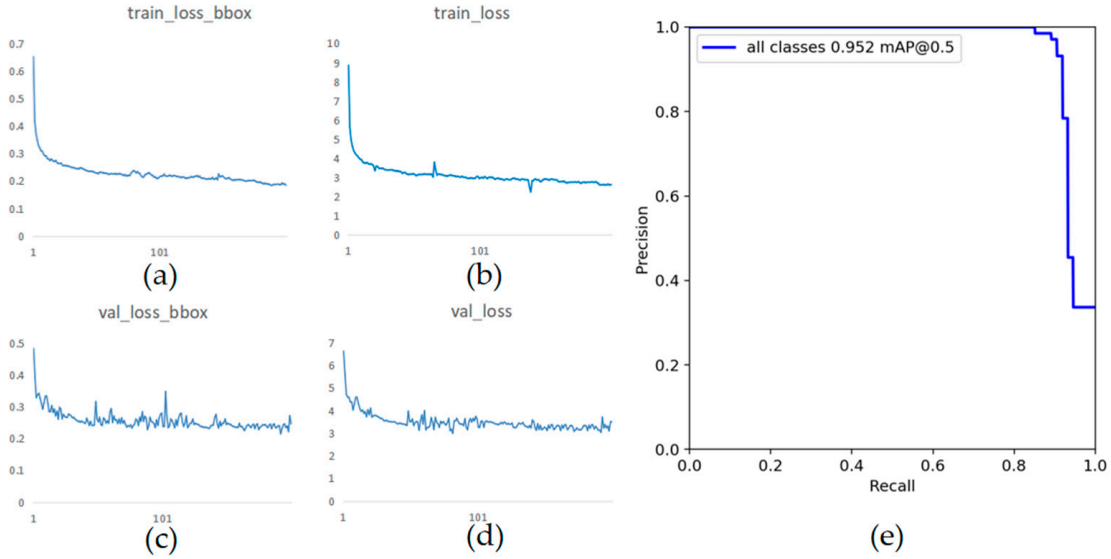


Figure 10. The variation in the loss values: (**a**) the bounding box loss of the training set; (**b**) the total loss of the training set; (**c**) the bounding box loss of the validation set; (**d**) the total loss of the validation set; (**e**) the precision–recall curve.

4.4. Performance Comparison of YOLOv5 and DETR

4.4.1. Principle of Comparison

In the experiment, the development time of the two algorithms should also be considered when comparing their performance. The YOLO algorithm has a long history of development, and it has gained significant attention from researchers. Continuous advancements and innovations have contributed to the growing maturity of the algorithm over time. The Transformer network appeared in 2017, initially being used for natural language processing rather than computer vision, and the DETR based on Transformer for object detection was proposed in 2020, with relatively immature development compared with YOLO. Until now, there is still substantial room for further development. Therefore, in the end, the future trends and prospects of the two will be predicted based on their development time and speed.

The experimental dataset covers images taken both underwater and above the water surface. Therefore, to compare the two algorithms more comprehensively, the selected validation set will be briefly classified. The underwater images in the dataset are either clearer or blurrier, whereas the images above the water surface are generally clearer. Therefore, the validation set can be divided into three categories: underwater clear, underwater blurry, and above water surface. The recognition performance of YOLOv5 and DETR on these three types of datasets is discussed for a more comprehensive comparison. Meanwhile, to test the performance of the two models in recognizing dense targets, the samples will be classified according to the number of targets contained in them. Here, images containing more than three targets are considered and categorized into two classes: images with multiple targets and those with few ones.

4.4.2. Comparison of Simulation Results

During the training of the network model, all the training sets are put into training without subdivision, and all the validation sets are used to test the performance of the trained model. In the former section, YOLOv5 and DETR were successfully trained using the allocated dataset, and preliminary results have been obtained. In the performance comparison testing stage, only the trained network models need to be called: the original validation set is simply replaced with multiple previously divided validation sets. Next, the performance evaluation section is run in the training code to obtain the relevant performance of the trained network model for different types of samples.

After testing various validation sets through experiments, the performance of the two models for three kinds of validation sets, namely, underwater blurry, underwater clear, and above the water surface, is obtained. The testing performance of the two models varies. In Tables 3 and 4, 'total' represents the test performance on all samples.

Table 3. Test performance of YOLOv5 on various validation sets (above water surface or in underwater area).

	Precision	*Recall*	*mAP*@0.5	*mAP*@0.5:0.95
Above water surface	88.9%	97.3%	98.4%	86.7%
Underwater clear	73.4%	100%	97.5%	79.9%
Underwater blurry	76%	96.1%	96.4%	76.6%
Total	78.1%	97.5%	97.8%	81.0%

Table 4. Test performance of DETR on various validation sets (above water surface or in underwater area).

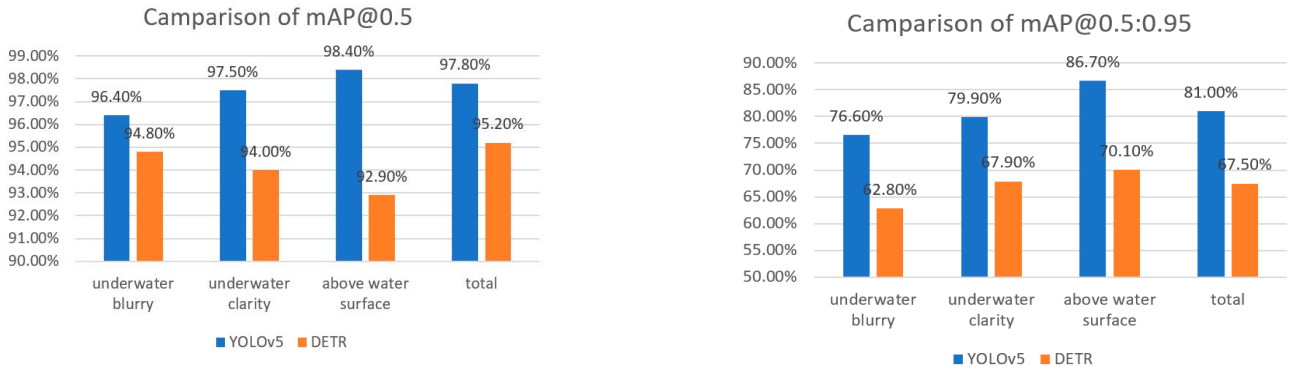
	Precision	*Recall*	*mAP*@0.5	*mAP*@0.5:0.95
Above water surface	84.6%	91.7%	92.9%	70.1%
Underwater clear	74.1%	95.4%	94.0%	67.9%
Underwater blurry	72.4%	94.6%	94.8%	62.8%
Total	76.5%	95.4%	95.2%	67.5%

After comparison, it can be found that the YOLOv5 algorithm surpasses the DETR algorithm in all aspects, and the overall detection performance of YOLOv5 is better. As for the value of *mAP*@0.5, YOLOv5 is overall 2% to 3% higher than the DETR algorithm.

It can be seen that the YOLOv5 model has a decline in performance compared with all samples when facing underwater images, and the performance with clear samples is also higher than that with blurry ones. It has the best detection performance above the water surface, whether it is *mAP*@0.5 or *mAP*@0.5:0.95, which complies with the above rules. The YOLOv5 has only a 2% difference in *mAP*@0.5 between blurry samples and above-water surface ones, indicating that YOLOv5 has high robustness. The detection performance with underwater blurry samples is still relatively good.

The DETR model exhibits opposite trends in *mAP*@0.5 and *mAP*@0.5:0.95 values for three different samples, with *mAP*@0.5:0.95 being the highest for the samples above the water level and the lowest for the underwater blurry samples, *mAP*@0.5 being the highest for the underwater blurry samples and the lowest for the samples above the water level. Observing the samples, it is not difficult to find that many of the samples above the water level contain multiple targets. From the internal structure analysis of the DETR model, it sets up 100 decoders and predicts 100 frames, which is relatively small. This may be the reason for the lower *mAP*@0.5 value of samples above the water surface. Due to the attention mechanism of the DETR model, the bounding box has a higher IoU compared to the ground truth. Therefore, for these three kinds of regions—underwater blurry, underwater clear, and above the water surface—the mAP@0.5:0.95 values of the three samples still maintain an overall upward trend. A comparison of the performance of the two algorithms is depicted in Figure 11.

The recognition effects of the two algorithms for the three types of samples, which are sea cucumbers above the water surface, in a clear underwater area, and in a turbid underwater region, are demonstrated in Figure 12. The images in the upper row of Figure 12 display the YOLOv5 detection effect, and the ones below show the DETR detection effect.



(**a**) (**b**)

Figure 11. Performance comparison on samples with different clarities: (**a**) Comparison under the value of *mAP*@0.5; (**b**) comparison under the value of mAP@0.5:0.95.

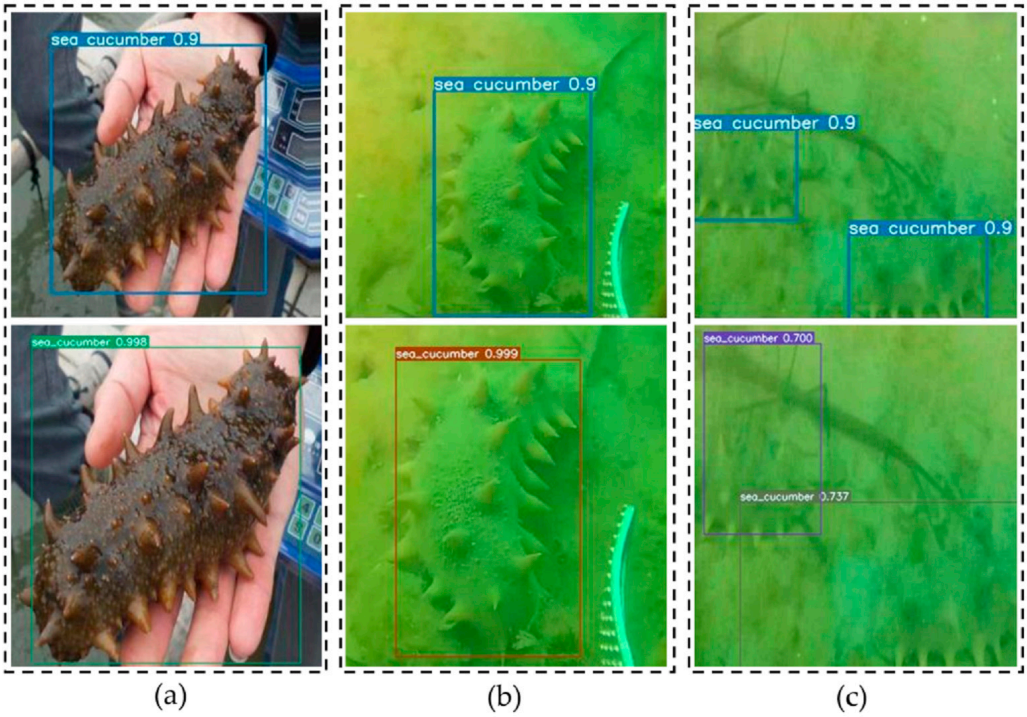


Figure 12. Detection performances of the two algorithms: (**a**) detection effect with samples above the water surface; (**b**) detection effect with samples in clear underwater area; (**c**) detection effect with samples in turbid underwater zone.

After comparison, it can be seen that even for underwater blurry images, both the DETR and YOLOv5 algorithm models achieve good detection performances. This indicates that even in complex environments with weakened underwater light, both algorithm models can also detect targets and be applied to underwater image-processing applications.

Next, the detection performance of the two models is tested for environments in which there are dense targets. For cases with multiple targets, the pre-divided validation set will be replaced with the original validation set to obtain the results. The recognition performances of the two models are demonstrated in Tables 5 and 6.

Table 5. Test performance of YOLOv5 on various validation sets with various density of targets.

	Precision	*Recall*	*mAP*@0.5	*mAP*@0.5:0.95
More targets	69.6%	97.5%	95.6%	74.7%
Few targets	93.2%	98.3%	99.3%	87.9%
Total	78.1%	97.5%	97.8%	81.0%

Table 6. Test performance of DETR on various validation sets with various density of targets.

	Precision	*Recall*	*mAP*@0.5	*mAP*@0.5:0.95
More targets	60.1%	91.2%	90.9%	57.3%
Few targets	87.2%	96.8%	97.2%	76.3%
Total	76.5%	95.4%	95.2%	67.5%

From the above Tables 5 and 6, it can be found that YOLOv5 has better detection performance for multiple targets. In terms of detection performance for samples containing multiple targets, the *mAP*@0.5 value of YOLOv5 is 5% greater than that of the DETR algorithm, since the YOLOv5 has three outputs, which are used to predict large, medium, and small targets, and the grid division is also quite precise. The extracted features have a deeper level and are more adaptable to densely distributed targets. However, the DETR predicts 100 bounding boxes, which may not be sufficient in terms of distribution and quantity, and the detection performance for some features with occlusion relationships is not satisfactory.

Then, by comparing the processing ability of one algorithm for samples with various targets, the YOLOv5 shows a difference of only 3.7% in the *mAP*@0.5 value and 13.2% in the *mAP*@0.5:0.95 value of multi-target samples compared with those with few. This indicates that YOLOv5 has relatively strong processing ability for samples with dense features, whereas the DETR has a difference of 6.3% in the *mAP*@0.5 value and 19% in the *mAP*@0.5:0.95 value of the two samples, indicating that improvements are required in its processing ability for samples with dense targets. The comparison of the performance of the two algorithms is clarified in Figure 13.

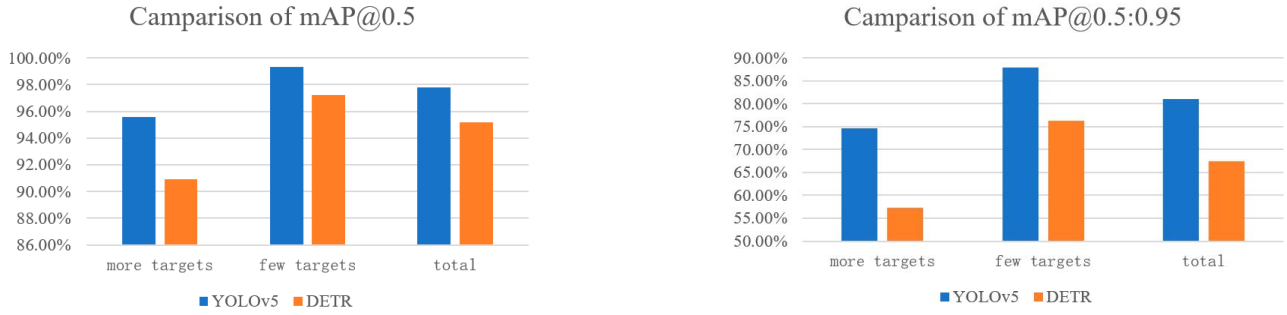


Figure 13. Performance comparison on samples with different number of targets: (**a**) comparison under the value of mAP@0.5; (**b**) comparison under the value of mAP@0.5:0.95.

Since the processing ability of the algorithm model for dense targets is given much attention, the specific recognition effects of the two algorithm models for samples containing dense targets are demonstrated in Figure 14. The left side displays the processing effects of YOLOv5, and the right shows the performance of DETR.

It can be seen that although the DETR algorithm can detect most of the targets, there may still be some omissions and deviations. Nonetheless, in underwater environments, its performance is seriously lower than that of YOLOv5. To sum up, the YOLOv5 model can locate objects more accurately, and there are fewer missing features. The overall performance of YOLOv5 is better, and even in subsea regions, it has a high recall rate.

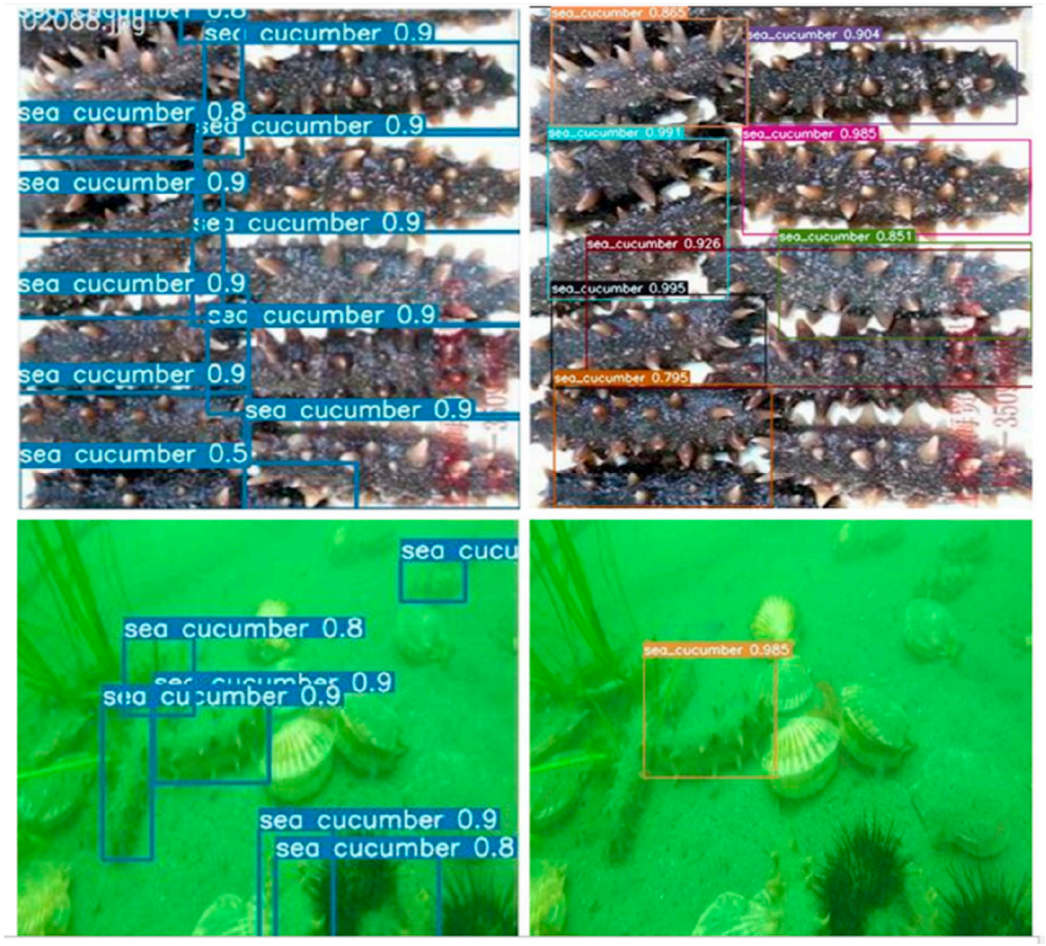


Figure 14. Sample detection performance with multiple targets.

5. Conclusions and Future Work

Unlike conventional terrestrial imagery, underwater imagery is influenced by a diversity of factors, including water turbidity, floating objects in the water, light refraction, absorption, etc., which may result in image distortion or reduced visibility of the target. In underwater applications, smaller and lighter devices are required, and large models may not be applicable. Therefore, it becomes a challenge to find a balance between model size and performance. In summary, the development of underwater target detection approaches is summarized in this present work, including conventional object detection approaches and methods based upon deep learning. Then, in view of the analysis of state-of-the-art underwater sea cucumber detection approaches and aiming to provide a reference for practical underwater identification, adjacent and overlapping sea cucumber detection based on YOLOv5 and DETR, which are examples of one-stage and anchor-free deep leaning methods, respectively, is investigated and compared thoroughly. Detection experiments with these two approaches are deployed on the measured dataset; these demonstrate the outstanding performance of YOLOv5 in accuracy and speed. The results prove that YOLOv5 outperforms DETR in terms of low computing consumption and high precision, particularly in detecting for small and dense features. However, it is worth noting that DETR has shown rapid development of the model and holds promising prospects in underwater object detection applications due to its relatively simple architecture and innovative attention mechanism.

The similarity of all samples is high, so there is no significant difference between the validation set and the training set, which will result in a higher *mAP* value in the validation set. As a next stage, so as to further deepen the current research, the principal future work is as follows:

1. Improving detection accuracy and processing time and optimizing the architecture and hyperparameters of both the YOLOv5 and DETR models;
2. Exploring and evaluating the performances of YOLOv5 and DETR for detecting other marine species;
3. Developing new data augmentation techniques to expand the diversity and quantity of training data for underwater target detection;
4. Developing real-time object detection systems using YOLOv5 and DETR and evaluating their performance in practical scenarios.
5. Introducing appropriate image preprocessing techniques during training and detection to further enhance the performance of the model.

Author Contributions: X.Y. and N.L. proposed the idea. X.Y., S.F. and Q.M. designed the experiments. Z.W., M.G., P.T. and C.Y. analyzed the experiments. X.Y. and S.F. wrote the manuscript. Y.W. and J.-F.M.O. edited and proofread the article. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to the presented results has been undertaken within the National Natural Science Foundation of China (Youth Project) under Grant No. 62101159, the Chinese Shandong Provincial Natural Science Foundation (General Program) under Grant No. ZR2021MF055, and the Chinese Shandong Provincial Key Research and Development Plan under Grant No. 2021CXGC010702, 2022CXGC020410, 2022CXGC020412 and 2020CXGC10103. The research was also supported by the SWARMs European project under Grant Agreement No. 662107-SWARMs-ECSEL-2014-1, partially sponsored by the ECSEL JU and the Spanish Ministry of Economy and Competitiveness (Ref: PCIN-2014-022-C02-02).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank the editors and reviewers for their valuable suggestions.

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

Abbreviations

The following abbreviations are used in this manuscript:

ROV	Remotely Operated Vehicle
AUV	Autonomous Underwater Vehicle
SIFT	Scale-Invariant Feature Transform
HOG	Histograms of Oriented Gradients
SVM	Support Vector Machine
SSD	Single Shot MultiBox Detector
YOLO	You Only Look Once
DETR	DEtection TRansformer
DPM	Deformable Part Model
RCNN	Region Convolutional Neural Network
RFCN	Region-based Fully Convolutional Network
UDCP	Under Dark Channel Prior
UD-ETR	Under Dark Energy Transmission Restoration
SWIPENet	Sample-Weighted Hyper Network
DSSD	Deconvolutional Single Shot Detector
LWAP	Local Wavelet Acoustic Pattern
MLP	Multi-layer perceptron
PAN	Pyramid Attention Network
FPN	Feature Pyramid Network
mAP	Mean average precision
AON	Adversarial Occlusion Network
S-FPN	Shortcut Feature Pyramid Network

SRCNN	Super-Resolution Convolutional Neural Network
AFFM	Attentional Feature Fusion Module
APAN	Attention Path Aggregation Network
USBL	Ultrashort Base Line
AP	Average precision
NMS	Non-Maximum Suppression
ReLU	Rectified Linear Unit
CNN	Convolutional neural network

References

1. Sahoo, A.; Dwivedy, S.K.; Robi, P. Advancements in the field of autonomous underwater vehicle. *Ocean Eng.* **2019**, *181*, 145–160. [CrossRef]
2. Xu, F.Q.; Ding, X.; Peng, J.; Yuan, G.; Wang, Y.; Zhang, J.; Fu, X. Real-time detecting method of marine small object with underwater robot vision. In Proceedings of the 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, Japan, 28–31 May 2018; pp. 1–4.
3. Lei, F.; Tang, F.; Li, S. Underwater target detection algorithm based on improved YOLOv5. *J. Mar. Sci. Eng.* **2022**, *10*, 310. [CrossRef]
4. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
5. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
6. Platt, J. Sequential minimal optimization: A fast algorithm for training support vector machines. *Adv. Kernel Methods-Support Vector Learn.* **1998**, *208*, 1–21.
7. Villon, S.; Chaumont, M.; Subsol, G.; Villeger, S.; Claverie, T.; Mouillot, D. Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and HOG plus SVM methods. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS), Lecce, Italy, 24–27 October 2016; pp. 160–171.
8. Xu, S.B.; Zhang, M.H.; Song, W.; Mei, H.B.; He, Q.; Liotta, A. A systematic review and analysis of deep learning-based underwater object detection. *Neurocomputing* **2023**, *527*, 204–232.
9. Dhiraj, N.; Jongwon, S. A review on deep learning-based approaches for automatic sonar target recognition. *Electronics* **2020**, *9*, 1972.
10. Han, P.L. Research on Polarization Imaging Exploration Technology of Underwater target. Ph.D. Thesis, Xidian University, Xi'an, China, June 2018.
11. Amer, K.O.; Elbouz, M.; Alfalou, A.; Brosseau, C.; Hajjami, J. Enhancing underwater optical imaging by using a low-pass polarization filter. *Environ. Sci.* **2019**, *27*, 621–643. [CrossRef]
12. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef]
13. Er, M.J.; Chen, J.; Zhang, Y.; Gao, W. Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review. *Sensors* **2023**, *23*, 1990.
14. Shen, Z.; Liu, Z.; Li, J.; Jiang, Y.G.; Chen, Y.; Xue, X. Dsod: Learning deeply supervised object detectors from scratch. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1919–1927.
15. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1–8.
16. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
17. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
18. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
19. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 386–397. [CrossRef]
20. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot multi-box Detector. In *Computer Vision-ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 21–37.
22. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
23. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
24. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.

25. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
26. Glenn, J. YOLOv5 Is Here: State-of-the-Art Object Detection at 140 FPS. Roboflow, 2020. Available online: https://blog.roboflow.com/yolov5-is-here/ (accessed on 22 September 2023).
27. Thomas, R.; Thampi, L.; Kamal, S.; Balakrishnan, A.A.; Mithun Haridas, T.P.; Supriya, M.H. Dehazing underwater images using encoder decoder based generic model-agnostic convolutional neural network. In Proceedings of the 2021 International Symposium on Ocean Technology (SYMPOL), Kochi, India, 9–11 December 2021; pp. 1–4.
28. Martin, M.; Sharma, S.; Mishra, N.; Pandey, G. UD-ETR Based Restoration & CNN Approach for Underwater Object Detection from Multimedia Data. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020.
29. Chen, L.; Liu, Z.; Tong, L.; Jiang, Z.; Wang, S.; Dong, J.; Zhou, H. Underwater object detection using invert multi-class adaboost with deep learning. In Proceedings of the 2020 International Joint Conference on Neural Networks, Glasgow, UK, 19–24 July 2020; pp. 1–8.
30. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
31. Berman, D.; Levy, D.; Avidan, S.; Treibitz, T. Underwater single image color restoration using haze-lines and a new quantitative dataset. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 2822–2837. [CrossRef] [PubMed]
32. Qiao, W.; Khishe, M.; Ravakhah, S. Underwater targets classification using local wavelet acoustic pattern and multi-layer perceptron neural network optimized by modified whale optimization algorithm. *Ocean Eng.* **2021**, *219*, 108415. [CrossRef]
33. Yeh, C.H.; Lin, C.H.; Kang, L.W.; Huang, C.H.; Lin, M.H.; Chang, C.Y.; Wang, C.C. Lightweight deep neural network for joint learning of underwater object detection and color conversion. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6129–6143. [CrossRef]
34. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
35. Hu, X.; Liu, Y.; Zhao, Z.; Liu, J.; Yang, X.; Sun, C.; Chen, S.; Li, B.; Zhou, C. Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network. *Comput. Electron. Agric.* **2021**, *185*, 106135. [CrossRef]
36. Zeng, L.; Sun, B.; Zhu, D. Underwater target detection based on Faster R-CNN and adversarial occlusion network. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104190. [CrossRef]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Peng, F.; Miao, Z.; Li, F.; Li, Z. S-FPN: A shortcut feature pyramid network for sea cucumber detection in underwater images. *Expert Syst. Appl.* **2021**, *182*, 115306. [CrossRef]
39. Lin, W.H.; Zhong, J.X.; Liu, S.; Li, T.; Li, G. Roimix: Proposal-fusion among multiple images for underwater object detection. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2588–2592.
40. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
41. Li, M.; Mathai, A.; Lau, S.L.; Yam, J.W.; Xu, X.; Wang, X. Underwater object detection and reconstruction based on active single-pixel imaging and super-resolution convolutional neural network. *Sensors* **2021**, *21*, 313. [CrossRef] [PubMed]
42. Strong, D.; Chan, T. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Probl.* **2003**, *19*, S165–S187. [CrossRef]
43. Park, J.H.; Kang, C. A study on enhancement of fish recognition using cumulative mean of YOLO network in underwater video images. *J. Mar. Sci. Eng.* **2020**, *8*, 952. [CrossRef]
44. Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight underwater object detection based on YOLO v4 and multi-scale attentional feature fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]
45. Gao, S.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P.H. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef]
46. Tan, M.; Le, Q.V. Mixconv: Mixed depthwise convolutional kernels. *arXiv* **2019**, arXiv:1907.09595.
47. Li, L.; Wang, Z.; Zhang, T. Gbh-yolov5: Ghost convolution with bottleneckcsp and tiny target prediction head incorporating yolov5 for PV panel defect detection. *Electronics* **2023**, *12*, 561. [CrossRef]
48. Li, J.; Liu, C.; Lu, X.; Wu, B. CME-YOLOv5: An efficient object detection network for densely spaced fish and small targets. *Water* **2022**, *14*, 2412. [CrossRef]
49. Wen, G.; Li, S.; Liu, F.; Luo, X.; Er, M.J.; Mahmud, M.; Wu, T. YOLOv5s-CA: A modified YOLOv5s network with coordinate attention for underwater target detection. *Sensors* **2023**, *23*, 3367. [CrossRef]
50. Yu, H.; Li, X.; Feng, Y.; Han, S. Multiple attentional path aggregation network for marine object detection. *Appl. Intell.* **2022**, *53*, 2434–2451. [CrossRef]
51. Liu, L.X. Research on Target Detection and Tracking Technology of Imaging Sonar. Ph.D. Thesis, Harbin Engineering University, Harbin, China, 20 November 2015.
52. Mandić, F.; Rendulić, I.; Mišković, N.; Nađ, Đ. Underwater object tracking using sonar and USBL measurements. *J. Sens.* **2016**, *2016*, 8070286. [CrossRef]

53. Wang, C.-Y.; Bochikovskiy, A.; Mark Liao, H.-Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time-object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
54. Shandong Future Robot Co., Ltd. Available online: http://www.vvlai.com/ (accessed on 22 September 2023).
55. COCO Dataset. Available online: https://cocodataset.org/ (accessed on 22 September 2023).