applied sciences

MDPI

Article

Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network

Minghua Zhang, Zhihua Wang, Wei Song, Danfeng Zhao * and Huijuan Zhao

College of Information Technology, Shanghai Ocean University, Shanghai 201306, China; mhzhang@shou.edu.cn (M.Z.); m210911556@st.shou.edu.cn (Z.W.); wsong@shou.edu.cn (W.S.); hjzhao@shou.edu.cn (H.Z.)
* Correspondence: dfzhao@shou.edu.cn

Abstract: Underwater object detection plays a significant role in marine ecosystem research and marine species conservation. The improvement of related technologies holds practical significance. Although existing object-detection algorithms have achieved an excellent performance on land, they are not satisfactory in underwater scenarios due to two limitations: the underwater objects are often small, densely distributed, and prone to occlusion characteristics, and underwater embedded devices have limited storage and computational capabilities. In this paper, we propose a high-precision, lightweight underwater detector specifically optimizing for underwater scenarios based on the You Only Look Once Version 8 (YOLOv8) model. Firstly, we replace the Darknet-53 backbone of YOLOv8s with FasterNet-T0, reducing model parameters by 22.52%, FLOPS by 23.59%, and model size by 22.73%, achieving model lightweighting. Secondly, we add a Prediction Head for Small Objects, increase the number of channels for high-resolution feature map detection heads, and decrease the number of channels for low-resolution feature map detection heads. This results in a 1.2% improvement in small-object detection accuracy, while the remaining model parameters and memory consumption are nearly unchanged. Thirdly, we use Deformable ConvNets and Coordinate Attention in the neck part to enhance the accuracy in the detection of irregularly shaped and densely occluded small targets. This is achieved by learning convolution offsets from feature maps and emphasizing the regions of interest (RoIs). Our method achieves 52.12% AP on the underwater dataset UTDAC2020, with only 8.5 M parameters, 25.5 B FLOPS, and 17 MB model size. It surpasses the performance of large model YOLOv8l, at 51.69% AP, with 43.6 M parameters, 164.8 B FLOPS, and 84 MB model size. Furthermore, by increasing the input image resolution to 1280 × 1280 pixels, our model achieves 53.18% AP, making it the state-of-the-art (SOTA) model for the UTDAC2020 underwater dataset. Additionally, we achieve 84.4% mAP on the Pascal VOC dataset, with a substantial reduction in model parameters compared to previous, well-established detectors. The experimental results demonstrate that our proposed lightweight method retains effectiveness on underwater datasets and can be generalized to common datasets.

Keywords: YOLO; lightweight detector; small-object detection; deep learning

Citation: Zhang, M.; Wang, Z.; Song, W.; Zhao, D.; Zhao, H. Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network. *Appl. Sci.* **2024**, *14*, 1095. https://doi.org/10.3390/app14031095

Academic Editor: Sungho Kim

Received: 12 December 2023
Revised: 17 January 2024
Accepted: 23 January 2024
Published: 27 January 2024

1. Introduction

More than 70% of the Earth is covered by water, and our oceans play a vital role in the survival of humans all around the globe. Compared to the level of development on land, the oceans are still veiled in a layer of mystery, holding a vast amount of untapped resources. The marine environment has also been continuously under threat in recent years, making research on the marine environment meaningful. An increasing number of researchers are focusing on the development of underwater technologies, such as underwater acoustics, underwater magnetism [1], underwater vehicle systems, underwater sensing components, and underwater unexploded ordnance detection [2], among others. With the advancement of computer vision, exploring the oceans using computer vision technology has become

a new avenue. Underwater optical images have a relatively high resolution and contain rich information, making them particularly advantageous for short-range underwater target detection tasks. Fueled by the winds of computer vision, object detection has gradually become one of the hottest technologies in ocean exploration, making a significant contribution to the development of marine resources.

The field of Generic Object Detection [3–5] has evolved for more than two decades, progressing from traditional methods to the deep learning techniques used at present. Over this time, there has been a notable increase in accuracy, coupled with an improved processing speed. However, Generic Object Detection in the underwater domain still faces significant challenges that need to be addressed. Firstly, underwater environments suffer from issues such as uneven lighting, low contrast, blurriness, and bright spots, which impact the quality of features extracted from underwater images. Secondly, underwater organisms are typically small and densely distributed, often leading to overlapping and occlusion. Detecting small objects is a challenging aspect of generic object detection. Thirdly, underwater embedded devices often have limited computational and storage capabilities, making it difficult to deploy large models on these devices. Therefore, finding solutions to allow for the accurate and rapid detection of objects in complex underwater environments is an urgent problem that needs to be addressed.

Existing underwater object detection methods have seen numerous improvements. Most underwater objects are small and densely packed, making it challenging for general detectors to detect these small and blurry objects. Song et al. [6] introduced a two-stage underwater detector with three key components, addressing uncertainty modeling and hard example mining. The RetinaRPN network, the first component, utilizes objectness and IoU prediction to generate high-quality proposals. The second component, a Probabilistic Inference Pipeline, combines prior uncertainty from the first stage with the second stage's classification score to obtain the final detection score. The third component employs "Boosting Reweighting", a novel hard example mining method that amplifies the classification loss of challenging examples while reducing the weight of easy examples. This approach facilitates the acquisition of a robust detection head in the second stage. During the inference stage, the integration of R-CNN rectifies errors from the first stage, resulting in an overall performance improvement. Their work yielded positive results in the domain of underwater images, but the composed backbones were characterized by a large number of parameters, making them unsuitable for real-time applications. Zhang et al. [7] proposed a lightweight underwater detector based on YOLOv4 [8], integrating MobileNetv2 [9] and depth-wise separable convolution [10] to reduce the network parameters. This work achieved real-time underwater target detection and performed well compared to traditional detectors. However, there is still room for performance improvements compared to state-of-the-art detectors.

To address these issues, we introduce a lightweight detector for underwater images based on YOLOv8, optimizing the detections on small and desensed underwater objects. Our contributions are summarized as follows:

(1) To achieve model lightweighting, we use FasterNet-T0 [11] to replace the backbone of YOLOv8, slightly reducing model accuracy in exchange for a faster training speed and fewer model parameters.

(2) In order to enhance the accuracy of small-object detection, we first integrate a prediction head for small objects into YOLOv8 because underwater images often contain many small objects. The prediction head we add is generated from high-resolution feature maps, making it more sensitive to small objects. We also perform specific optimizations for the number of channels in different resolution feature maps. Second, we enhance the performance of detecting small objects and handling occlusions in dense underwater images by utilizing Deformable ConvNets v2 [12] and incorporating Coordinate Attention [13] to embed positional information into channel attention, which incurs almost no computational overhead but helps the network find regions of interest in images.

(3) With our lightweight model, we achieve 52.12% AP on the UTDAC2020 underwater dataset [6], surpassing the larger YOLOv8l model (AP 51.69%). When increasing the input resolution to 1280, the AP reach 53.18%. Additionally, we obtain outstanding results of mAP 84.4% on the Pascal VOC dataset, surpassing previous well-established detectors. These results demonstrate the effectiveness of our method in underwater environments and its generalization to common datasets.

2. Related Work

2.1. YOLOv8 Network

YOLO [3], introduced by Joseph Redmon et al. in CVPR 2016, revolutionized object detection with a real-time end-to-end approach. The methodology involves a two-step process, where the first step detects potential object regions, and the second step utilizes a classifier for these proposals. Unlike Fast R-CNN [14], YOLO adopts a more straightforward output approach, incorporating both probabilities for classification and box coordinates for regression in a single output, enhancing its efficiency and simplicity in comparison to the two separate outputs approach of Fast R-CNN.

YOLOv8, a cutting-edge state-of-the-art (SOTA) model, builds upon the success of its predecessors by introducing new features and improvements. The primary objective is to enhance overall performance and flexibility. Similar to YOLOv5, the architecture consists of a backbone, head, and neck. The backbone and neck part draw inspiration from the YOLOv7 [15] ELAN [16] design. They substituted the YOLOv5 C3 structure with a C2f structure, which facilitates a more extensive gradient flow. Additionally, they finetuned the number of channels for various scale models, leading to a substantial improvement in the overall performance of the model. The head part contains significant changes compared to YOLOv5. It adopts the currently mainstream decoupled head, separating the classification and the regression tasks, and replaces anchor-based with anchor-free. In the loss part, the task-aligned assigner [17] label-matching strategy is employed, and distribution focal loss [18] is introduced. YOLOv8 employs mosaic augmentation during training but disables it for the final 10 epochs to mitigate potential detrimental effects throughout the entire training process.

YOLOv8 offers five scaled versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large). Notably, YOLOv8x, evaluated with the MS COCO dataset test-dev 2017, achieved an impressive AP of 53.9% with an image size containing a longer side of 640 pixels, surpassing YOLOv5's performance of 50.7% on the same input size. Additionally, YOLOv8x demonstrated a remarkable speed of 280 FPS on an NVIDIA A100 and TensorRT during evaluation.

2.2. Lightweight Networks

Following AlexNet's [19] triumph in the 2012 ImageNet [20] Challenge, the rapid advancement of graphics processing units (GPUs) has propelled deep neural networks (DNNs) to show significant potential in various AI domains. Meanwhile, resource-constrained devices such as mobile phones and edge devices have become increasingly common. These devices often have limited computational power, as well as limited energy resources and a limited storage capacity, which pose challenges when deploying DNNs. Therefore, reducing model parameters and computational complexity while maintaining model accuracy has become an urgent task.

At present, lightweight research commonly utilizes two main approaches: network architecture design and model compression. The former focuses on exploring and designing efficient network structures that reduce model parameters and the number of floating-point operations (FLOPs) while maintaining a good performance. Some notable examples include SqueezeNet [21], ShuffleNet (V1, V2) [22,23], MobileNet (V1, V2, V3) [9,24,25], EfficientNet [26], GhostNet [27], MobileViT [28], and the FasterNet [11] used in this paper. These networks have significantly contributed to the development of deep learning on mobile devices. The latter involves various techniques to reduce the size and computational

complexity of deep neural networks. Recent popular methods include pruning, quantization, knowledge distillation, and a neural architecture search. Achieving model compression while meeting specific performance constraints (e.g., accuracy and latency) can be challenging. Researchers are now looking into joint research on hardware, software, and algorithm optimization as the next trend in model compression. This holistic approach considers not only accuracy but also energy efficiency and hardware costs during the design phase, leading to more efficient and effective optimizations for real-world applications.

In the field of object detection, a common approach to model lightweighting is to use a lightweight backbone and replace the convolutional layers. Chen et al. [29] used the ImageNet pre-trained FasterNet as a backbone and integrated it with the popular Mask R-CNN detector [30], resulting in a faster and better backbone compared to others. Depth-wise Separable Convolutions [10], Pconv [11], and similar convolutional techniques have also shown significant effectiveness in reducing model parameters. Currently, lightweight networks have found widespread applications in embedded systems, such as surface scratch detection [31]. In this paper, we approach the design of a lightweight underwater network by applying FasterNet as the backbone for YOLOv8. Although this leads to a slight decrease in accuracy, subsequent targeted optimizations for underwater scenarios have improved the model accuracy.

2.3. Small-Object Detection

Recent advancements in generic object detection have been achieved through the application of deep learning techniques. However, detecting small objects in images remains a complex challenge due to their limited size, subtle appearance, and intricate geometry cues. Compounding this difficulty is the absence of extensive datasets dedicated to small objects. Improving the ability to detect small objects holds great practical significance in real-world applications, such as underwater robotics, autonomous driving for vehicles, and drone-based detection, among others.

Current trends in small-object detection encompass key techniques such as multiscale representation, contextual information, super-resolution, and region proposal. Multiscale representation combines specific location details extracted from low-level feature maps with abundant semantic information derived from high-level feature maps. For instance, the Feature Pyramid Network (FPN) [32] algorithm simultaneously utilizes low-level features with a high resolution and high-level features with high levels of sembantic information. By fusing features from different layers, it achieves effective predictions. Experiments suggest that simply increasing the depth of the network may not be an effective solution to the challenge of detecting small objects. Small-object detection benefits from the fusion of high-resolution and semantically rich feature maps. Leveraging the relationship between an object and its surrounding environment is a novel approach to improving small-object detection accuracy. Extracting additional contextual information as a supplement to the original region of interest (ROI) features is crucial since the ROI features extracted from small objects are often limited. Attention mechanisms are one example of a technique inspired by cognitive attention in artificial neural networks. These mechanisms enhance the importance of certain parts of the input data while reducing the importance of others based on context. They are trained using gradient descent. Super-resolution techniques aim to enhance or reconstruct low-resolution images to a higher resolution, allowing for the recovery of more details, especially for small objects. For instance, SRGAN [33] was the first paper to apply GANs to the super-resolution domain. It combined GANs with SRResNet [34], introducing new loss functions such as content loss and adversarial loss to address the challenge of recovering high-frequency information in super-resolution. Region proposal is a strategy to design more suitable anchors for small objects. For example, YOLOv2 [35] uses anchor boxes to predict bounding boxes, effectively improving the model recall capability, which is particularly beneficial for small-object detection.

Currently, deep-learning-based small-object detection has found numerous applications, such as garbage waste management in smart cities [36]. In our work, we enhanced

small-object detection by incorporating three key techniques: a prediction head for small objects utilizing multiscale representation, coordinate attention [13] to improve semantic information in feature maps without a significant increase in computational load, and Deformable ConvNets v2 [12] convolution for adaptively learning feature point receptive fields, ultimately enhancing detection accuracy, particularly for small objects in complex environments.

3. Approach

In this paper, we first replace the YOLOv8 Darknet-53 backbone with FasterNet-T0 [11] to reduce model parameters and flops, speed up model training, and achieve model lightweighting. Secondly, we add a prediction head for small objects, generating low-level, high-resolution feature maps that are more sensitive to small-object detection. Thirdly, we introduce Coordinate Attention [13] to help the network find the regions of interest in the images. Finally, we replace tShe convolutions in the Neck with Deformable ConvNets v2 [12], and replace the 3×3 convolutions in the Bottleneck of the C2f structure with Deformable ConvNets v2. This deformable convolution can automatically augment the offsets of feature respective fields, leading to more accurate feature extraction and improved detection accuracy. The improved YOLOv8 overall structure is shown in Figure 1, and the details of the improvement modules are described in the following sections.

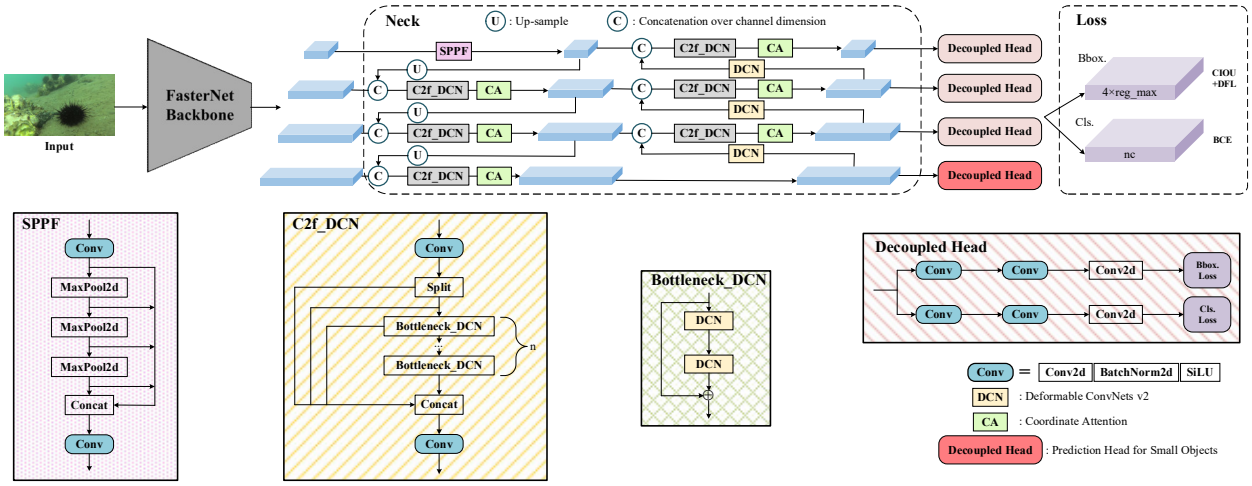


Figure 1. The overall architecture of the improved YOLOv8 network.

3.1. Faster Neural Networks

Efforts to design faster neural networks have been centered on reducing floating-point operations (FLOPs). However, it is important to note that a decrease in FLOPs does not necessarily translate to a proportional reduction in latency. Chen et al. [11] emphasize that the low FLOPS observed is primarily attributed to frequent memory access, particularly in operators like DWConv [10].

In response to these challenges, the authors introduce partial convolution (PConv) as a solution. PConv aims to improve spatial feature extraction while simultaneously minimizing redundant computation and memory access. This innovation is incorporated into the FasterNet architecture, a new neural network family featuring four hierarchical stages. Each stage integrates an embedding or merging layer for spatial downsampling and channel expansion. The FasterNet block structure, which is present within each stage, consists of a PConv layer, followed by two pointwise convolution [9] layers. PConv utilizes specific consecutive channels as representatives for computation, with an increased number of channels in the middle layer, and incorporates a shortcut connection to reuse input features. The overall architecture of FasterNet and how PConv works is shown in Figure 2.

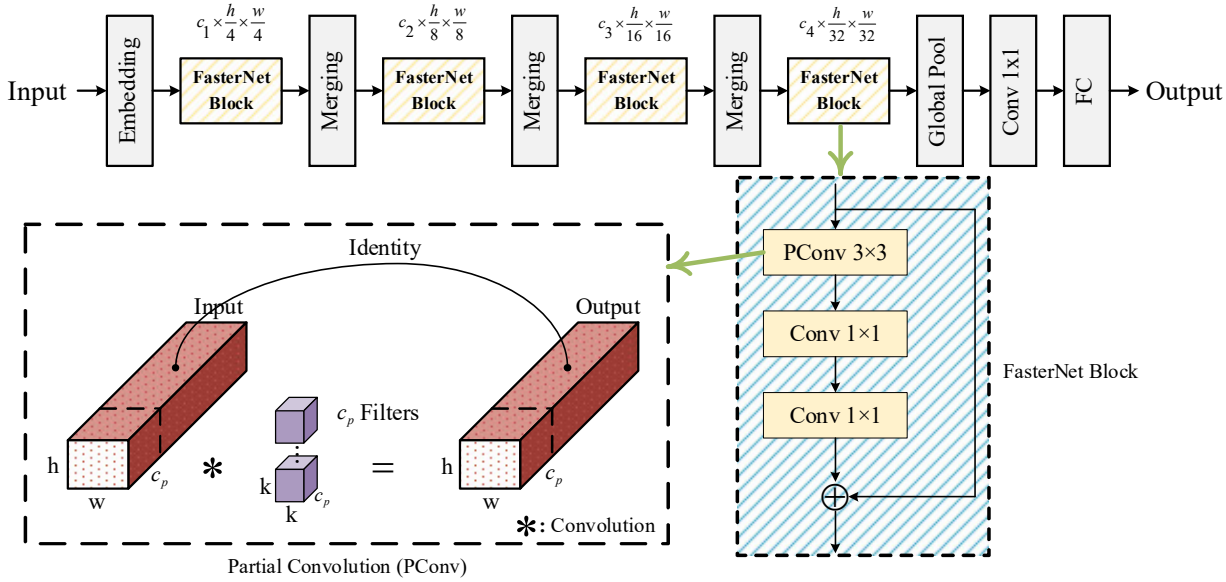


Figure 2. The comprehensive structure of FasterNet.

FasterNet has many variants. In our experiments, we replaced the YOLOv8s' Darknet-53 backbone with FasterNet-T0. This replacement led to a 22.52% reduction in model parameters (8.6 M vs. 11.1 M) and a 23.59% reduction in flops (21.7 B vs. 28.4 B), while causing only a slight drop in AP and AP50, of 0.81% and 0.87%, respectively.

3.2. Prediction Head for Small Objects

We explored the underwater dataset UTDAC2020 and identified numerous instances of extremely small objects. Zhang et al. [7] speculated that shallower features might be effective when the targets are small and the features of the targets are not obvious. Consequently, we introduced an additional prediction head specifically for detecting small objects with shallower features. This four-head structure, in conjunction with the other three prediction heads, mitigates the adverse impact of significant variations in object scale. As illustrated in Figure 2, our added prediction head derives from a high-resolution feature map, making it more sensitive to small objects. Through experimentation, we increased the number of channels in the high-resolution feature map detection head while reducing the channels in the low-resolution feature map detection head, maintaining model size and memory usage. This resulted in a significant improvement in small-object detection performance, increasing the AP by 1.2% overall and by 2.1% for small targets like scallops on the UTDAC2020 underwater dataset.

3.3. Coordinate Attention

The Coordinate Attention [13] module addresses attention mechanisms for mobile networks by embedding positional information into channel attention. It factorizes channel attention into two 1D feature-encoding processes, aggregating features independently along two spatial directions. This design captures long-range dependencies along one spatial direction while preserving precise positional information along the other. The module structure is depicted in Figure 3, showcasing a simple design that seamlessly integrates into classic mobile networks with minimal computational overhead. Demonstrating excellent performance, it excels in ImageNet classification, object detection, and semantic segmentation.

In underwater images, various factors often result in poor image features. Using coordinate attention not only extracts the attention area to help YOLOv8 cope with image lighting imbalances, blurriness, and glare, but also incurs minimal computational costs and minimal memory usage.

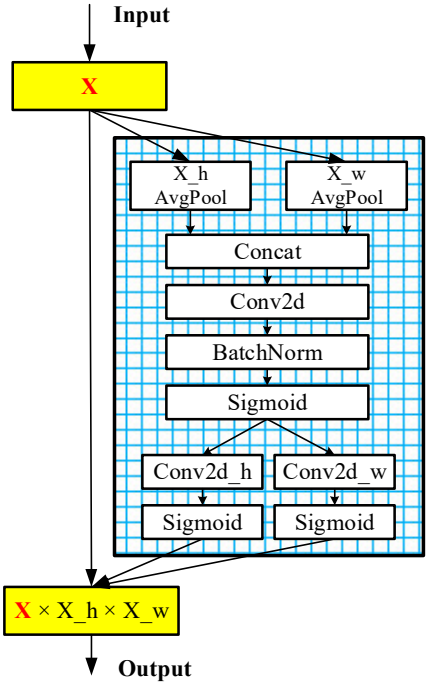


Figure 3. Coordinate attention module overview.

3.4. Deformable ConvNets V2

The modeling of geometric transformations in convolutional neural networks (CNNs) is inherently constrained by the fixed grid structures of the kernels. Dai et al. [37] have introduced two innovative modules, namely deformable convolution and deformable RoI pooling, which significantly augment the CNNs' ability to model geometric transformations.

Deformable convolution innovates by incorporating 2D offsets into standard grid sampling, enabling a flexible deformation of the sampling grid. To facilitate effective learning, offsets are derived from preceding feature maps through additional convolutional layers, ensuring deformation is adaptively conditioned on local input features. The above process can be represented as follows:

$$Y(p_0) = \sum_{p_n \in \mathrm{R}} w(p_\mathrm{n}) \cdot x(p_0 + p_n + \triangle p_n) \tag{1}$$

$$R = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\} \tag{2}$$

where $Y(p_0)$ signifies the output feature map value at position p_0, utilizing a 3×3 kernel grid (R) with dilation 1. Here, x represents the input feature map, w denotes the sampled values' weights, p_n enumerates the coordinates in R, and $\triangle p_n$ represents the deformable convolution augmented offsets.

Deformable RoI pooling introduces adaptive part localization in objects with varying shapes by incorporating learned offsets into regular RoI pooling positions. These offsets are derived from preceding feature maps and RoIs. In the RoI pooling process, given an input feature map x and an RoI of size w $\times$ h with the top-left corner at p_0, the RoI is divided into k $\times$ k bins to generate a k $\times$ k feature map Y. The process described above can be represented as follows:

$$Y(i,j) = \sum_{p \in bin(i,j)} x(p_0 + p + \Delta p_{ij})/n_{ij} \tag{3}$$

where $Y(i,j)$ denotes the values from deformable RoI pooling, and n_{ij} represents the number of pixels in spatial binning positions. Δp_{ij} is computed by a fully connected layer, generating normalized offsets $\Delta \hat{p}_{ij}$. These offsets are then modulated by a scalar

γ (empirically set to 0.1) and multiplied element-wise with RoI's width and height, expressed as $\Delta p_{ij} = \gamma \cdot \Delta \hat{p}_{ij} \circ (w, h)$.

While Deformable ConvNets v1 shows better spatial feature extraction capabilities compared to regular ConvNets, it can sometimes introduce irrelevant context, which can hurt the algorithm's performance. To address this, Zhu et al. [12] introduced Deformable ConvNets v2, which adds weights to the sampling points in Deformable ConvNets v1. Figure 4 illustrates how Deformable ConvNets v2 works on underwater datasets. This can be expressed as follows:

$$Y(p_0) = \sum_{p_n \in \mathrm{R}} w(p_\mathrm{n}) \cdot x(p_0 + p_n + \triangle p_n) \cdot \Delta m_{p_n} \tag{4}$$

$$Y(i,j) = \sum_{p \in bin(i,j)} x(p_0 + p + \Delta p_{ij}) \cdot \Delta m_{ij} / n_{ij} \tag{5}$$

where Δm_{p_n} and Δm_{ij} represent the modulation scalars for each position, with values ranging from 0 to 1, adding input features to adjust their strength at offset positions. This adjustment allows the module to change the spatial distribution of the samples and their mutual influence.

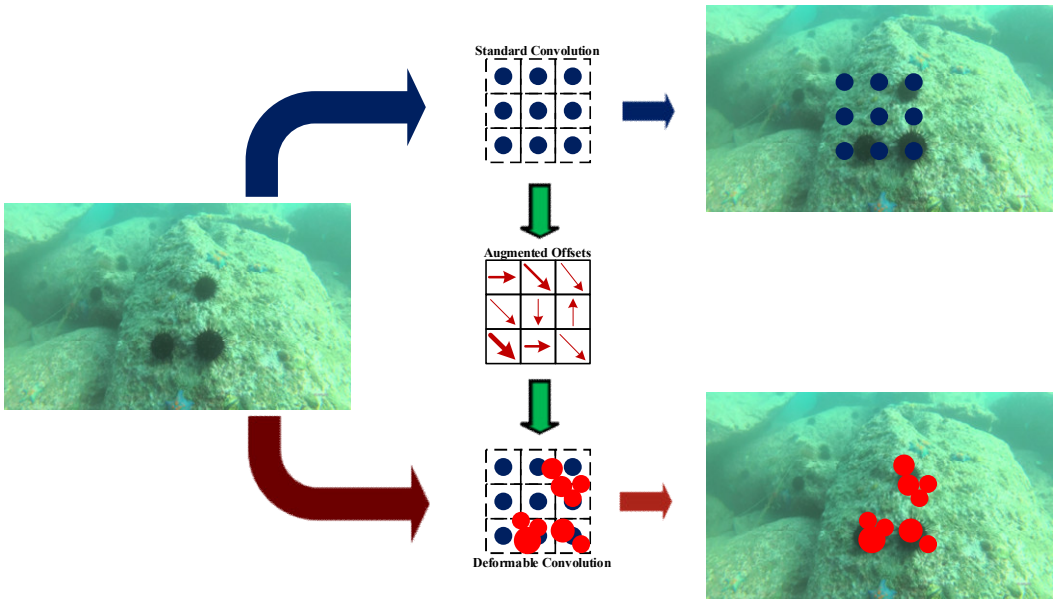


Figure 4. Illustration of Deformable ConvNets v2 on the underwater image.

In our work, we improve YOLOv8 by incorporating Deformable ConvNets v2. We replace the Conv layers in the YOLOv8 Neck, as well as the 3 × 3 convolution in the bottleneck of the C2f, with Deformable ConvNets v2. The experimental results show a significant improvement in accuracy, with an increase of 0.72% in AP on the underwater dataset. Specifically, the AP for the irregular category 'holothurian' increased by 1.39%.

4. Experiments

4.1. Datasets

We experimented with two challenging object detection datasets to assess and validate the generalization performance of our model.

4.1.1. UTDAC2020

UTDAC2020, originating from the 2020 Underwater Object Detection Algorithm Competition, serves as an underwater dataset with four classes: echinus, starfish, holothurian, and scallop. The dataset included 5168 training images and 1293 validation images, with four resolutions: 3840 × 2160, 1920 × 1080, 720 × 405, and 586 × 480. Notably, it presented

with significant imbalances in resolution and category samples, posing challenges for model training.

4.1.2. Pascal VOC

PASCAL VOC provides a comprehensive and standardized dataset for image recognition and classification. It organized an annual image recognition challenge from 2005 to 2012. The main dataset included VOC 2007 and VOC 2012, which are divided into four major categories and twenty subcategories, making them a benchmark for object detection algorithms. In the VOC 2007 dataset, there were 5011 annotated images in the trainval set and 4952 annotated images in the test set, totaling 9963 annotated pictures. In the VOC 2012 dataset, there were 11,540 annotated images in the trainval set. We trained our detector on the combined trainval dataset and evaluated its performance on the VOC 2007 test set.

4.2. Implementation Details

Table 1 provides details on the hardware and software setup used in the experiment.

Table 1. The experimental setting.

Environment	Versions or Model Number
CPU	Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz
GPU	GeForce RTX 2080 Ti, Two GPUs, Memory of 11 G
OS	Ubuntu 18.04
CUDA	V 10.2
CUDNN	V 7.6.5
PyTorch	V 1.12.1
Python	V 3.8.16

During the training, consistent training parameters were applied to each experimental group to ensure the precision of the experiments. The input resolution was configured with the longer side set to 640 pixels, preserving the original aspect ratio of the images, and the batch size was fixed at 32. In the training process, if the model did not show an improvement within 50 epochs, the training was terminated early, with a maximum of 300 epochs allowed. Optimization of the loss function was achieved through the utilization of the Stochastic gradient descent (SGD) algorithm, incorporating a momentum value of 0.937 and a weight decay coefficient of 5×10^{-4}. The initial learning rate was set at 0.01, and the confidence threshold was defined as 0.25. Mosaic data augmentation was employed, while all other parameters were kept consistent with those in YOLOv8.

During the inference, a standardized input resolution with the longer side set to 640 pixels was used, while preserving the original aspect ratio of the images. The confidence threshold was precisely defined at 0.001, and the intersection over union (IOU) threshold was established at 0.7. In the context of speed testing, singular GPU utilization was implemented, and the batch size was specifically set to 1, denoting the sequential processing of individual images.

In this context, we utilized the widely accepted metrics for object detection, as outlined in Table 2. The measurements for the parameters and FLOPs were evaluated with the longer side set to 640 pixels.

Table 2. The metrics used in our experiments.

Metrics	Description
AP_{50}	The mean average precision (mAP) at an intersection over union (IoU) of 0.50.
AP	The mAP at IoU of 0.50:0.05:0.95.
Parameters	The overall count of parameters in the network.
FLOPs	Floating-point operations per second.

4.3. Comparisons with Other State-of-the-Art Methods

4.3.1. Results on UTDAC2020

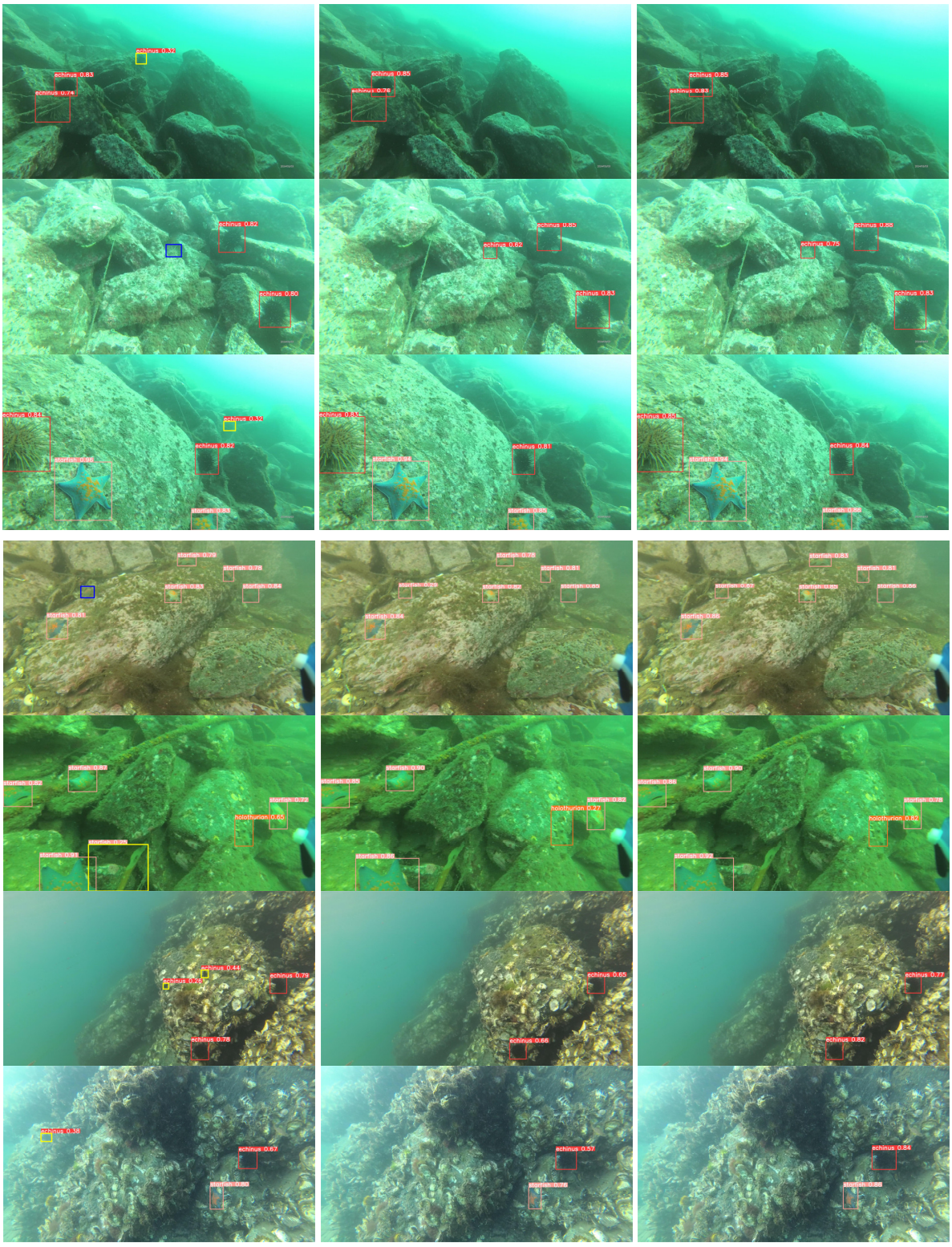
The experiment results obtained from the UTDAC2020 dataset are presented in Table 3. As this work focuses on model lightweighting research, we employed AP and AP50 to assess the model accuracy and used parameters, FLOPS, and model size to compare the model scale. The highest level is indicated by text in bold.

To validate the effectiveness of our proposed method, we compared it with the YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l models, and other well-established detectors results from the paper [6]. As shown in Table 3, our method reduces the parameters, FLOPS, and model size of the YOLOv8s model by 23.42% (8.5 M vs. 11.1 M), 10.21% (25.5 B vs. 28.4 B), and 22.73% (17 MB vs. 22 MB), respectively. However, because our model is specifically optimized for underwater scenarios, its accuracy surpasses even that of the larger YOLOv8l model, achieving an AP of 52.12% and AP50 of 85.49%. This represents an improvement of 1.62% (52.12% vs. 50.50%) and 0.76% (85.49% vs. 84.73%) compared to YOLOv8s, respectively. The experimental results confirm the effectiveness of our method in underwater scenarios and its high precision. To further enhance the model accuracy, especially considering the prevalence of small objects in underwater scenarios, we increased the image input of the longer side to 1280 pixels. This led to a further improvement in model accuracy, with an AP of 53.18% and an AP50 of 86.21%. We conducted processing speed tests using a single GeForce RTX 2080 Ti, testing over 1000 images from the UTDAC2020 dataset and averaging the results. Our method, at 640, achieved a processing speed of 68.03 frames per second (FPS), while at 1280, our method achieved 41.49 FPS. Despite a decrease in processing speed when increasing the input image size, both methods maintain real-time performance (30 FPS or better) [38]. Users can choose the most appropriate method based on the specific underwater scenario. To the best of our knowledge, our method achieves state-of-the-art (SOTA) performance on the UTDAC2020 dataset.

Table 3. Comparisons with different object detectors on UTDAC2020 dataset (The symbol * signifies enhanced version of the model.).

Method	Backbone	AP	AP_{50}	Parameters (M)	FLOPs (G)	Model Size (MB)
Faster R-CNN w/FPN [29]	ResNet50	44.50	80.90	41.14	63.26	~
Cascade R-CNN [39]	ResNet50	46.60	81.50	68.94	91.06	~
RetinaNet [40]	ResNet50	43.90	80.40	36.17	52.62	~
FCOS [41]	ResNet50	43.90	81.10	31.84	50.36	~
Deformable DETR [42]	ResNet50	46.60	84.10	~	~	~
Libra R-CNN [43]	ResNet50	45.80	82.00	41.40	63.53	~
Dynamic R-CNN [44]	ResNet50	45.60	80.10	41.14	63.26	~
ATSS [45]	ResNet50	46.20	82.50	31.89	51.58	~
Boosting R-CNN [6]	ResNet50	48.50	82.40	43.55	53.17	~
Boosting R-CNN * [6]	ResNet50	51.40	85.50	45.91	54.67	~
YOLOv8n	Darknet-53	49.07	82.73	3.0	8.1	6
YOLOv8s	Darknet-53	50.50	84.73	11.1	28.4	22
YOLOv8m	Darknet-53	51.74	85.11	25.8	78.7	50
YOLOv8l	Darknet-53	51.69	84.85	43.6	164.8	84
Ours	FasterNet-T0	52.12	85.49	8.5	25.5	17
Ours (1280)	FasterNet-T0	**53.18**	**86.21**	8.5	25.5	18

Thanks to the prediction head for small objects, DCNv2, and coordinate attention, false positives and false negatives in the Echinus and Starfish categories significantly decreased. To accurately illustrate the differences, the baseline YOLOv8s used the image with a longer side of 640 pixels as the input, while our method used input images with longer sides of 640 pixels and 1280 pixels, respectively. We visualized the detection results for the underwater UTDAC2020 dataset. As shown in Figure 5, the YOLOv8s model exhibits some instances of false positives (yellow boxes) and false negatives (blue boxes), especially with small objects.



YOLOv8s Ours (640) Ours (1280)

Figure 5. Comparison of the detection results of the URPC2020 underwater dataset with YOLOv8s using visualization (**Column Left:** YOLOv8s sets the longer side of the image input to 640 pixels, **Column Middle:** Ours method sets the longer side of the image input to 640 pixels, **Column Right:** Ours method sets the longer side of the image input to 1280 pixels). Additionally, false positives are indicated by yellow boxes, and false negatives are represented by blue boxes.

4.3.2. Results on Pascal VOC

To validate the generalization of our proposed method, we conducted experiments on the Pascal VOC dataset and compared the performance with two-stage detectors, one-stage detectors, and lightweight detectors. The experimental results are shown in Table 4, in which the performance indicators of the compared detectors were obtained from their original articles. As the results demonstrate, our method achieves high accuracy (mAP 84.4%) while maintaining lightweight parameters (8.5 M). Compared to YOLOv8s, it not only improves mAP by 0.5% (84.4% vs. 83.9%) but also reduces the parameter count by 23.42% (8.5 M vs. 11.1 M). Compared to YOLOv8m, our method significantly reduces these parameters by 67.18% (8.5 M vs. 25.9 M), while only experiencing a slight decrease in mAP, of 1.7% (84.4% vs. 86.1%). The results indicate that our method significantly improves performance in terms of lightweighting and model accuracy compared to previous mature detectors, demonstrating its generalization on the common dataset.

Table 4. Comparisons with different object detectors on PASCAL VOC dataset (The symbol * signifies enhanced version of the model.).

Method	Backbone	Input Size	mAP	Parameters (M)
Two-Stage Detector:				
Faster RCNN [46]	VGGNet	1000 × 600	73.2	134.7
Faster RCNN [5]	ResNet-101	1000 × 600	76.4	60.13
MR-CNN [47]	VGG16	1000 × 600	78.2	~
R-FCN [48]	ResNet50	1000 × 600	77.4	31.9
CoupleNet [49]	ResNet101	1000 × 600	82.7	~
DSOD300 [50]	DS/64-192-48-1	300 × 300	77.7	14.8
Boosting R-CNN [6]	ResNet50	1000 × 600	81.9	43.6
Boosting R-CNN* [6]	ResNet50	1000 × 600	83.0	45.9
One-Stage Detector:				
SSD512 [51]	VGG16	512 × 512	76.8	~
STDN513 [52]	DenseNet169	513 × 513	80.9	~
RefineDet512 [53]	VGG16	512 × 512	81.8	~
DSSD513 [54]	ResNet101	513 × 513	81.5	~
RetinaNet [40]	ResNet50	1000 × 600	77.3	36.2
FERNet [55]	VGG16 + ResNet50	512 × 512	81.0	~
DES512 [56]	VGG16	512 × 512	81.7	~
DFPR512 [57]	VGG16	512 × 512	81.1	~
EFIPNet512 [58]	VGG16	512 × 512	81.8	~
RFBNet512 [59]	VGG16	512 × 512	82.1	~
Lightweight detectors:				
SqueezeNet-SSD [60]	SqueezeNet	300 × 300	64.3	5.5
MobileNet-SSD [60]	MobileNet	300 × 300	68.0	5.5
Pelee [61]	PeleeNet	300 × 300	70.9	6.0
Tiny-DSOD [60]	G/32-48-64-80	300 × 300	72.1	1.0
YOLO detectors:				
YOLOv8n	Darknet-53	640 × 640	80.4	3.0
YOLOv8s	Darknet-53	640 × 640	83.9	11.1
YOLOv8m	Darknet-53	640 × 640	86.1	25.9
Ours	FasterNet-T0	640 × 640	84.4	8.5

4.4. Ablation Study

To assess the effectiveness of different modules, we conducted ablation experiments on the underwater UTDAC2020 dataset, and the results are shown in Table 5. We compared our proposed method with the baseline YOLOv8s. When we replaced the YOLOv8's backbone Darknet-53 with FasterNet-T0, the model parameters, FLOPS, and model size decreased by 22.52% (8.6 M vs. 11.1 M), 23.59% (21.7 B vs. 28.4 B), and 22.73% (17 MB vs. 22 MB), respectively. However, AP only decreased by 0.81% (49.69% vs. 50.50%),

demonstrating the effectiveness of FasterNet in lightweighting YOLOv8 for underwater environments. After adding the prediction head for small objects, AP increased by 1.2% (50.89% vs. 49.69%). Through experiments, we found that the low-resolution feature map detection head had an insignificant effect on the UTDAC2020 dataset and reduced its channel number, so the model size underwent a slight decrease instead of an increase.

Table 5. Ablation study on UTDAC2020.

Setting	AP	Echinus	Starfish	Holothurian	Scallop	Parameters (M)	FLOPs (B)	Model Size (MB)
Baseline-YOLOv8s	50.50	52.46	55.38	40.36	53.80	11.1	28.4	22
+FasterNet-T0	49.69	52.04	54.33	39.34	53.05	8.6	21.7	17
+FasterNet-T0, +Phead	50.89	53.28	55.21	39.94	55.15	8.0	30.7	16
+FasterNet-T0, +Phead, +CA	51.40	53.11	56.54	41.12	54.82	8.0	30.8	16
+FasterNet-T0, +Phead, +CA, +DCNv2 (640)	52.12	**53.92**	56.85	42.51	55.22	8.5	25.5	17
+FasterNet-T0, +Phead, +CA, +DCNv2 (1280)	**53.18**	53.08	**57.64**	**44.87**	**57.13**	8.5	25.5	18

Finally, based on our experience, increasing the input image resolution had a significant effect on small-object detection. By increasing the image input's longer side from 640 pixels to 1280 pixels, we achieved a 1.06% (53.18% vs. 52.12%) improvement in AP. This enhances the model's underwater detection capabilities and meets various speed and accuracy requirements in different scenarios.

5. Discussion

5.1. The Impact of High-Resolution Feature Maps (Prediction Head for Small Objects)

In order to validate the role of the prediction head for the small objects module in small-object detection, we visualized it using a small-object sample image from the UTDAC2020 dataset. As shown in Figure 6, the top image displays the detection results of YOLOv8s + FasterNet-T0, while the bottom image shows the detection results of the YOLOv8s + FasterNet-T0 + prediction head for small objects (Phead). By comparing these two images, we can observe the significant improvements when using the prediction head for the small objects module in small-object detection.

To further demonstrate the effectiveness of the prediction head for small objects module, we conducted an analysis from the perspective of feature maps using Figure 7. YOLOv8s has three detection heads, and we visualized nine examples of feature maps for each detection head, as shown in the left three columns in Figure 7. Our proposed method introduces an additional high-resolution feature map detection head (prediction head for small objects), totaling four detection heads. Similarly, we visualized nine examples of feature maps for each detection head, as shown in the middle three columns in Figure 7. Additionally, based on empirical knowledge, using large-sized input images increases the overall scale of feature maps in the network, which is advantageous for small-object detection. According to our experimental results, this indeed brings about substantial improvements. We also performed a feature map visualization for our method with the image input's longer side set to 1280 pixels as a comparison, as shown in the right three columns in Figure 7.

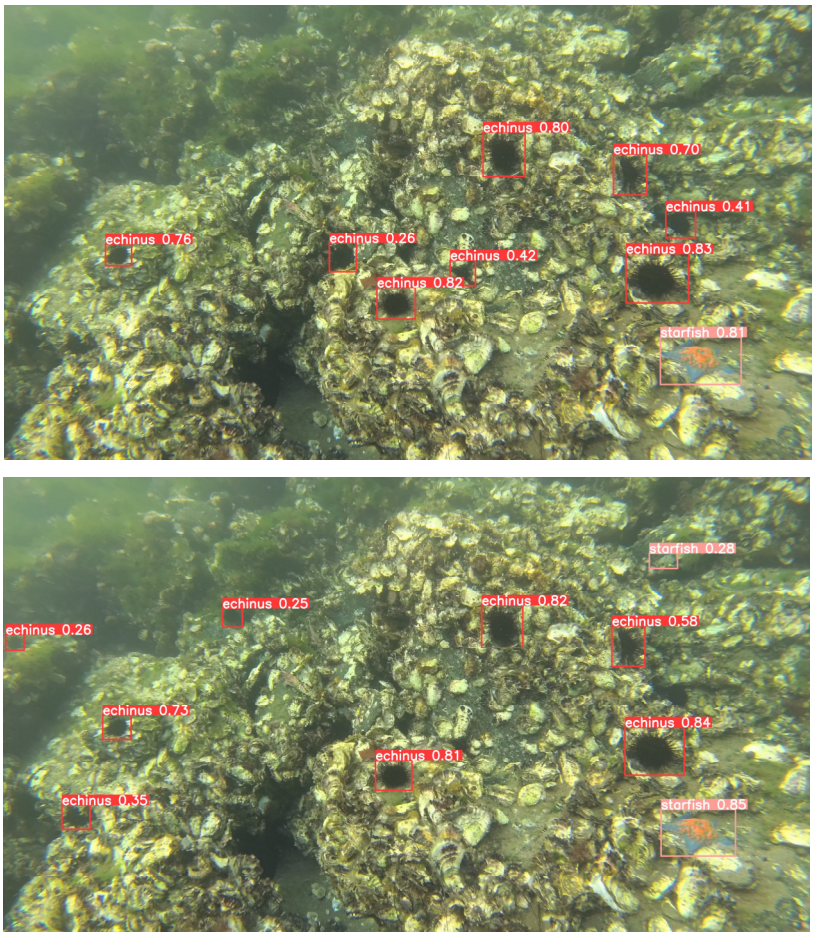


Figure 6. The detection results for a small object sample image from the UTDAC2020 dataset using YOLOv8s + FasterNet-T0 (**top**) and YOLOv8s + FasterNet-T0 + Phead (**bottom**).

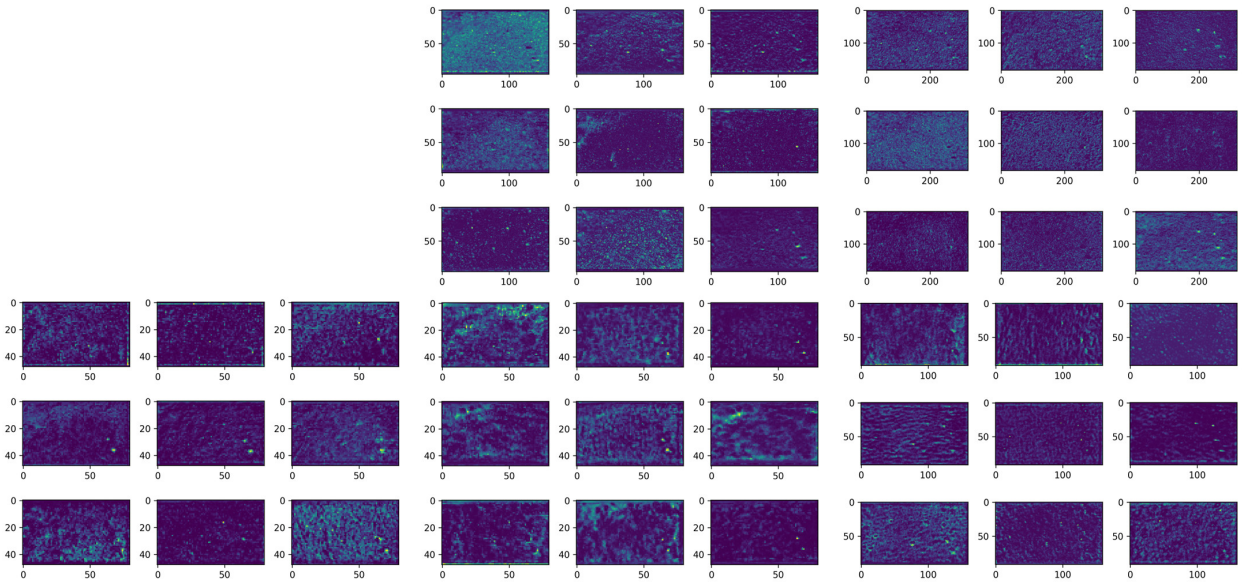


Figure 7. *Cont.*

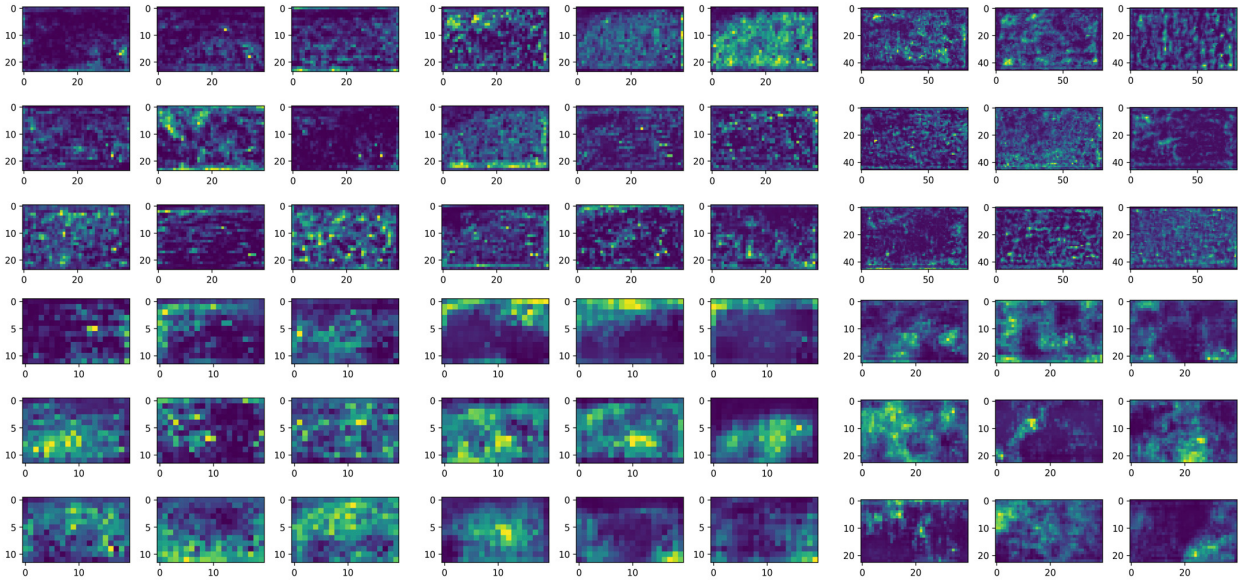


Figure 7. The feature maps of corresponding detection heads using different methods (YOLOv8s 640, ours 640 and ours 1280) for the underwater image of Figure 6.

Through observation, the object outlines in the first row of feature maps in Figure 7 are proven to be clearer (highlighted by yellow-green bright spots), while those in the second row are somewhat blurry, and the outlines are completely blurred from the third row onwards. The first row of Figure 7 represents the high-resolution feature map detection head that we added, further highlighting the effectiveness of the prediction head for small objects module. This also suggests that high-resolution feature maps are more conducive to detecting small objects.

5.2. The Impact of Low-Resolution Feature Maps on Large Object Detection

Similarly, we also analyzed the advantage of low-resolution feature maps for detecting large objects. Figure 8 shows the large object sample image from UTDAC2020 dataset with the detection results using our method (1280). We analyzed the three different scale feature maps (320 × 184, 160 × 92, 80 × 46) with our method (1280), using Grad-CAM [62] attention maps. As shown in Figure 9, there is a more concentrated attention on the large object in the 80 × 46 feature map, while it appears more dispersed in the 320 × 184 and 160 × 92 feature maps. This suggests that a low-resolution feature map is more favorable for detecting large objects.

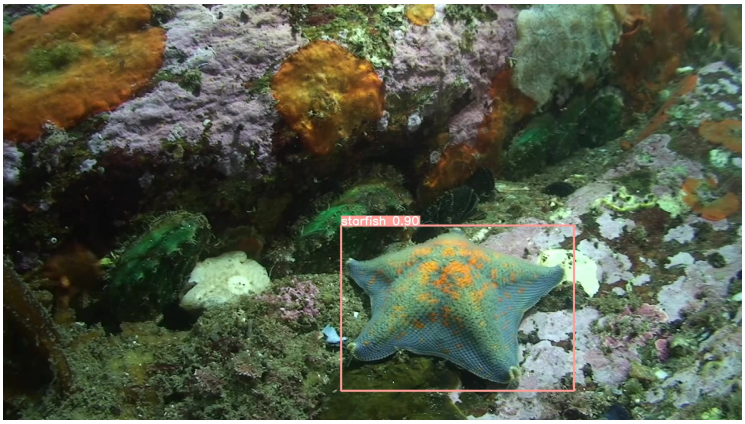


Figure 8. The large-object sample image from UTDAC2020 dataset with the detection results using our method (1280).

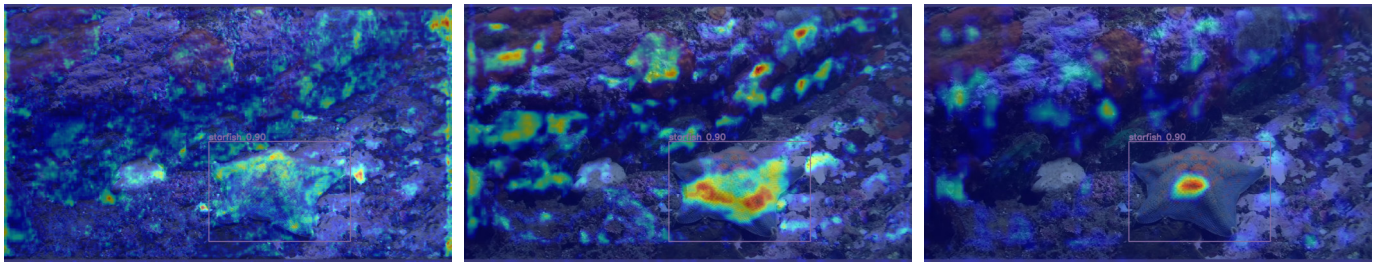


Figure 9. The feature maps visualizing the image in Figure 8 using our method (1280) at different scales. **Column 1**: feature map of 320 × 184. **Column 2**: feature map of 160 × 92. **Column 3**: feature map of 80 × 46.

5.3. Applicable in Various Underwater Marine Scenarios

To assess the generalization of our method on the underwater dataset, we tested the performance of the model trained on the UTDAC2020 dataset using an unseen underwater dataset, UODD [63], which includes underwater cultured products such as sea cucumbers, sea urchins, and scallops. We selected cases with occlusion, dense small objects, and complex backgrounds to simulate real underwater scenarios, and the results are shown in Figure 10. The results demonstrate that our model performs well and generalizes effectively in real underwater scenarios.

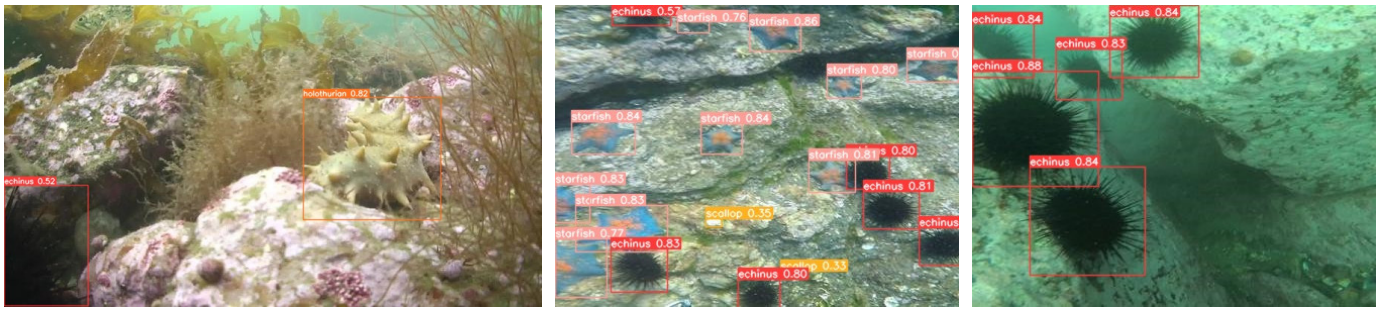


Figure 10. The prediction results of our method (640) on the unseen underwater dataset UODD.

However, in underwater environments with poor visibility and a high degree of target overlap, our detector may miss certain targets. We speculate that incorporating an underwater image enhancement network during the preprocessing of model data may improve this phenomenon, albeit at the cost of increased model parameters, FLOPs, and a slower inference speed. It is worth considering whether integrating an underwater image enhancement network into the model would be beneficial.

6. Conclusions

Underwater object detection faces complex challenges, such as uneven lighting, low contrast, dense object distributions, occlusions, and constraints on computational resources and storage in underwater embedded devices, leading to limitations in the performance of conventional object detectors. This work enhances the YOLOv8 algorithm, a cutting-edge object detection approach, with four modules aimed at lightweighting and enhancing small-object detection accuracy. According to experiments, adding the prediction head for small objects module is the most direct and effective method to improve the accuracy of detecting small objects. The use of Deformable ConvNets also significantly enhances the detection of small objects. The results demonstrate that our approach achieves a state-of-the-art (SOTA) performance on the UTDAC2020 underwater dataset, maintaining lightweight parameters (8.5 M), low flops (25.5 B), and a small model size (18 MB), while achieving an accuracy of AP 53.18% and AP50 86.21%. Notably, it even surpasses the larger YOLOv8l model with an AP of 51.69%. Additionally, parallel experiments on the Pascal VOC dataset show

the generalization of our approach, outperforming many well-established detectors while maintaining a lightweight model.

In future work, we will continue to explore models that balance model size and detection accuracy to further advance the field of underwater object detection. Additionally, the scarcity of underwater datasets remains a challenge, requiring more high-quality data to effectively improve model performance in underwater environments.

Author Contributions: Conceptualization, W.S.; methodology, M.Z. and Z.W.; validation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, M.Z., Z.W., W.S., D.Z. and H.Z.; investigation, M.Z.; supervision, W.S. and M.Z.; funding acquisition, W.S. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (61972240), and the Young Scientists Fund of the National Natural Science Foundation of China (42106190).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The UTDAC2020 underwater dataset presented by Song et al. and the dataset is available at https://doi.org/10.48550/arXiv.2206.13728, accessed on 11 December 2023.

Acknowledgments: This research was supported by the Digital Oceanography Institute of Shanghai Ocean University.

Conflicts of Interest: We declare no conflicts of interest.

References

1. Deans, C.; Marmugi, L.; Renzoni, F. Active underwater detection with an array of atomic magnetometers. *Appl. Opt.* **2018**, *57*, 2346–2351. [CrossRef]
2. Czub, M.; Kotwicki, L.; Lang, T.; Sanderson, H.; Klusek, Z.; Grabowski, M.; Szubska, M.; Jakacki, J.; Andrzejewski, J.; Rak, D. Deep sea habitats in the chemical warfare dumping areas of the Baltic Sea. *Sci. Total Environ.* **2018**, *616*, 1485–1497. [CrossRef]
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
5. 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.
6. Song, P.; Li, P.; Dai, L.; Wang, T.; Chen, Z. Boosting R-CNN: Reweighting R-CNN samples by RPN's error for underwater object detection. *Neurocomputing* **2023**, *530*, 150–164. [CrossRef]
7. 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]
8. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
9. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
10. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
11. Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. *arXiv* **2023**, arXiv:2303.03667.
12. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
13. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
14. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
15. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
16. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.

17. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3490–3499.
18. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
20. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
21. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
22. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
23. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
24. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
25. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
26. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
27. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
28. Mehta, S.; Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv* **2021**, arXiv:2110.02178.
29. Huang, H.; Zhou, H.; Yang, X.; Zhang, L.; Qi, L.; Zang, A.-Y. Faster R-CNN for marine organisms detection and recognition using data augmentation. *Neurocomputing* **2019**, *337*, 372–384. [CrossRef]
30. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
31. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef]
32. 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.
33. Takano, N.; Alaghband, G. Srgan: Training dataset matters. *arXiv* **2019**, arXiv:1903.09922.
34. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
35. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
36. Alsubaei, F.S.; Al-Wesabi, F.N.; Hilal, A.M. Deep learning-based small object detection and classification model for garbage waste management in smart cities and iot environment. *Appl. Sci.* **2022**, *12*, 2281. [CrossRef]
37. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
38. Sadeghi, M.A.; Forsyth, D. 30hz object detection with dpm v5. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part I 13. pp. 65–79.
39. 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.
40. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
41. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
42. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
43. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.

44. Zhang, H.; Chang, H.; Ma, B.; Wang, N.; Chen, X. Dynamic R-CNN: Towards high quality object detection via dynamic training. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part XV 16. pp. 260–275.
45. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9759–9768.
46. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 91–99.
47. Gidaris, S.; Komodakis, N. Object detection via a multi-region and semantic segmentation-aware cnn model. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1134–1142.
48. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
49. Zhu, Y.; Zhao, C.; Wang, J.; Zhao, X.; Wu, Y.; Lu, H. Couplenet: Coupling global structure with local parts for object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4126–4134.
50. 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.
51. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14; pp. 21–37.
52. Zhou, P.; Ni, B.; Geng, C.; Hu, J.; Xu, Y. Scale-transferrable object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 528–537.
53. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
54. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
55. Fan, B.; Chen, W.; Cong, Y.; Tian, J. Dual refinement underwater object detection network. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part XX 16. pp. 275–291.
56. Zhang, Z.; Qiao, S.; Xie, C.; Shen, W.; Wang, B.; Yuille, A.L. Single-shot object detection with enriched semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5813–5821.
57. Kong, T.; Sun, F.; Tan, C.; Liu, H.; Huang, W. Deep feature pyramid reconfiguration for object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 169–185.
58. Pang, Y.; Wang, T.; Anwer, R.M.; Khan, F.S.; Shao, L. Efficient featurized image pyramid network for single shot detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7336–7344.
59. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
60. Li, Y.; Li, J.; Lin, W.; Li, J. Tiny-DSOD: Lightweight object detection for resource-restricted usages. *arXiv* **2018**, arXiv:1807.11013.
61. Wang, R.J.; Li, X.; Ling, C.X. Pelee: A real-time object detection system on mobile devices. In Proceedings of the Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 1967–1976.
62. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
63. Jiang, L.; Wang, Y.; Jia, Q.; Xu, S.; Liu, Y.; Fan, X.; Li, H.; Liu, R.; Xue, X.; Wang, R. Underwater species detection using channel sharpening attention. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 4259–4267.