激光与光电子学进展
先进成像

# 基于YOLOv8轻量化水下光学图像识别算法

成顺[1,2]，李建荣[1*]，王志乾[1]，刘绍锦[1]，王牧原[1,2]
[1]中国科学院长春光学精密机械与物理研究所，吉林 长春 130033；
[2]中国科学院大学大珩学院，北京 100049

**摘要** 针对水下光学目标识别算法识别精度低、计算复杂度高的情况，提出一种自动色彩均衡(ACE)图像增强的轻量化YOLOv8水下光学识别算法。首先，使用ACE图像增强算法对图像进行预处理。其次，用改进的SENetV2骨干网络替换YOLOv8骨干网络，提高网络的特征提取能力，并用轻量级的跨尺度特征融合模块替换颈部网络，降低计算量。然后，用DySample替换传统上采样器，提升图像处理的效率，并在头部改进DyHead检测头，增强模型对目标的感知能力。最后，用基于最小点距离交并比(MPDIoU)的InnerMPDIoU替换YOLOv8的损失函数，提高边界框回归的准确率。实验结果表明，提出的SCDDI-YOLOv8算法在URPC2020数据集和UWG数据集上平均精度均值达到77.3%和71.5%，相比原YOLOv8n算法，所提算法的参数量下降～20.7%、浮点运算量减小$6\times10^8$、模型大小减小1.2 MB。与其他先进算法相比，所提算法能满足边缘设备计算资源敏感的需求。

**关键词** 水下光学图像识别；自动色彩均衡图像增强；轻量化YOLOv8；跨尺度特征融合模块；InnerMPDIoU
**中图分类号** TP393 **文献标志码** A **DOI**：10.3788/LOP241313

# Lightweight Underwater Optical Image Recognition Algorithm Based on YOLOv8

Cheng Shun[1,2], Li Jianrong[1*], Wang Zhiqian[1], Liu Shaojin[1], Wang Muyuan[1,2]
[1]*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun* 130033, *Jilin, China*;
[2]*Daheng College, University of Chinese Academy of Sciences, Beijing* 100049, *China*

**Abstract** To address the challenges of low recognition accuracy and high computational complexity in underwater optical target recognition algorithms, a lightweight YOLOv8 underwater optical recognition algorithm based on automatic color equalization (ACE) image enhancement is proposed. Initially, we apply the ACE image enhancement algorithm to preprocess images. Subsequently, we improve the feature extraction capabilities by replacing the YOLOv8 backbone with an upgraded SENetV2 backbone network. To further decrease computational quantity, we introduce a lightweight cross-scale feature fusion module in place of the neck network. Then, we utilize DySample as a substitute for the traditional upsampler to improve image processing efficiency. We refine the DyHead detection head to better perceive targets. Finally, we enhance the accuracy of bounding box regression by replacing loss function of YOLOv8 with InnerMPDIoU based on the minimum point distance intersection ratio (MPDIoU). Experimental results show that the proposed SCDDI-YOLOv8 algorithm achieves a mean average precision of 77.3% and 71.5% on the URPC2020 and UWG datasets, respectively, while reducing parameters by ～20.7%, floating-point operations by $6\times10^8$, and model size by 1.2 MB compared with the original YOLOv8n. Compared with other advanced algorithms, the proposed algorithm can meet the sensitive computational needs of edge devices.

**Key words** underwater optical image recognition; automatic color equalization image enhancement; lightweight YOLOv8; cross-scale feature fusion module; InnerMPDIoU

## 1 引 言

水下目标识别技术是水下探测任务采用的重要技术。传统水下目标识别主要依靠人工巡检和目视观察，这种方法不仅效率低下，而且存在较大的安全隐患。近年来，由于深度学习具备去人工化、自动特征学

**收稿日期**：2024-05-17；**修回日期**：2024-06-14；**录用日期**：2024-07-29；**网络首发日期**：2024-08-09
**基金项目**：吉林省科技发展计划(20210201048GX)
**通信作者**：*lijianrong@ciomp.ac.cn

习、数据拟合能力强等优点，其在目标识别系统中得到广泛应用。

然而由于水体对光具有吸收作用，水下图像的色彩往往受到严重影响，出现明显的亮度降低和失真现象[1]。另外，水体对光的散射作用可导致图像特征模糊，还可能形成严重的背景噪声[2]。多种因素的作用使水下图像的质量严重下降。水下环境的复杂性增加了识别水下目标的难度，严重降低对水下目标的识别精度。为解决这个严峻的问题，基于水下光学图像的目标识别算法引起学者的广泛兴趣。

目标识别算法可分为一阶段算法如 YOLO[3]和 SSD[4]等、二阶段算法如 Faster R-CNN[5]等，以及目前最新的 DETR(detection Transformer)端到端方法如实时 DETR(RT-DETR)[6]。其中：一阶段算法直接将目标的边界框和概率视为回归问题；二阶段算法先生成候选目标区域，再对这些区域进行进一步的分类和定位；RT-DETR 利用 Transformer 的先进思想，对输入特征进行高效编码，再利用解码器将这些特征用于目标识别，摒弃非极大值抑制(NMS)操作。对于以上 3 种识别算法，国内外学者进行了一系列的研究。姜丽梅等[7]利用改进的 RT-DETR 算法检测图像中的动态物体并根据检测框划分动态区域，但 RT-DETR 参数量太大，计算复杂度太高。宋绍剑等[8]利用 Mask R-CNN 对水下生物进行识别，准确率达到 97.30%，Mask R-CNN 的识别精度较高，但增加了计算资源的开销。王蓉蓉等[9]提出改进的 CenterNet 对水下目标进行识别，但对比 YOLO 系列算法，CenterNet 在速度、精度和计算量方面很难达到综合的平衡。Zhang 等[10]提出基于 MobileNetv2 和注意力特征融合模块(AFFM)的 YOLOv4 水下目标检测方法，但 MobileNetv2 的计算效率不高，且泛化能力有限。Yu 等[11]将卷积块注意力模块(CBAM)集成到 YOLOv4 算法中，以便在物体密集的场景中找到注意力区域，但 CBAM 包含通道注意力模块和空间注意力模块，计算较复杂。赵磊等[12]利用 YOLOv5 对瓶盖封装缺陷进行检测，并利用轻量化骨干网络对模型进行改进。吴萌萌等[13]在 YOLOv5 的基础上设计自适应双向特征融合网络，提高模型对小目标的检测精度。Wu 等[14]提出信道信息改进的通道路径聚合网络-特征金字塔网络(PAN-FPN)的 YOLOv5s 算法，并将其用于水下垃圾检测。Zhang 等[15]提出用于水下目标检测的基于全局注意力机制(GAM)和深度自动化机器学习(DAMO)融合的 YOLOv5 算法，GAM 在通道、空间宽度和高度这 3 个维度上捕捉特征信息，能处理跨维度交互信息，但 GAM 计算复杂度相对较高。Guo 等[16]用 FasterNet 模块取代 YOLOv8 骨干网络，使模型轻量化，但原检测头在不同的检测任务中不能动态调整特征通道。以上算法都有一定的局限性。另外，二阶段算法速度较慢且计算资源需求较高，需提前设置候选框，DETR 方法对小目标的检测效果较差，且计算复杂度较高，不适用于水下嵌入式系统或移动设备。

为解决以上问题，本文提出基于图像增强和轻量化 YOLOv8 网络模型的水下光学图像目标识别算法，以实现对水下目标图像的高精度检测，同时满足轻量化需求。

## 2 自动色彩均衡(ACE)图像增强

ACE 算法[17]基于 Retinex 理论[18]，考虑图像中颜色和亮度的空间位置关系，进行局部特性的自适应滤波，用像素点之间的亮度差异信息来校正图像，使图像的色彩分布更加均衡和自然。ACE 算法包括区域自适应滤波和色调重整拉伸、动态扩展两个步骤。

对输入图像的像素进行自适应滤波，完成色差校正，得到空域重构后的中间图像，计算公式为

$$\boldsymbol{S}_{\mathrm{c}}(p)=\sum_{j\in Q,j\neq p}\frac{R[I_{\mathrm{c}}(p)-I_{\mathrm{c}}(j)]}{d(p,j)} \tag{1}$$

式中：$Q$ 为图像像素集；$\boldsymbol{S}_{\mathrm{c}}$ 为得到的中间图像；$I_{\mathrm{c}}(p)-I_{\mathrm{c}}(j)$ 为像素点 $p$ 和 $j$ 之间的灰度值之差；$d(p,j)$ 为两个像素点的欧氏距离，映射出滤波的区域适应性；$R(x)$ 为亮度表现函数，本文选择经典饱和函数。令 $\alpha$ 为饱和函数的阈值，饱和函数的计算公式为

$$R(x)=\begin{cases}1, & x<-\alpha\\ \dfrac{x}{\alpha}, & -\alpha\leqslant x\leqslant\alpha\\ -1, & x>\alpha\end{cases} \tag{2}$$

对图像的动态范围进行全局调整，并使图像满足灰度世界理论和白斑点假设。算法首先针对单通道，再延伸应用到 RGB 彩色空间的 3 通道图像，最后将式(1)中得到的中间量拉伸映射到[0,255]中，计算公式为

$$\boldsymbol{F}(x)=\frac{\boldsymbol{S}_{\mathrm{c}}(x)-\min(\boldsymbol{S}_{\mathrm{c}})}{\max(\boldsymbol{S}_{\mathrm{c}})-\min(\boldsymbol{S}_{\mathrm{c}})} \tag{3}$$

式中：$\boldsymbol{F}(x)$ 为得到的最终结果。这种算法能实现具有局部和非线性特征的图像亮度、色彩与对比度的调整，同时满足灰色世界理论假设和白色斑点假设，使图像整体看起来和谐，色调自然。

为验证 ACE 算法的性能，对比常见图像增强方法如暗通道先验(DCP)[19]、直方图均衡化(HE)、自适应直方图均衡化(AHE)、多尺度 Retinex(MSR)[20]、单尺度 Retinex(SSR)等，结果如图 1 所示。DCP 算法通过计算每个像素局部邻域内的最小值来构建暗通道图像，利用这个暗通道图像来估计透射率，根据估计的透射率和大气光恢复出清晰的无雾图像，但大气光模型并不适用于水下光学。HE 可有效增强图像的对比度和视觉效果，使图像的细节更加清晰，但会导致图像的细节消失，且会造成不自然的过分增强。AHE 在局部应用 HE，限制局部对比度增强的幅度，但 AHE 在处

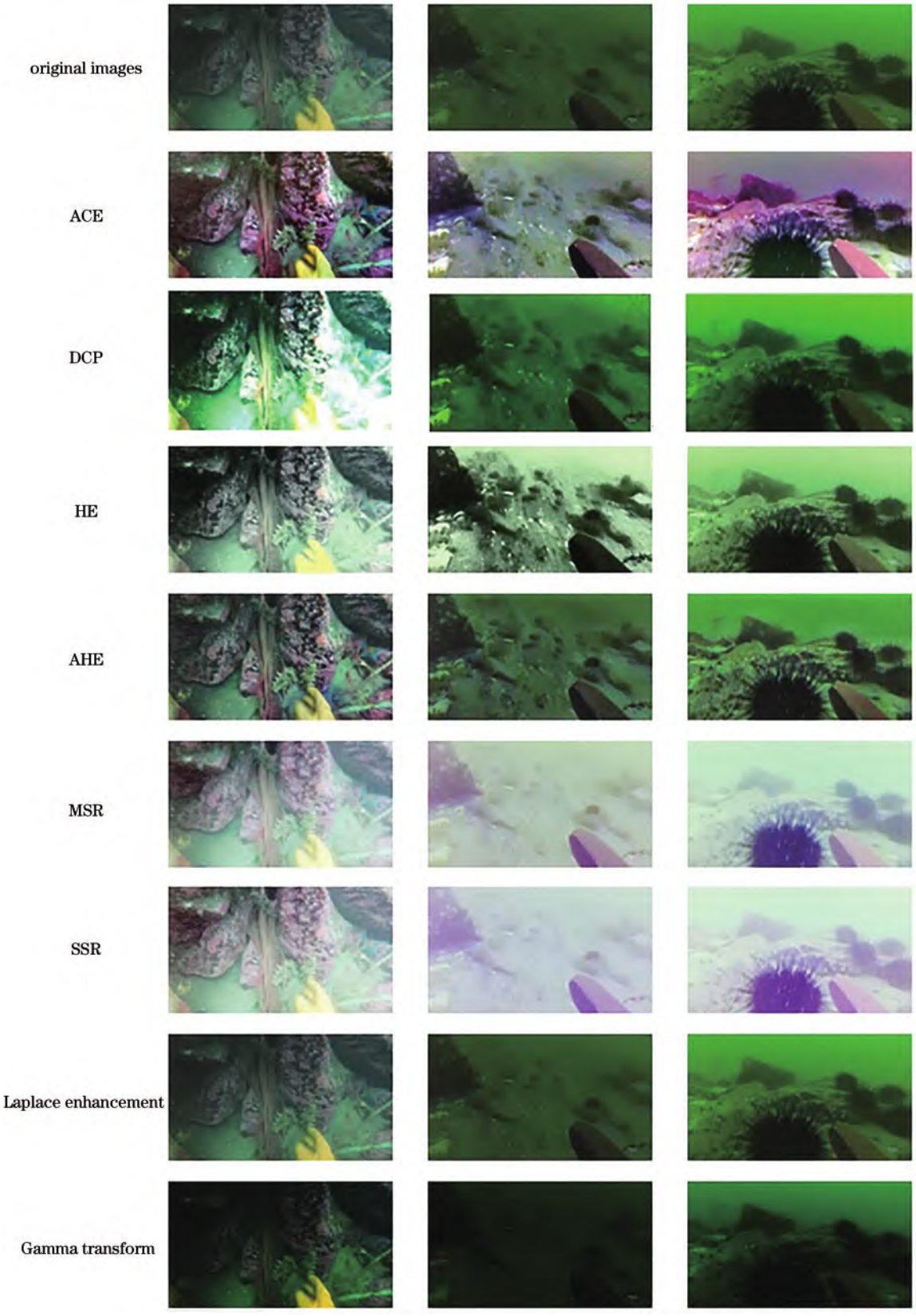


图 1　图像增强效果对比

Fig. 1　Comparison of the image enhancement effects

理具有特殊纹理或结构的图像时无法准确保留其细节信息。MSR[20]模拟人眼对光线的感知过程，计算图像中每个像素周围区域的平均亮度，并将其作为该像素的参考亮度从而增强图像，但在计算中会出现光晕现象和亮区域细节丢失问题。为定量评估增强算法的效果，以信息熵、均值、标准差和平均梯度作为评价指标，对比结果如表 1 所示。由表 1 可知，与对比算法的结果相比，ACE 算法的效果最好。

表 1　增强算法定量分析

Table 1　Quantitative analysis of the enhanced algorithms

| Algorithm | Information entropy | Mean | Standard deviation | Average gradient |
|---|---|---|---|---|
| **ACE** | **7.52** | 0.36 | 0.19 | **459** |
| DCP | 7.51 | 0.53 | 0.34 | 419 |
| HE | 7.97 | 0.50 | 0.28 | 378 |
| AHE | 7.29 | 0.35 | 0.16 | 386 |
| MSR | 7.52 | 0.63 | 0.19 | 196 |
| SSR | 5.54 | 0.80 | 0.06 | 69 |
| Laplace enhancement | 6.92 | 0.29 | 0.13 | 226 |
| Gamma transform | 6.44 | 0.12 | 0.12 | 138 |

## 3　改进 YOLOv8 模型

YOLOv8 模型是一阶段性能良好的目标检测算法。本文以 YOLOv8 模型为基础，设计水下光学图像识别算法，并将其命名为 SCDDI-YOLOv8。所提模型结构主要包含改进的骨干网络、跨层特征融合网络、改进上采样算子、用于预测的检测头和改进的损失函数。首先将经过 ACE 增强后的图像输入模型中，由骨干网络进行特征提取，将图像转化为具有丰富语义信息的特征表示。然后为使网络更好地关注输入数据中的重要部分，在骨干网络中加入 SENetv2 结构，根据不同通道的重要性来分配权重，帮助网络更好地利用输入数据中的不同特征。接着在跨尺度特征融合模块（CCFM）中将骨干网络中的丰富语义信息进行融合，将不同尺度的特征图整合起来，帮助模型更好地捕捉图像中的细节信息。随后在上采样过程中利用 DySample 点采样提高资源效率，使模型更加轻量化，更适用于水下边缘计算设备。接下来 DyHead 根据输入图像的尺寸和特征图的通道数，动态调整检测头的宽度和深度从而执行具体的目标识别任务。最后以基于最小点距离交并比（MPDIoU）的 InnerMPDIoU 为损失函数，从而更好地计算模型预测的边界框与真实边界框之间的重叠程度，帮助模型更好地定位目标并提高检测性能。改进的 SCDDI-YOLOv8 模型结构如图 2 所示，其中 SPPF 为快速空间金字塔池化。

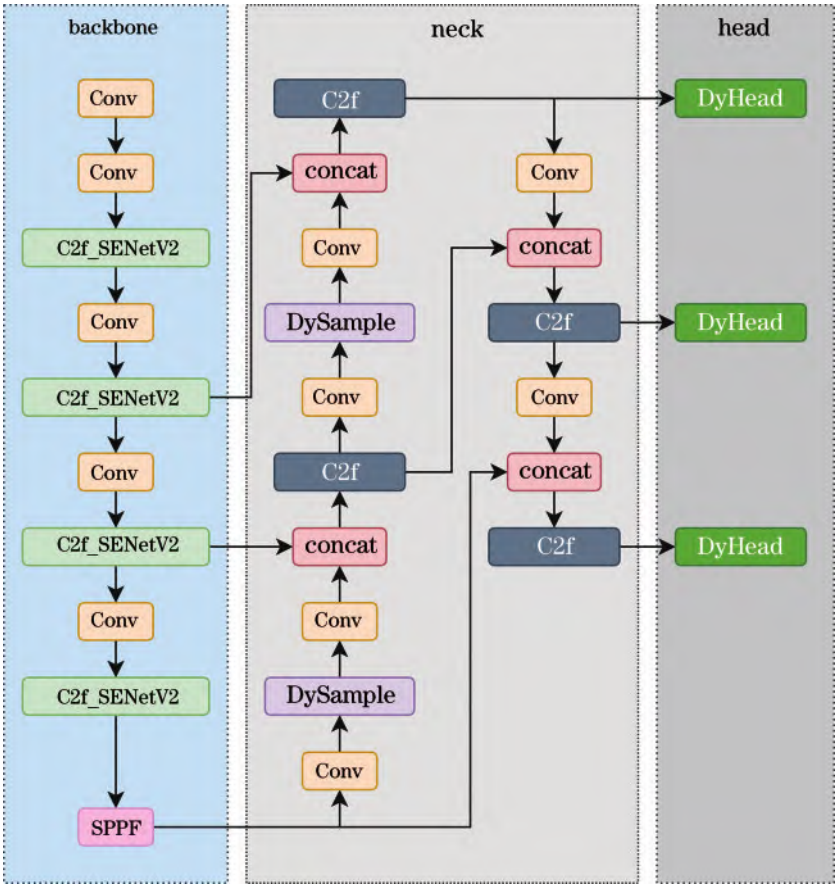


图 2　SCDDI-YOLOv8 模型结构

Fig. 2　Structure of the SCDDI-YOLOv8 model

### 3.1　改进骨干网络

SENetV2[21] 是改进的 SENet 网络，其引入挤压聚合激励（SaE）模块来提升网络的表征能力。SaE 模块利用多分支的稠密层来增强网络的特征表示能力，从而提升模型的性能。在 SENetV2 网络中，利用标准卷积操作处理输入特征后，首先利用全局平均池化来挤压特征，然后通过多分支全连接（FC）层和激活函数来获取通道权重，最后对卷积特征进行缩放。这种

设计使网络能自动学习特征图的每个通道的重要程度，并据此为每个特征赋予权重，从而使网络更关注重要的特征通道。将C2f与SENetV2网络结合，用结合网络替换骨干网络中传统的C2f网络，可在保持较低模型复杂度的同时提高模型的分类精度。改进的C2f_SENetV2结构如图3所示，其中CBS为卷积＋批归一化(BN)＋SiLU激活函数。

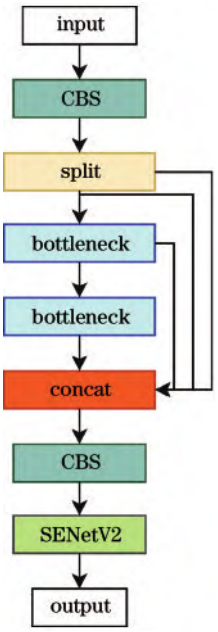


图3　C2f_SENetv2网络结构

Fig. 3　Structure of the C2f_SENetv2 network

### 3.2　改进特征融合网络

采用RT-DETR的CCFM替换YOLOv8中的特征融合模块。不同于PANet结构，CCFM在融合路径上插入由卷积层组成的Fusion模块，模块中的卷积层负责融合相邻尺度的特征。CCFM可有效合并不同尺度的特征，从而提升模型对不同尺度目标的识别能力，且能在保持较高的运算效率的同时，提高模型对小尺度对象的检测能力，适用于计算资源有限的设备。CCFM结构如图4所示。

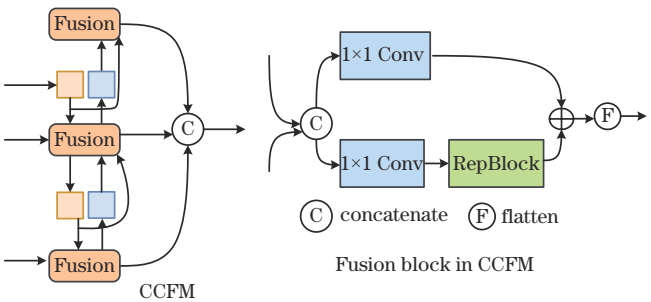


图4　CCFM网络结构

Fig. 4　Structure of the CCFM network

### 3.3　改进上采样器

目前，很多研究人员已提出很多优秀的动态上采样器如内容感知特征重组(CARAFE)[22]，使原始的网络性能得到提升。但CARAFE在进行实时处理时速度较慢，泛化能力较弱。因此，2023年Liu等[23]提出先进的DySample采样器，从点采样的角度制定上采样。DySample首先根据输入特征图的内容和结构，自适应确定初始的采样位置并设置偏移范围，以便捕获重要的特征信息。然后利用生成的内容感知偏移，DySample执行基于点采样的上采样操作，能更有效地利用特征信息，同时减少不必要的计算量。最后将采样得到的特征进行重组，并输出最终的上采样结果。该方法可保留重要的特征信息，并提高上采样的准确性。DySample网络结构如图5所示，其中$H$、$W$为特征垂直和水平方向的维度，$C$为输入特征通道数，$s$为上采样尺度因子。

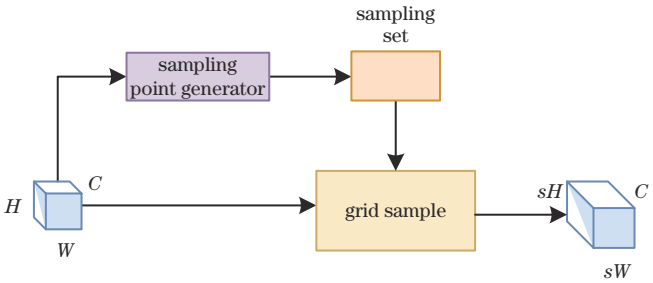


图5　DySample网络结构

Fig. 5　Structure of the DySample network

### 3.4　改进检测头

检测头的功能是目标识别与定位，常见检测头有基础检测头和AsDDet检测头。基础检测头从特征图中提取信息，输出边界框预测和类别概率。AsDDet检测头采用非对称结构设计，在调整通道数后将骨干网络的特征图划分为两个预测分支，解耦分类和回归两个任务。但AsDDet检测头对计算资源的要求较高，且泛化能力较弱，不适合用作通用检测头。为克服以上缺点，使用动态检测头(DyHead)[24]替代基础检测头。DyHead通过自注意力机制来增强目标检测头的尺度感知能力、空间感知能力和任务感知能力：在特征层级之间应用自注意力机制，可帮助模型更好地识别不同大小的对象；在空间位置之间应用自注意力机制，使模型能更关注图像中的特定区域，提高对目标位置的识别准确性；在输出通道之间应用自注意力机制，使模型能根据不同的任务需求调整注意力分布，提升模型的灵活性和适应性。DyHead网络结构如图6所示。

3个感知增强模块如图6所示，通过3个位置的注意力模块，即可得到增强后的输出：

$$W_{\mathrm{L}}(\boldsymbol{F})=\pi_{\mathrm{C}}\left\{\pi_{\mathrm{S}}\left[\pi_{\mathrm{L}}(\boldsymbol{F})\boldsymbol{F}\right]\boldsymbol{F}\right\}\boldsymbol{F} \tag{4}$$

$$\pi_{\mathrm{L}}(\boldsymbol{F})=\sigma\left[f\left(\frac{1}{SC}\boldsymbol{F}\right)\right]\boldsymbol{F} \tag{5}$$

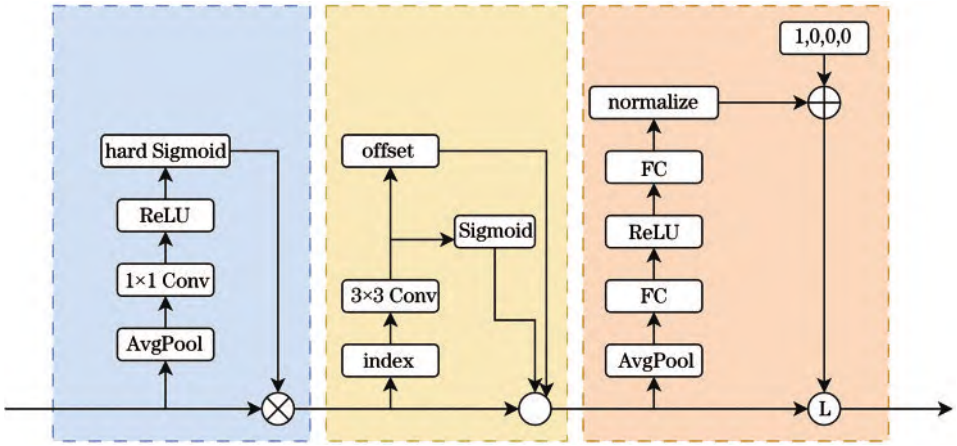


图 6　DyHead 网络结构

Fig. 6　Structure of the DyHead network

$$\pi_S(\boldsymbol{F})\boldsymbol{F}=\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K}\boldsymbol{w}_{l,k}\boldsymbol{F}(l;\boldsymbol{p}_k+\Delta\boldsymbol{p}_k;c)\Delta m_k \tag{6}$$

$$\pi_C(\boldsymbol{F})\boldsymbol{F}=\max\left[\alpha_1(\boldsymbol{F})\boldsymbol{F}_c+\beta_1(\boldsymbol{F}),\alpha_2(\boldsymbol{F})\boldsymbol{F}_c+\beta_2(\boldsymbol{F})\right] \tag{7}$$

式中：$W_L(\boldsymbol{F})$为注意力函数；$\pi_L$为尺度感知注意力；$\pi_S$为空间感知注意力；$\pi_C$为任务感知注意力；$K$为稀疏采样位置数量；$\sigma(x)$为 hard-Sigmoid 激活函数；$f(\cdot)$为线性函数；$L$为特征金字塔中的层数；$S$为特征尺度大小，即特征图长和宽的乘积；$\boldsymbol{w}_{l,k}$为跨级别的特征；$p_k+\Delta p_k$为通过自学习的空间偏移；$\Delta p_k$为歧义区域；$\Delta m_k$为自学习的重要性；$\boldsymbol{F}_c$为第 $c$ 个通道切分的特征；$\alpha_1(\boldsymbol{F})$、$\alpha_2(\boldsymbol{F})$、$\beta_1(\boldsymbol{F})$、$\beta_2(\boldsymbol{F})$为超参数函数，用于学习控制激活阈值。

### 3.5　改进损失函数

损失函数是用于度量模型预测值与实际观测值之间不一致程度的函数。损失应尽可能小，才能使模型的预测接近实际结果。常见的交并比（IoU）损失函数各有特点：广义交并比（GIoU）[25]在 IoU 基础上考虑最小闭包区域，能学习调整预测框的位置；距离交并比（DIoU）[26]不仅考虑预测框和真实框之间的重叠区域，还关注相对位置关系，解决 GIoU 在某些情况下对重叠度变化不敏感的问题；完全交并比（CIoU）[27]考虑边界框之间的重叠区域、中心点距离和纵横比，使其在处理不同形状和大小的物体时更加有效；高效交并比（EIoU）在 CIoU 的基础上增加对宽高比的考量，使模型能更好地处理不同形状的物体。然而这些损失函数难以区分在宽高比相同、具体尺寸不同时的预测框和真实框。为应对以上问题，Ma 等[28]提出 MPDIoU。MPDIoU 基于水平矩形的最小点距离来计算损失，能综合考虑重叠区域、中心点距离、宽度和高度的偏差，为模型提供更精确的损失度量方法。MPDIoU 计算示意图如图 7 所示。

MPDIoU 的计算公式为

$$d_1^2=\left(x_{1,\mathrm{prd}}-x_{1,\mathrm{gt}}\right)^2+\left(y_{1,\mathrm{prd}}-y_{1,\mathrm{gt}}\right)^2 \tag{8}$$

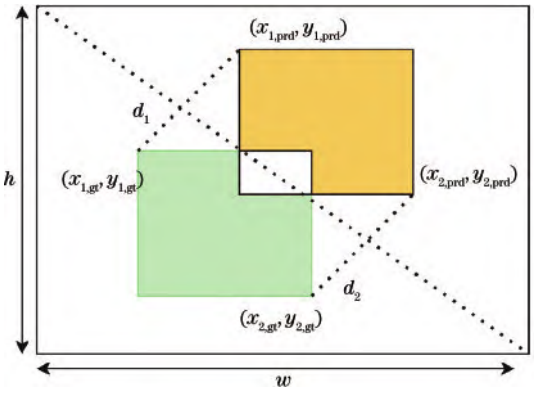


图 7　MPDIoU 计算示意图

Fig. 7　Schematic diagram of the MPDIoU calculation

$$d_2^2=\left(x_{2,\mathrm{prd}}-x_{2,\mathrm{gt}}\right)^2+\left(y_{2,\mathrm{prd}}-y_{2,\mathrm{gt}}\right)^2 \tag{9}$$

$$L_{\mathrm{MPDIoU}}=R_{\mathrm{IoU}}-\frac{d_1^2}{w^2+h^2}-\frac{d_2^2}{w^2+h^2} \tag{10}$$

式中：$(x_{1,\mathrm{prd}},y_{1,\mathrm{prd}})$、$(x_{2,\mathrm{prd}},y_{2,\mathrm{prd}})$分别为预测框的左上角和右下角点的坐标；$(x_{1,\mathrm{gt}},y_{1,\mathrm{gt}})$、$(x_{2,\mathrm{gt}},y_{2,\mathrm{gt}})$分别为真实框的左上角和右下角点的坐标；$d_1$、$d_2$分别为预测框与真实框左上角、右下角坐标点的距离；$R_{\mathrm{IoU}}$为 IoU 的值；$w$为物体的宽；$h$为物体的高。采用 Inner-IoU[29]思想对 MPDIoU 进行改进，Inner-IoU 专注于边界框内部的重叠部分，通过区分不同的回归样本并使用不同尺度的辅助边框来计算损失，可有效加速边框回归过程，显著提高模型的泛化能力。将 Inner-IoU 集成至 MPDIoU 损失函数中，能有效提升模型对目标的检测精度和泛化能力，加快收敛速度。

## 4　实验与结果分析

### 4.1　实验条件

模型训练和性能评价实验均在 GPU 服务器上完成。服务器硬件配置如下：CPU 为 12th Gen Intel(R) Core(TM) i5-12490F，显卡型号为 RTX 3070Ti（8 GB）。软件环境如下：Python 版本为 3.10，Torch 版本为

2.0.1，计算统一设备体系结构（CUDA）版本为11.8。为控制变量，实验设置统一的训练参数，优化器采用标准随机梯度下降（SGD），单次训练样本数量为4，迭代次数为200。

### 4.2　评价指标

为衡量模型的性能，使用IoU阈值为0.5的平均精度均值（mAP50）、IoU阈值为0.5～0.9的平均精度均值（mAP50-90）、平均精度（AP）、参数量（Parameters）、浮点计算量（FLOPs）、模型大小进行评估。通过这些指标可全面地评价模型在特定任务上的表现和效率，其中：参数量反映所需的计算资源和模型的复杂度，参数量越大模型越复杂；FLOPs指执行模型推理时所需的浮点运算次数，反映模型的计算复杂度；模型大小指模型在存储和传输过程中所需的数据量，包括模型的权重、偏置和其他相关参数，直接影响部署成本和模型性能。AP和mAP的计算公式为

$$A_{\mathrm{AP}}=\int_0^1 P(t)\mathrm{d}t \tag{11}$$

$$A_{\mathrm{mAP}}=\frac{1}{N}\sum_{n=1}^{N}A_{\mathrm{AP},n} \tag{12}$$

式中：$P$为精确度；$N$为类别总数。

### 4.3　数据增强效果

为验证ACE图像增强算法的效果，分别使用原始模型在原始URPC数据集和使用ACE增强后的UPRC数据集上进行训练，结果如表2所示。使用ACE进行数据增强后，算法的mAP50增加0.7百分点，mAP50-90增加3.0百分点，检测海胆、海参、扇贝的AP分别增加0.9百分点、0.8百分点、1.7百分点。结果表明，使用ACE进行数据增强的策略可提高图像的整体质量，能有效避免低频背景的干扰，且ACE在不同种类的图像上都能取得良好的增强效果。后续实验皆在使用ACE增强后的URPC2020数据集上进行。

表2　ACE增强效果

Table 2　Enhancement effect by ACE　　unit: %

| Method | AP | | | | mAP50 | mAP50-90 |
|---|---|---|---|---|---|---|
| | Echinus | Starfish | Holothurian | Scallop | | |
| URPC | 69.7 | **90.6** | 82.4 | 63.4 | 76.5 | 39.9 |
| URPC+ACE enhanced | **70.6** | 89.9 | **83.2** | **65.1** | 77.2 | **42.9** |

### 4.4　SOAT对比实验

为验证算法的性能，在URPC2020数据集上，将SCDDI-YOLOv8与YOLOv3-tiny、YOLOv5n、YOLOv7-tiny、YOLOv8n，以及先进的YOLOv9c、RT-DETR模型进行对比，各检测模型的参数对比结果如表3所示。从表3中可以看出，与YOLOv3-tiny、YOLOv7-tiny、YOLOv8n模型相比，SCDDI-YOLOv8模型的精度更高、参数量更小。与YOLOv5n相比，SCDDI-YOLOv8模型的精度略差，但参数量较小。最先进的YOLOv9c和RT-DETR模型的精度较好，但参数量巨大，并不适用于水下资源受限的边缘计算设备。所提SCDDI-YOLOv8算法有效平衡精度和轻量化，在保证轻量化的同时提高模型的识别精度。

表3　不同检测模型参数结果

Table 3　Results of parameters for different detection models

| Model | Parameters /$10^6$ | mAP50 /% | mAP50-90 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|---|
| YOLOv3-tiny | 8.67 | 71.0 | 34.6 | 12.9 | 17.4 |
| YOLOv5n | 7.02 | 79.2 | 44.9 | 15.8 | 14.4 |
| YOLOv7-tiny | 6.02 | 74.9 | 38.4 | 13.2 | 12.3 |
| YOLOv8n | 3.00 | 76.7 | 42.9 | 8.1 | 6.3 |
| YOLOv9c | 25.30 | **79.9** | **45.6** | 102.3 | 51.6 |
| RT-DETR | 15.50 | 79.8 | 44.5 | 39.7 | 31.5 |
| SCDDI-YOLOv8 | **2.38** | 77.3 | 41.4 | **7.5** | **5.1** |

### 4.5　算法的泛化性

为验证改进算法的泛化能力，将所提算法在水下垃圾（UWG）数据集上进行实验，并分别与原始模型、RT-DETR模型进行对比。如表4所示，改进的

表4　不同模型在UWG数据集的检测效果

Table 4　Detection effect of different models on the UWG dataset

| Model | Parameters /$10^6$ | mAP50 /% | mAP50-90 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|---|
| YOLOv8n | 3.00 | 69.7 | 44.3 | 8.1 | 6.3 |
| RT-DETR | 15.59 | 64.5 | 41.6 | 37.6 | 31.5 |
| SCDDI-YOLOv8 | **2.38** | **71.5** | **44.5** | **7.5** | **5.1** |

SCDDI-YOLOv8算法在UWG数据集的性能依旧领先其他算法，说明所提算法具有较好的泛化性。

### 4.6 骨干网络对比

为验证改进骨干网络的性能，对比所提SENetV2与主流轻量化骨干网络，结果如表5所示。与MobileNetV3、EfficientNetV2、RepViT骨干网络相比，SENetV2的mAP50比EfficientNetV2减少3.7百分点，略低于另外两个模型精度，但参数量分别减少～47%、～86%、～27%，模型大小分别减少5.4、38.0、2.4 MB。与ShuffleNetV2、GhostNetV2、EfficientViT2、SwinTransformer骨干网络相比，SENetV2的mAP50分别提高1.2百分点、1.9百分点、1.3百分点、3.2百分点，mAP50-90分别提高1.8百分点、1.9百分点、1.5百分点、3.7百分点，且参数量小于Ghostnetv2、EfficientViT2、SwinTransformer网络，实现高精度与轻量化的统一。与先进模型RT-DETR的骨干网络PPHGNetV2相比，SENetV2的mAP50只降低1.7百分点，但其参数量和计算量仅为PPHGNetV2的25%和～44%，且模型大小减少18.0 MB。SENetV2通过多分支FC层来执行SaE操作并进行特征缩放，提升特征表达的精细度和全局信息的整合能力，在保证计算效率的同时显著提升网络的特征提取和表示能力。

表5 不同骨干网络的性能对比

Table 5 Performance comparison of different backbone networks

| Backbone network | Parameters /$10^6$ | mAP50 /% | mAP50-90 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|---|
| SENetV2 | 3.00 | 76.6 | 43.5 | 12.9 | 6.3 |
| MobileNetV3 | 5.65 | 77.3 | 43.6 | 10.7 | 11.7 |
| ShuffleNetV2 | **2.79** | 75.4 | 41.7 | **7.4** | **5.9** |
| EfficientNetV2 | 21.75 | **80.3** | **45.7** | 55.1 | 44.3 |
| GhostNetV2 | 6.30 | 74.7 | 41.6 | 8.7 | 13.3 |
| EfficientViT2 | 4.01 | 75.3 | 42.0 | 9.5 | 8.7 |
| RepViT | 4.12 | 76.8 | 42.9 | 11.8 | 8.7 |
| SwinTransformer | 29.90 | 73.4 | 39.8 | 402.1 | 60.5 |
| PPHGNetV2 | 12.00 | 78.3 | 44.4 | 29.2 | 24.3 |

### 4.7 颈部网络对比

为验证改进颈部网络的性能，分别对比所提算法采用的CCFM和当前主流颈部网络的性能，结果如表6所示。CCFM的参数量、FLOPs、模型大小分别为$1.96\times10^6$、$6.7\times10^9$、4.2 MB，均小于常见特征融合网络，且CCFM的mAP50与其他特征融合网络相比相差小于2.0百分点。CCFM通过轻量级的设计，使模型在处理不同大小和形状的物体时增强对尺度变化和小尺度对象的适应性，提高目标检测模型的泛化性及适应性。

表6 不同颈部网络的性能对比

Table 6 Performance comparison of different neck networks

| Neck network | Parameters /$10^6$ | mAP50 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|
| CCFM | **1.96** | 75.6 | **6.7** | **4.2** |
| Slim-Neck | 2.79 | 76.3 | 7.4 | 5.9 |
| BiFPN | 2.78 | 77.4 | 8.1 | 5.8 |
| RepGFPN | 3.28 | 75.6 | 8.5 | 6.8 |
| Gold-YOLO | 8.06 | **76.4** | 17.6 | 16.6 |
| ASF-YOLO | 3.05 | 76.3 | 8.6 | 6.4 |

### 4.8 上采样器对比

为验证加入的DySample上采样器的效果，对比不同的采样器，结果如表7所示。相较于CARAFE上采样机制，DySample的参数量减少$1.3\times10^5$，mAP50增加0.4百分点，模型大小减少0.2 MB。DySample作为轻量级的上采样机制，不需要额外定制CUDA包，计算资源需求较小，可实现图像分辨率的提升。

表7 不同采样器的性能对比

Table 7 Performance comparison of different samplers

| Sampler | Parameters /$10^6$ | mAP50 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|
| DySample | **3.01** | **76.7** | 12.9 | **6.3** |
| CARAFE | 3.14 | 76.3 | 8.4 | 6.5 |

### 4.9 检测头对比

为验证加入的DyHead检测头对模型性能的影响，在原始模型中加入不同的检测头进行对比，结果如表8所示。相较于其他检测头，加入DyHead的模型的精度最高、复杂度最低、规模最小，其mAP50达到77.0%，参数量为$2.50\times10^6$，FLOPs为$7.8\times10^9$，模型大小为5.3 MB。相较于其他常规检测头，加入DyHead的模型的性能均有所提升，且优于最新RT-DETR模型的检测头RT-DETRDecoder。

### 4.10 损失函数对比

为验证加入的InnerMPDIoU损失函数对模型精度的影响，在原始模型中加入不同的边界框损失函数并进行对比，结果如表9所示。InnerMPDIoU考虑边界框之间的尺度差异，能更准确地反映预测框与真实框之间的匹配程度，提高模型的定位性能。相较于其他损失函数，InnerMPDIoU的mAP最高。

表 8 不同检测头的性能对比

Table 8 Performance comparison of different detection heads

| Detection head | Parameters /$10^6$ | mAP50 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|
| DyHead | **2.50** | **77.0** | **7.8** | **5.3** |
| Detect_FRM | 87.90 | 75.8 | 76.3 | 176.2 |
| Detect_FASFF | 4.31 | 75.1 | 15.4 | 9.1 |
| Detect_ASFF | 4.37 | 76.1 | 10.4 | 9.0 |
| RT-DETR_Decoder | 9.48 | 71.8 | 16.7 | 19.2 |

表 9 损失函数对比

Table 9 Comparison of loss functions units: %

| Loss function | AP | | | | mAP50 | mAP50-90 |
|---|---|---|---|---|---|---|
| | Echinus | Starfish | Holothurian | Scallop | | |
| InnerMPDIoU | 70.9 | **90.6** | 82.0 | **64.7** | **77.1** | **40.6** |
| MPDIoU | **71.8** | 90.3 | 81.1 | 63.3 | 76.6 | 40.5 |
| InnerCIoU | 71.5 | 90.0 | **82.6** | 63.7 | 77.0 | 40.6 |
| InnerWIoU | 58.9 | 77.6 | 63.2 | 39.9 | 59.9 | 28.3 |
| InnerSIoU | 70.3 | 90.5 | 81.9 | 63.2 | 76.5 | 40.2 |

### 4.11 消融实验

为验证加入的改进模块对整体模型精度的影响，以及模块之间相互耦合对性能的影响，在原始模型 YOLOv8n 上进行 5 个改进模块的消融实验。在 URPC2020 数据集上的实验结果如表 10 所示，其中“√”表示选择该模块。与基线模型 YOLOv8n（方法 a）相比：引入 CCFM 后，参数量减少～34.7%，引入 DyHead 模块后，mAP50 提高 0.3 百分点，参数量减少～16.7%；引入 InnerMPDIoU 损失函数后，mAP50 提高 0.4 百分点。说明以上 3 种改进方法均能促进模型检测性能的提升。为进一步验证模块耦合的促进作用，对添加多种模块的效果进行对比，结果表明，耦合多种模块的模型性能优于添加单个模块的性能，其中包含 SENetV2、CCFM、DyHead、DySample、InnerMPDIoU 模块的模型（方法 j）表现最好，与原始模型相比，方法 j 的 mAP50 提高 0.6 百分点，参数量减少～20.7%，FLOPs 减少 $6\times10^8$，模型大小减小 1.2 MB。这说明：利用 DyHead 能较好地提升尺度变化目标、空间分布目标和多任务目标的检测效果；CCFM 模块在处理多尺度目标和提高小目标检测率方面表现出色，且能实现模型轻量化，减少计算成本；DySample 轻量化上采样器能提高上采样效率；InnerMPDIoU 可提高各类别边界框回归的精度。改进 SCDDI-YOLOv8 模型在 URPC2020 数据集上的整体性能最优，且模型复杂度较低，有利于应用于水下边缘计算设备。同样在 UWG 数据集进行消融实验的结果也验证了 SCDDI-YOLOv8 算法的优越性。

表 10 URPC2020 数据集上的消融实验结果

Table 10 Results of the ablation experiments on the URPC2020 dataset

| Method | SENetV2 | CCFM | DyHead | DySample | InnerMPDIoU | Parameters /$10^6$ | mAP50 /% | mAP50-90 /% | FLOPs /$10^9$ | Model size /MB |
|---|---|---|---|---|---|---|---|---|---|---|
| a | | | | | | 3.00 | 76.7 | 42.9 | 8.1 | 6.3 |
| b | √ | | | | | 3.00 | 76.6 | **43.5** | 12.9 | 6.3 |
| c | | √ | | | | **1.96** | 75.6 | 42.5 | **6.7** | **4.2** |
| d | | | √ | | | 2.50 | 77.0 | 43.4 | 7.8 | 5.3 |
| e | | | | √ | | 3.01 | 76.7 | 43.0 | 8.2 | 6.3 |
| f | | | | | √ | 3.00 | 77.1 | 40.6 | 8.1 | 6.3 |
| g | √ | √ | | | | 2.78 | 76.5 | 43.2 | 7.8 | 5.9 |
| h | √ | √ | √ | | | 2.30 | 77.2 | **43.5** | 7.5 | 5.1 |
| i | √ | √ | √ | √ | | 2.38 | 75.6 | 42.7 | 7.5 | 5.1 |
| j | √ | √ | √ | √ | √ | 2.38 | **77.3** | 41.4 | 7.5 | 5.1 |

### 4.12 验证实验

为进一步验证所提算法的先进性，在 URPC2020 数据集上对目标进行预测，可视化结果如图 8 所示。对比 YOLOv8 和 ACE+YOLOv8 结果可以看出，经过 ACE 增强后，模型检测的目标数量较原始模型明显增加，能识别更多较暗的目标和较小的目标。ACE 增强

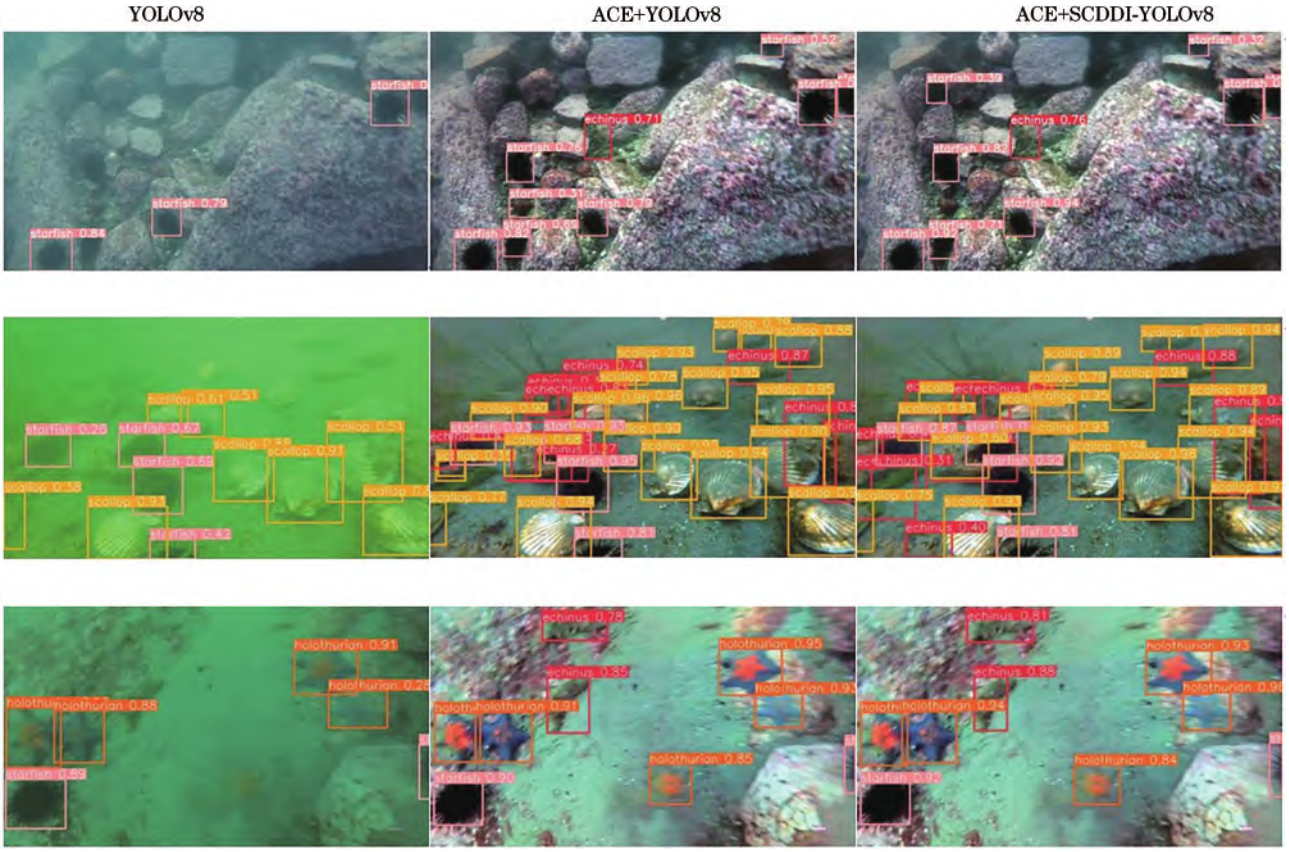


图 8　不同算法在URPC2020数据集上的验证结果

Fig. 8　Validation results on the URPC2020 dataset using different algorithms

算法能够显著增强图像的对比度，有效减少低频背景对图像质量的干扰，使图像的主要特征更加突出，有效改善水下图片质量。对比ACE+YOLOv8和ACE+SCDDI-YOLOv8识别结果可以看出，所提算法识别到的小目标数量明显增多，锚框中小目标识别精度明显提高。对比图中第二行发现，SCDDI-YOLOv8能较好地识别水中大尺寸目标，提高模型对大尺寸目标的检测精度，这说明DyHead的多尺度感知、空间感知、任务感知分别能更好地识别不同尺度的物体、增强模型对物体位置的识别能力、更好地适应不同的检测任务，显著提升模型精度。MPDIoU考虑边界框之间距离的IoU，它不仅关注重叠区域，还关注非重叠区域的分布。而InnerMPDIoU引入新的尺度因子和MPDIoU结合，能更灵活地适应不同大小的物体，提高模型对物体定位的准确性。

## 5　结　　论

针对水下目标检测算法精确率低、算法复杂度高的问题，提出一种基于ACE图像增强和轻量化YOLOv8的水下目标检测算法，并将其命名为SCDDI-YOLOv8。所提算法可降低检测模型的计算复杂度，实现边缘计算系统对水下目标的高精度检测。在URPC2020数据集上，SCDDI-YOLOv8模型的mAP50达到77.3%，模型大小降至5.1 MB，参数量减少至$2.38\times10^{6}$，与原始YOLOv8n模型相比，参数量下降~20.7%，FLOPs减少$6\times10^{8}$。在UWG数据集上SCDDI-YOLOv8模型的检测效果同样明显，表明所提模型具有较好的泛化能力。与其他先进算法进行对比，所提算法提高对水下目标的识别精度，并显著降低参数量，更适合水下计算资源受限的系统，提高对边缘设备的友好性。但该研究停留在水下目标的定位与识别，没有关注水中的动态目标，下一步研究将进行水下目标的识别与追踪。

## 参　考　文　献

[1] Jaffe J S. Computer modeling and the design of optimal underwater imaging systems[J]. IEEE Journal of Oceanic Engineering, 1990, 15(2): 101-111.

[2] 孙传东, 陈良益, 高立民, 等. 水的光学特性及其对水下成像的影响[J]. 应用光学, 2000, 21(4): 39-46.
Sun C D, Chen L Y, Gao L M, et al. Water optical properties and their effect on underwater imaging[J]. Journal of Applied Optics, 2000, 21(4): 39-46.

[3] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 779-788.

[4] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[M]//Leibe B, Matas J, Sebe N, et al. Computer vision-ECCV 2016. Lecture notes in

computer science. Cham: Springer International Publishing, 2016, 9905: 21-37.

[5] Ren S Q, He K M, Girshikck R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[6] Zhao Y A, Lü W Y, Xu S L, et al. DETRs beat YOLOs on real-time object detection[EB/OL]. (2023-04-17)[2024-05-06]. https://arxiv.org/abs/2304.08069v3.

[7] 姜丽梅, 陈信威. 动态场景下基于特征点筛选的视觉SLAM算法[J/OL]. 系统仿真学报: 1-12[2024-04-18]. https://doi.org/10.16182/j.issn1004731x.joss.23-1406.
Jiang L M, Chen X W. Visual SLAM algorithm based on feature point selection in dynamic scenes[J/OL]. Journal of System Simulation: 1-12[2024-04-18]. https://doi.org/10.16182/.joss.23-1406.

[8] 宋绍剑, 朱靖旭. 基于Mask R-CNN和迁移学习的水下生物目标识别研究[J]. 计算机应用研究, 2020, 37(s2): 386-388, 391.
Song S J, Zhu J X. Research on underwater biological target recognition based on mask R-CNN and transfer learning[J]. Application Research of Computers, 2020, 37(s2): 386-388, 391.

[9] 王蓉蓉, 蒋中云. 基于改进CenterNet的水下目标检测算法[J]. 激光与光电子学进展, 2023, 60(2): 0215001.
Wang R R, Jiang Z Y. Underwater object detection algorithm based on improved CenterNet[J]. Laser & Optoelectronics Progress, 2023, 60(2): 0215001.

[10] Zhang M H, Xu S B, Song W, et al. Lightweight underwater object detection based on YOLO v4 and multi-scale attentional feature fusion[J]. Remote Sensing, 2021, 13(22): 4706.

[11] Yu K, Cheng Y F, Tian Z T, et al. High speed and precision underwater biological detection based on the improved YOLOV4-tiny algorithm[J]. Journal of Marine Science and Engineering, 2022, 10(12): 1821.

[12] 赵磊, 矫立宽, 翟冉, 等. 基于YOLOv5的瓶盖封装缺陷轻量化检测算法[J]. 激光与光电子学进展, 2023, 60(22): 2210009.
Zhao L, Jiao L K, Zhai R, et al. YOLOv5-based lightweight algorithm for detecting bottle-cap packaging defects[J]. Laser & Optoelectronics Progress, 2023, 60(22): 2210009.

[13] 吴萌萌, 张泽斌, 宋尧哲, 等. 基于自适应特征增强的小目标检测网络[J]. 激光与光电子学进展, 2023, 60(6): 0610004.
Wu M M, Zhang Z B, Song Y Z, et al. Small-target detection network based on adaptive feature enhancement[J]. Laser & Optoelectronics Progress, 2023, 60(6): 0610004.

[14] Wu G H, Ge Y, Yang Q. UTD-YOLO: underwater trash detection model based on improved YOLOv5[J]. Journal of Electronic Imaging, 2023, 32(6): 063034.

[15] Zhang J, Chen H D, Yan X Y, et al. An improved YOLOv5 underwater detector based on an attention mechanism and multi-branch reparameterization module [J]. Electronics, 2023, 12(12): 2597.

[16] Guo A, Sun K Q, Zhang Z Y. A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection[J]. Journal of Real-Time Image Processing, 2024, 21(2): 49.

[17] Gatta C, Rizzi A, Marini D. ACE: an automatic color equalization algorithm[J]. Conference on Colour in Graphics, Imaging, and Vision, 2002, 1(1): 316-320.

[18] Land E H, McCann J J. Lightness and retinex theory[J]. Journal of the Optical Society of America, 1971, 61(1): 1-11.

[19] He K M, Sun J, Tang X O. Single image haze removal using dark channel prior[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12): 2341-2353.

[20] Rahman Z, Jobson D J, Woodell G A. Multi-scale retinex for color image enhancement[C]//Proceedings of 3rd IEEE International Conference on Image Processing, September 19-19, 1996, Lausanne, Switzerland. New York: IEEE Press, 2002: 1003-1006.

[21] Narayanan M. SENetV2: aggregated dense layer for channelwise and global representations[EB/OL]. (2023-11-17)[2024-05-06]. https://arxiv.org/abs/2311.10807v1.

[22] Wang J Q, Chen K, Xu R, et al. CARAFE: content-aware ReAssembly of FEatures[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV), October 27-November 2, 2019, Seoul, Korea (South). New York: IEEE Press, 2019: 3007-3016.

[23] Liu W Z, Lu H, Fu H T, et al. Learning to upsample by learning to sample[C]//2023 IEEE/CVF International Conference on Computer Vision (ICCV), October 1-6, 2023, Paris, France. New York: IEEE Press, 2023: 6004-6014.

[24] Dai X Y, Chen Y P, Xiao B, et al. Dynamic head: unifying object detection heads with attentions[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 20-25, 2021, Nashville, TN, USA. New York: IEEE Press, 2021: 7369-7378.

[25] Rezatofighi H, Tsoi N, Gwak J, et al. Generalized intersection over union: a metric and a loss for bounding box regression[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 658-666.

[26] Zheng Z H, Wang P, Liu W, et al. Distance-IoU loss: faster and better learning for bounding box regression[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12993-13000.

[27] Zheng Z H, Wang P, Ren D W, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE Transactions on Cybernetics, 2022, 52(8): 8574-8586.

[28] Ma S L, Xu Y, Ma S L, et al. MPDIoU: a loss for efficient and accurate bounding box regression[EB/OL]. (2023-07-14)[2024-05-06]. https://arxiv.org/abs/2307.07662v1.

[29] Zhang H, Xu C, Zhang S J. Inner-IoU: more effective intersection over union loss with auxiliary bounding box [EB/OL]. (2023-11-06)[2024-05-06]. https://arxiv.org/abs/2311.02877v4.