Applied Intelligence (2025) 55:163
https://doi.org/10.1007/s10489-024-05971-4

Multi-scale feature map fusion encoding for underwater object segmentation

Chengxiang Liu[1] · Haoxin Yao[1] · Wenhui Qiu[1] · Hongyuan Cui[1] · Yubin Fang[1] · Anqi Xu[2]

Accepted: 5 October 2024 / Published online: 13 December 2024

Abstract

Underwater object segmentation presents significant challenges due to the degradation of image quality and the complexity of underwater environments. In recent years, deep learning has provided an effective approach for object segmentation. However, DeepLabV3+, as a classical model for general scenes, shows limitations in achieving accurate and real-time segmentation in complex underwater conditions. To address this issue, we propose a DeepLab-FusionNet, an extended version of DeepLabV3+, specifically designed for underwater object segmentation. The model utilizes a multi-resolution parallel branch structure to extract multi-scale information and employs an improved inverted residual structure as the basic feature extraction module in the encoding network. Structural reparameterization technique is introduced to optimize inference speed and memory access costs during the inference stage. Additionally, a module for linking deep and shallow level information is constructed to reduce the loss of detail and spatial information during downsampling and convolution. Evaluation on the SUIM dataset shows a 3.3% increase in mean Intersection over Union (mIoU) and a speed improvement of 34 frames per second (FPS) compared to the baseline model DeepLabV3+. Further comparisons with other classic lightweight models and Transformer-based models on the UIIS and TrashCan datasets demonstrate that our model achieves good accuracy and balanced computational efficiency in challenging underwater environments. Although there is room for improvement due to overfitting and fixed convolution kernel limitations, future integration with Transformer methods is planned. Our model offers an effective solution for real-time target segmentation for underwater robots, with broad applications in human exploration and development of marine resources. Our codes are available at: https://github.com/sunmer1rain/deeplabv_fusionnet

Keywords Underwater object segmentation · Convolutional neural network · Hybrid encoding · Multi-scale feature fusion

✉ Anqi Xu
xuanqi@szu.edu.cn

Chengxiang Liu
chxliu@szu.edu.cn

Haoxin Yao
2210295006@email.szu.edu.cn

Wenhui Qiu
1059713136@qq.com

Hongyuan Cui
2210295009@email.szu.edu.cn

Yubin Fang
2210295058@email.szu.edu.cn

[1] College of Mechatronics and Control Engineering, Shenzhen University, 518060 Shenzhen, China

[2] College of Physics and Optoelectronic Engineering, Shenzhen University, 518060 Shenzhen, China

1 Introduction

In recent years, the rapid iteration and development of autonomous underwater robots, such as Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs), has provided a new approach to human exploration and development of marine resources [1–3]. In underwater operational tasks, vision-guided robots heavily rely on the data from close-range sensors, such as sonar and optical images, for perceiving and understanding the surrounding underwater environment. Therefore, the perception system of these robots requires high accuracy and real-time performance to complete complex tasks effectively [4]. However, underwater segmentation has not yet been widely adopted on AUVs and ROVs because implementing segmentation algorithms that are both accurate and capable of real-time inference on embedded platforms remains challenging [5].

At present, deep learning is the most effective approach for visual perception tasks [6]. Visual perception techniques based on semantic segmentation align with the way biological organisms perceive the environment and have shown high performance in many fields [7–9]. Semantic segmentation methods for general scenes, such as FCN [10],U-Net [11], and DeepLab series [12–15],can provide complete predictions for high-resolution images and accurately separate different objects from backgrounds. However, the visual information of underwater targets is not obvious due to the complex lighting conditions in underwater environments, which are affected by factors such as water depth, turbidity, and suspended particles. At the same time, reefs [16] and dense shifting seagrass [17] constitute a complex underwater background, which interferes with the segmentation of underwater targets. Furthermore, due to the scattering and refraction of light in the underwater environment, the target objects in the images captured by underwater robots appear blurred and distorted [18], making the accurate segmentation of small targets very difficult.These situations lead to general semantic segmentation models failing to achieve satisfactory results in underwater scenes [19].

Based on the above investigation, we have identified the DeepLabV3+ model, a classic approach in the field of general semantic segmentation, as the basic framework for underwater object segmentation model, called DeepLab-FusionNet. Our objective is to enhance and optimize its architecture for addressing the challenges and limitations in underwater object segmentation.

The main contributions of this paper are summarized as follows:

(1) Tailored adoptation of DeepLabV3+: DeepLabV3+ method is applied to underwater object segmentation and optimized to better adapt to underwater scenes, and optimizations are designed by considering the accuracy and the computing speed. The proposed model provides an effective approach for visual environment perception of underwater robots.
(2) Multi-Scale Feature Map Fusion: We propose a multi-scale feature map fusion module based on HRNet, which preserves detailed features and high-level semantic information across different scales. The inclusion of the Atrous Spatial Pyramid Pooling (ASPP) module enhances the model's ability to segment objects of various sizes by extracting multi-scale receptive field information. Additionally, our deep and shallow level information association structure improves the accuracy of the predicted mask images.
(3) Real-Time Efficiency with Structural Innovation: To meet the real-time requirement of underwater robots, we utilize an inverted residual structure with depth-wise separable convolution in the multi-scale feature map fusion module, which reduces the parameter and computational complexity of the model. Additionally, we apply structural reparameterization technique to convert dense multi-branch structures into single-branch structures during inference, significantly boosting speed without sacrificing accuracy.

In explaining the proposed DeepLab-FusionNet model, this paper is organized as follows: the related work is presented in Section 2. In Section 3, the detail of each method of the proposed model is described. The experimental results of qualitative and quantitative assessments of the proposed model in comparison with other segmentation method are presented in Section 4. Finally, the paper conclusion and future work directions are given in Section 5.

2 Related work

2.1 Underwater object segmentation

Given the good performance of convolutional networks, more and more studies employ them to improve the segmentation result in marine scenes. In 2018, Martin-Abadal et al. proposed a semantic segmentation method for seagrass based on a deep neural network, achieving high-precision segmentation of seagrass [20]. In 2020, Islam et al. proposed a large dataset for semantic segmentation of underwater images and a fully convolutional encoder-decoder model SUIM-Net [21], which balances accuracy and computational efficiency for visual perception tasks in autonomous pipeline operations for underwater robots. In 2021, Nezla et al. proposed an underwater target segmentation based on the U-Net network by employing the encoder-decoder structure [22], achieving an mIoU of 85.8 on the Fish4knowledge image dataset.

However, due to the degradation of underwater image quality and the complex underwater environment, current image segmentation models have great limitations when it comes to recognition and segmentation in underwater scenes [23], which is embodied in the following problems: first, the segmentation accuracy of the model for underwater object details and edge regions remains to be improved, which is the key for distinguishing targets. Second, existing models exhibit slow prediction speeds and high computational overhead during model execution, making it challenging to meet the real-time requirements for underwater environment perception tasks. The limitations mentioned above make it significantly restrict widespread application of semantic models in underwater object segmentation.

Our method is designed to address the specific challenges faced by underwater image segmentation. By tailoring the DeepLabV3+ architecture to better suit underwater environ-

ments and integrating multi-scale feature map fusion with advanced modules such as HRNet and ASPP, we enhance the model's ability to accurately capture fine details and edges of underwater objects. Furthermore, with structural innovations that include an inverted residual structure and reparameterization techniques, we ensure real-time efficiency without sacrificing segmentation accuracy.

2.2 DeeplabV3+ method

As a classic algorithm, DeepLabV3+ has shown exceptional performance and innovative techniques in the field of semantic segmentation [15]. It has demonstrated the effectiveness of the encoder-decoder structure and spatial pyramid pooling module constructed used in this algorithm for various general image segmentation tasks.

However, the experimental result, which will be presented in Section 4, shows that the DeepLabV3+ model does not achieve satisfactory underwater object segmentation performance after being retrained by an underwater dataset. It still suffers from a significant loss of detailed information in the results of image segmentation. The model exhibits weak segmentation performance for small and medium-sized objects as well as overlapping occlusions, and it lacks a strong correlation between semantic information from deep and shallow layers. Underwater scenes primarily consist of small to medium-sized objects, and the visibility in underwater environments is low with complex spatial relationships between objects. Therefore, simply applying DeepLabV3+ to marine object segmentation tasks is not feasible. Additionally, due to the large number of parameters in the DeepLabV3+ model, the inference speed cannot meet real-time requirements.

Our method addresses the shortcomings of DeepLabV3+ in underwater segmentation by enhancing detail preservation and improving the model's ability to discern small to medium-sized objects and complex occlusions. We've also optimized the model to reduce computational load, ensuring real-time performance essential for underwater applications.

3 Proposed method

To address the limitations of DeepLabV3+, this study proposes an extended model called DeepLab-FusionNet, specifically designed for underwater tasks. Considering the diverse sizes of marine objects and the challenging environmental characteristics of underwater scenes, the proposed model aims to enhance and optimize both the accuracy and speed of object segmentation, building upon the structure and design principles of DeepLabV3+.

The DeepLab-FusionNet segmentation model consists of an encoder network, a feature information transfer and association module and a decoder network, as shown in Fig. 1. The encoder network encodes the input image features through a preprocessing module, a multi-scale feature fusion backbone network (Stage1-Stage4), and an Atrous Spatial Pyramid Pooling module (ASPP Module). The feature information transfer and association module is responsible for concatenating the feature information from each stage of the encoder network along the depth channel to achieve multi-scale information fusion and transmits it to the decoder network. The decoder network decodes the fused features at multiple levels using a progressive upsampling approach, and then restores the resolution of the feature maps to the input size to obtain the final prediction mask.

3.1 Encoder network

In underwater perception tasks, the varying distances between target objects and the sampling device result in targets appearing at different sizes in the sampled images. In order to efficiently extract features and recognize various sizes of targets in the input images, the encoder network of the proposed model adopts the multi-resolution feature map upsampling and downsampling fusion approach used in the HRNet method [24, 25]. This method maintains high resolution throughout the process, with feature maps decomposed into multiple-resolution subnetworks in parallel through sampling. Finally, these subnetworks are connected in parallel for the exchange and fusion of feature information from different resolutions. Based on the multi-resolution feature map fusion pattern, each subnetwork contains rich multi-scale features and semantic information about the target during the process of feature processing and transmission.

Considering the real-time requirements for underwater perception tasks, this paper proposes a multi-scale feature map fusion module based on HRNet for underwater object segmentation. This module has been redesigned in terms of convolutional processing and network depth, incorporating the inverted bottleneck module (Inverted BottleNeck) proposed in the MobileNet series [26, 27] as the basic convolutional processing module for the multi-scale feature map fusion module. This aims to improve the efficiency of feature extraction and fusion across multiple scales. After adjusting the number of channels and applying upsampling and downsampling to the feature maps from different branches, they are added pointwise to achieve multi-scale fusion. Figure 2 illustrates the multi-scale feature map fusion structure used in the DeepLab-FusionNet model.

Semantic segmentation models like FCN, SegNet, and DeepLabV3+ often suffer from significant loss of features and details for small and medium-sized objects due to large downsampling rates during feature extraction. In the proposed DeepLab-FusionNet model, the encoding network based on parallel multi-branch processing structure effectively preserves the detailed features and high-level semantic

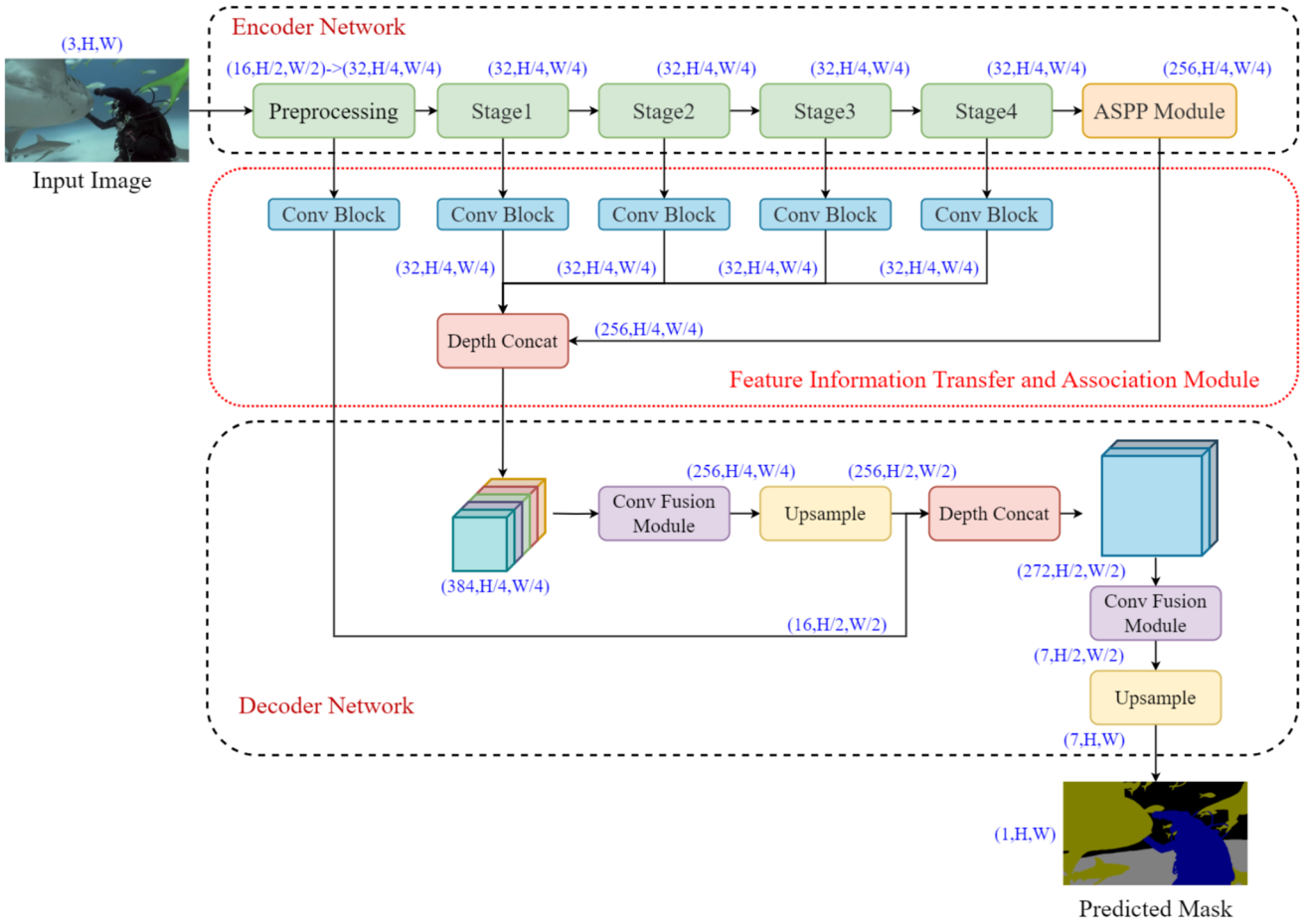

Fig. 1 Structure diagram of the DeepLab-FusionNet model

information of objects at different scales and couples the receptive field information of multiple scales in the same stage, which contributes to more accurate prediction and segmentation of the target region by the decoding network of the model.

3.2 Improved inverted residual convolution

To reduce parameter and computational complexity, the DeepLab-FusionNet model utilizes an inverted residual structure based on depth-wise separable convolution as the

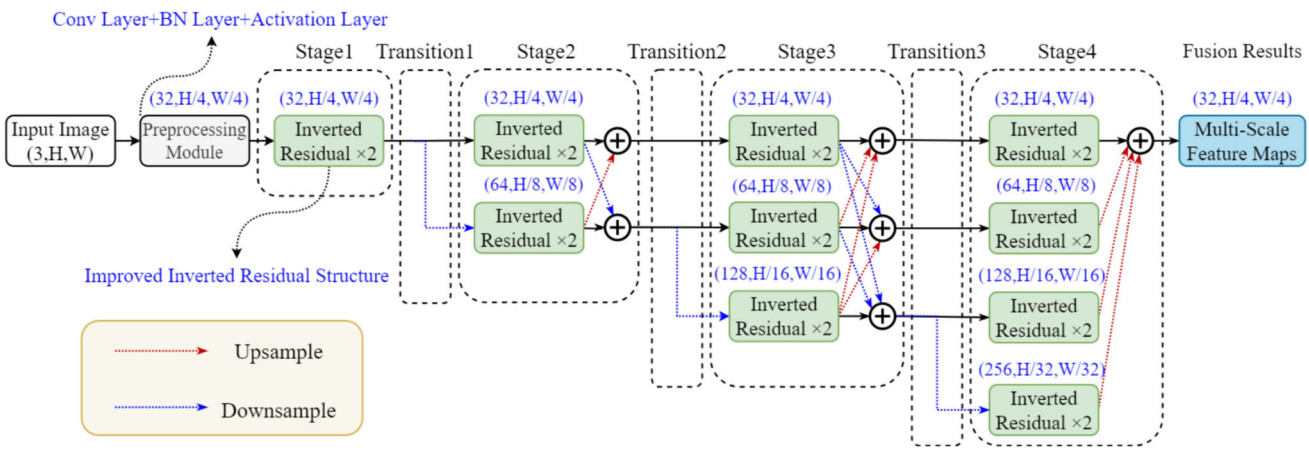

Fig. 2 Multi-scale feature fusion structure

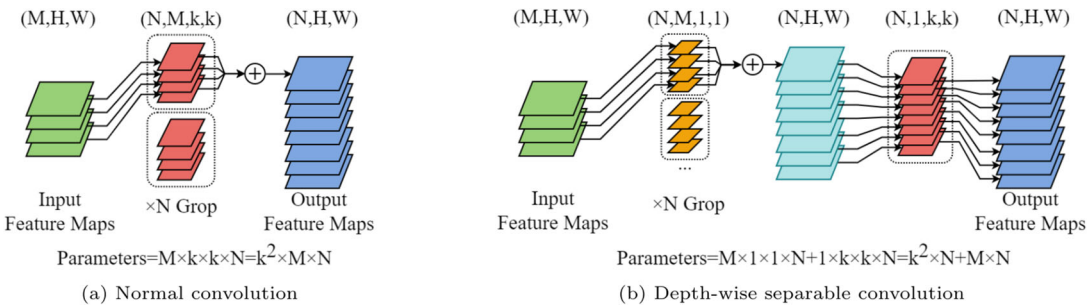

Fig. 3 Comparison of processing procedures between conventional convolution and depth-wise separable convolution

basic convolutional processing module in the multi-scale feature map fusion module. This module separates spatial and channel correlations in the feature map, allowing the neural network to perceive spatial information and extract semantic details. The comparison between regular convolution and depth-wise separable convolution is shown in Fig. 3.

The ratio of the parameter between depth-wise separable convolution and conventional convolution is calculated as:

$$ratio = \frac{M \times 1 \times 1 \times N + 1 \times k \times k \times N}{M \times k \times k \times N} = \frac{1}{k^2} + \frac{1}{M} \quad (1)$$

where M is the number of output channels, N is the number of input channels, and k is the kernel size. The $ratio$ indicates that the depth-wise separable convolution is much lighter than the normal convolution. In the DeepLab-FusionNet model, the four parallel branches have channel numbers of 32, 64, 128, and 256, respectively. Table 1 compares the lightweight evaluation indices before and after applying the depth-wise separable convolution technique.

Table 1 shows that the introduction of depth-wise separable convolution significantly improves the inference efficiency of the object segmentation model in underwater scenarios, facilitating its deployment on embedded devices. In addition to meeting real-time requirements, we also focus on efficiently coupling multiple receptive field features within the same stage of the object segmentation model. An improved structure with parallel convolution branches is designed in the inverted residual structure, including 3 × 3 and 1 × 1 convolution branches, as well as a linear mapping branch to capture and fuse different receptive field information. Figure 4 illustrates the improved inverted residual module structure used in DeepLab-FusionNet.

3.3 Structural reparameterization in validation mode

Existing research shows that multiple-branch structures can enhance the representational capacity of models. Models such as the Inception series [28–31], Xception [32], and DenseNet [33] use dense connections with multiple branches to enhance feature extraction and representation capabilities. However, dense multi-branch connections can lead to substantial memory and computational overhead. To address these issues, DeepLab-FusionNet incorporates the structural reparameterization technique from the RepVGG model [34]. The technique aims to enhance feature extraction during training with multi-branch convolution, and then convert the model into a single-branch structure during inference to accelerate prediction speed, leveraging the homogeneity and additivity of convolutional operations.

By utilizing the structural reparameterization technique, we incorporate multi-branch structures into the inverted residual framework, which aids in coupling multi-scale receptive field information during training and enhances the model's feature representation ability. In addition, switching to a single-branch structure during the inference reduces computational complexity and memory usage, thereby improving inference speed without sacrificing the segmentation accuracy.

Table 1 Comparison of lightweight evaluation index before and after introducing depth-wise separable convolution

Method	Parameters (M)	Floating-Point Operations (GFLOPS)	Predicted Frame Rate (FPS)	Weight File Size (MB)
Improved Pre-Mode	23.4	214.2	53.2	102.6
Improved Post-Model	**4.7**	**156.0**	**59.2**	**19.6**

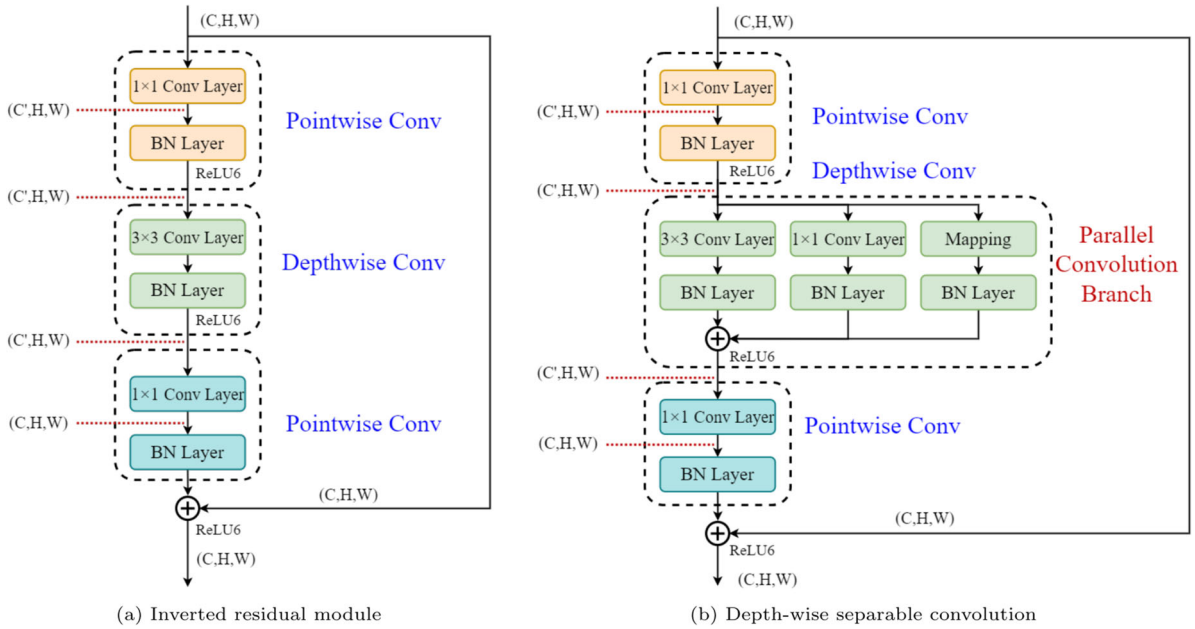

Fig. 4 Structural comparison between the original inverted residual module and the improved inverted residual module

The conversion from a multi-branch structure to a single-branch structure can be described in the following steps:

(1) The convolutional layers and batch normalization layers in the single-branch structure can be represented as follows:

$$Conv(x) = W * x \tag{2}$$

$$BN(x) = \gamma \times \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{3}$$

where x is the input feature map, W is the weight parameters of the convolutional layer, $*$ is the convolution operation, μ and σ represent the accumulated mean and variance of the batch normalization layer, γ and β represent the scalar factors and biases obtained from training the batch normalization layer, and ϵ is a very small but non-zero value. In the single-branch structure, the output y after convolutional and batch normalization computations is calculated as follows:

$$y = BN(Conv(x)) = \gamma \times \frac{W * x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{4}$$

Equation (4) can be rewritten as:

$$y = \frac{\gamma \times W}{\sqrt{\sigma^2 + \epsilon}} * x + \left(\beta - \frac{\gamma \times \mu}{\sqrt{\sigma^2 + \epsilon}}\right) \tag{5}$$

(2) Assume that W_{fused} and b_{fused} represent the weight and bias of the new fused convolutional layer. Therefore, we have:

$$W_{fused} = \frac{\gamma \times W}{\sqrt{\sigma^2 + \epsilon}} \tag{6}$$

$$b_{fused} = \beta - \frac{\gamma \times \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{7}$$

The final formulation of the processing of the input by the fused convolutional layer in the single-branch structure can be expressed as follows:

$$y = W_{fused} * x + b_{fused} \tag{8}$$

(3) The fusion process of the convolutional layer and batch normalization layer in the single-branch structure is achieved based on the homogeneity and additivity of convolutional operations. Before merging the three parallel branches, the convolutional kernels must be padded to the same size. Padding and zero-filling transform a 1×1 convolutional layer into a 3×3 convolutional layer. A 3×3 convolutional layer with identity mapping is added to the linear mapping branch. Finally, the parameters of the three branches are summed up. The equivalent process of merging the parallel branches can be represented as follows:

$$y_{output} = (M * W_3' + b_3')$$

$$
\begin{aligned}
&+ (M * W_1' + b_1') \\
&+ (M * W_0' + b_0') \\
&= M * (W_3' + W_1' + W_0') \\
&+ (b_3' + b_1' + b_0') \qquad (9)
\end{aligned}
$$

$$W_{new} = W_3' + W_1' + W_0' \qquad (10)$$

$$b_{new} = b_3' + b_1' + b_0' \qquad (11)$$

where M is the input feature matrix, y_{output} is the output of the model after structural reparameterization. W_3',W_1' and W_0' denote the fusion convolutional kernel weights of the three parallel branches, while b_3',b_1' and b_0' represent the fusion convolutional kernel biases of the three parallel branches. W_{new} and b_{new} are the equivalent weights and biases of the new convolutional kernel after switching from parallel branches to a single-branch structure.

The improved inverted residual module of the DeepLab-FusionNet method with introducing structural reparameterization technique during training and validation modes is shown in Fig. 5.

3.4 Atrous convolutional pooling pyramid structure

Due to light refraction and the underwater perspective effect, the proportions of underwater targets are prone to distortion. As the underwater robot moves, objects in underwater scenes exhibit a more pronounced characteristic of dynamically appearing in various sizes in the image than general image segmentation scenarios. To improve the segmentation capability of multi-size objects in the images, in this paper, we propose an approach that combines the DeepLab-FusionNet object segmentation method with the Atrous Spatial Pyramid Pooling (ASPP) module. The ASPP module incorporates parallel convolution branches with different atrous rates to efficiently extract multi-scale receptive field information from the feature maps. At the end of the module, a convolutional fusion process is employed to merge the rich multi-scale feature information.

The ASPP module used in the DeepLab-FusionNet method consists of five parallel branches: one 1 × 1 convolutional branch, three 3 × 3 convolutional branches, and one adaptive global average pooling branch. The outputs from the multi-scale feature fusion network in the DeepLab-FusionNet method have a resolution that is 1/4 of the original image due to downsampling. Therefore, the three 3 × 3 convolutional branches need to use convolutional layers with atrous rates of 24, 48, and 72, respectively, to capture a larger range of receptive field areas. The 1 × 1 convolutional branch serves as a skip connection, adjusting the channel number of the input feature map to 256. After batch normalization and non-linear activation, the outputs of 3 × 3 and 1 × 1 convolutional branches are passed to the fusion module. The adaptive global average pooling branch adds global contextual information to the ASPP module. This branch first performs global average pooling on the inputs, reducing the height and width of each channel dimension to 1. Then, after 1 × 1 convolution to adjust the channel number, batch normalization, and non-linear activation, bilinear interpolation upsampling

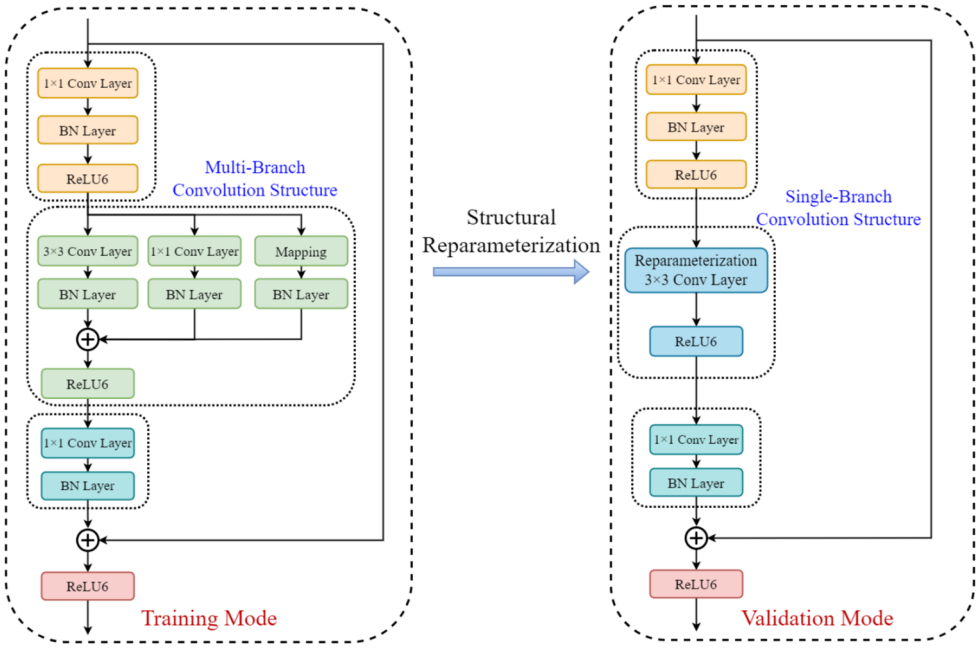

Fig. 5 Training mode structure (left) and validation mode structure (right) of the improved inverted residual module

is performed to obtain a feature map with global semantic information, which is also passed to the fusion module.

In the feature fusion module, feature maps obtained from five branches are concatenated along the depth dimension. A 1×1 convolutional layer is then applied to fuse the channel information, adjusting the number of channels to 256, thus completing the multi-scale feature fusion. The resulting feature map is finally passed to the decoding network. The structure of ASPP module used in the DeepLab-FusionNet method is shown in Fig. 6.

3.5 Deep and shallow level information association structure

In semantic segmentation models, it is necessary to increase the depth of convolutional networks and enlarge the downsampling rate to get high-level semantic information, which helps the model recognize the main parts of large-sized objects in the decoding network. However, during the process of extracting image features and semantic information, it is inevitable that some details and spatial information will be lost, which affects the segmentation accuracy of object details and edge regions. Although the number of pixels occupied by object edges in an image is small, segmenting the edge regions is crucial for distinguishing overlapping, occluded, and densely packed objects in semantic segmentation tasks.

The DeepLab-FusionNet method enhances the association between deep and shallow-level detail positional information and multi-level semantic feature information in the network. It achieves this by transferring the feature information from each Stage module of the encoding network to the decoding network for fusion. Through convolutional processing, it couples features from different receptive fields, strengthening the association of semantic information at different levels. This further optimizes the decoding network for the restoration and segmentation of the main object and edge details, resulting in more accurate and refined predicted mask images. The transfer and association structure of deep and shallow level feature information in the DeepLab-FusionNet method is shown in Fig. 1.

4 Experimental results

4.1 Dataset

We evaluated our model on SUIM dataset [21] (Semantic Segmentation of Underwater Imagery), a publicly available underwater environment dataset, which includes images captured in various complex underwater scenes that correspond to the practical operating environment of underwater robots. During the analysis of the dataset, we found some annotation issues in certain images, which we corrected by re-annotating them. Additionally, due to the small proportion of aquatic plants in the dataset and their similarity in texture and morphology to coral reefs, the aquatic plants category was merged with the coral reefs category. Therefore, the adjusted SUIM dataset consists of seven major object categories: Background Waterbody (BW), Human Divers (HD), Aquatic Plants (AP), Wrecks and Ruins (WR), Robots (RO), Reefs and Invertebrates (RI), Fish and Vertebrates (FV), and Sea-floor and Rocks (SR). The number of images for each category and the percentage of pixels occupied by each category in all images are shown in Table 2. The dataset's object category images and segmentation annotations are shown in Fig. 7.

Apart from the SUIM dataset, our model was compared with other recent lightweight models and transformer-based models on the UIIS [35] and TrashCan [36] datasets. The UIIS dataset comprised eight classes of underwater targets, whereas the TrashCan dataset included seventeen classes of underwater targets.

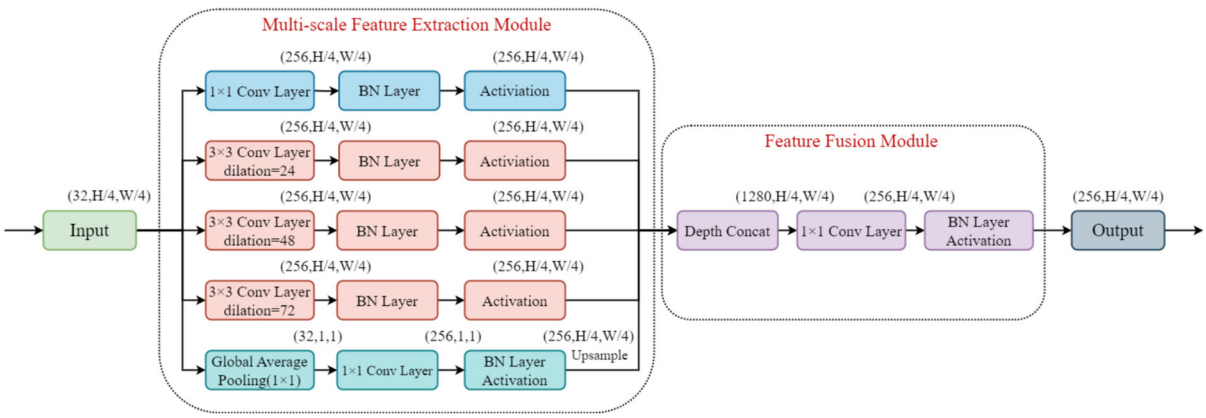

Fig. 6 Structure diagram of the ASPP module used in the DeepLab-FusionNet method

Table 2 Number of images and pixel ratios for all categories in the adjusted SUIM dataset

Dataset type	Comparison content	BW	HD	WR	RO	RI	FV	SR
Training	Number of images	1258	384	262	106	1062	1015	614
dataset	Pixel ratio	32.2%	2.35%	6.73%	0.77%	36.46%	7.15%	14.33%
Testing	Number of images	95	42	27	12	68	67	60
dataset	Pixel ratio	41.8%	3.69%	7.5%	0.9%	22.3%	6.82%	16.99%

4.2 Experimental environmental parameters

All the methods proposed in this study were implemented and built using the PyTorch machine learning framework, specifically version 1.11.0. To ensure a fair comparison among different models, a uniform input image resolution of 512 × 512 was set for both training and validation modes. To enhance the robustness of the models and increase the diversity of the data, all experimental methods employed the same image augmentation techniques, including random scaling, random flipping, Gaussian blur, random rotation, and color space transformation. The batch size for all methods was set to 16, ensuring no GPU memory overflow. The initial learning rate was set to 0.0005, with a training warm-up and cosine annealing strategy used to adjust the learning rate. The Adam optimizer was used for gradient descent, with a total of 500 training iterations performed. Before training, all experimental models were randomly initialized using the parameter initialization method provided by the PyTorch framework, without any additional pre-training or loading of pre-trained model weights. The hardware configuration of the experimental platform used in this study is shown in Table 3.

4.3 Experimental results and analysis

This paper compares different object segmentation models from two aspects: visualization analysis of experimental results and analysis of performance metrics. Visualization analysis compares the output images of different models and assesses the segmentation accuracy based on actual results. The performance metrics analysis evaluates the segmentation performance of the object segmentation models by comparing metrics such as mean Intersection over Union (mIoU), mean Precision (mPrecision), mean Recall (mRecall), Accuracy, and Average Frame Rate.

mIoU is calculated as the ratio of the intersection of the predicted and actual segmentation areas to their union, averaged across all classes. mIoU provides a robust measure of the model's overall accuracy in correctly segmenting different regions, making it a standard for comparing segmentation models. mPrecision calculates the ratio of correctly predicted positive pixels to the total predicted positive pixels across all classes. It reflects the model's ability to minimize false positives, ensuring that when the model predicts a positive class, it is likely to be correct. mRecall measures the ratio of correctly predicted positive pixels to the total actual positive pixels across all classes. It indicates the model's capability to capture as many true positives as possible, thereby minimizing false negatives. Accuracy measures the proportion of correctly classified pixels over the total number of pixels, offering a general assessment of the model's performance across all classes.

Finally, ablation experiments on SUIM dataset are conducted to validate the effectiveness of the different modules proposed in this paper.

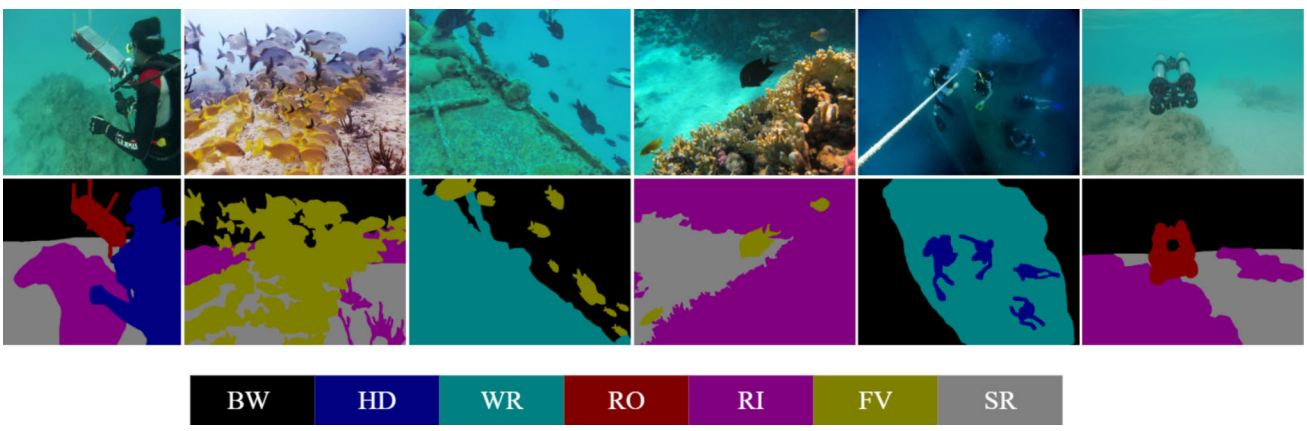

Fig. 7 Target class images and segmentation annotations of the SUIM Dataset

Table 3 Hardware configuration of experimental devices

Component name	Parameters
CPU	Intel-i9 10980XE
RAM	64GB
HDD	6TB
GPU	NVIDIA RTX3090(24GB)
OS	Ubuntu 21.10

4.3.1 Visualization analysis

The visualization results on the SUIM dataset of the proposed DeepLab-FusionNet model are compared with some classic segmentation models, such as: FCN-8S, U-Net, ASPP, HRNet, DeepLabV3+, and DeepLabV3+(HRNet), as shown in Figs. 8 and 9.

The proposed model achieved better segmentation accuracy compared to the baseline model on different-sized objects and edge details. For example, in the first row of the visualization comparison images (Figs. 8 and 9), our model effectively recognizes the cable parts of the underwater robot, while other segmentation models do not. Additionally, our model has better performance in recognizing and segmenting overlapping and occluded objects. As is shown in the second and the tenth rows, the DeepLab-FusionNet can accurately segment the overlapping divers, robots, sunken ships, and fish. Moreover, it also has a better understanding of complex environments, such as shown in the seventh and ninth rows, the DeepLab-FusionNet model successfully identifies and segments the significant target diver in the main part of the shipwreck debris while other experimental models mistakenly identify the diver as part of the shipwreck.

Table 4 presents the IoU comparison for each class in the SUIM dataset between our model and classic models. Bold and underlined fonts indicate the best and second-best results for IoU in each target class.

As can be seen from Table 4, our method achieved the best and second-best segmentation accuracy for small and medium-sized objects (underwater robot RO, vertebrate animal FV, and diver HD), and the second-best segmentation accuracy for large-sized objects (shipwreck WR). It also achieved good results for large-area distributed static objects (water background BW, coral reef RI, and seabed rock SR). Experimental results demonstrate that the multi-scale feature map fusion is highly beneficial for segmenting underwater targets of various sizes, particularly small to medium-sized targets.

Other models like HRNet excel in high-resolution feature extraction, making them perform better in segmenting complex and detail-rich targets, while LRASPP, due to its lightweight design, performs well in handling large homogeneous areas. Nevertheless, our method stands out in adapting to different shapes and sizes of objects, balancing computational efficiency and segmentation accuracy, and maintaining robustness and consistency across various underwater environments, making it particularly suitable for practical applications.

4.3.2 Performance metrics analysis of experimental results

(1) Analysis of Performance on SUIM Dataset Compared to Classic Segmentation Models

The prediction results of our model and the classic models on the SUIM dataset were separately calculated, as shown in Table 5.

From the results in Table 5, it can be seen that the DeepLab-FusionNet model outperforms other comparative classic models in terms of segmentation accuracy metrics, including mIoU, mPrecision, and mRecall. Under the condition of input image resolution of 512×512, the DeepLab-FusionNet model achieves mIoU, mPrecision, mRecall, and Accuracy of 71.8%, 84.9%, 82.3%, and 85.9%, respectively. Compared to the baseline model DeepLabV3+(Xception), it shows an improvement of 3.3% in mIoU, 3.1% in mPrecision, 1.6% in mRecall, and 1.2% in Accuracy. Compared to the HRNet segmentation model, it demonstrates an improvement of 1.6% in mIoU, 1.9% in mPrecision, 0.7% in mRecall, and 0.7% in Accuracy.

As for the prediction speed, the DeepLab-FusionNet model achieves an average frame rate of 59.2 FPS, which meets the frame rate range (30 FPS to 60 FPS) of underwater image capture devices in practical applications. Compared to the baseline model DeepLabV3+ and the HRNet segmentation model, the DeepLab-FusionNet model achieves average frame rate improvements of 34.0 FPS and 29.9 FPS, respectively. These results confirm that the tailored optimizations applied to the DeepLabV3+ architecture, as introduced in the DeepLab-FusionNet, effectively enhance both segmentation accuracy and inference speed.

The Accuracy metric measures the proportion of correctly predicted pixels to the total number of pixels, making it a straightforward and intuitive measure. However, in cases of class imbalance, such as when the background occupies a large portion of the image, the accuracy can be high even if the model performs poorly on small object segmentation. In the testing dataset of SUIM, Background Waterbody (BW), Reefs and Invertebrates (RI) and Sea-floor and Rocks (SR) together account for more than 80% of the pixels, both representing large objects. This could lead to models that excel at segmenting large objects performing better on the Accuracy metric.

The LRASPP network, compared to the network proposed in this paper, is more lightweight and simpler. A simplified network is typically easier to train and less prone to over-

Fig. 8 Visualization comparison of classic segmentation models on SUIM dataset

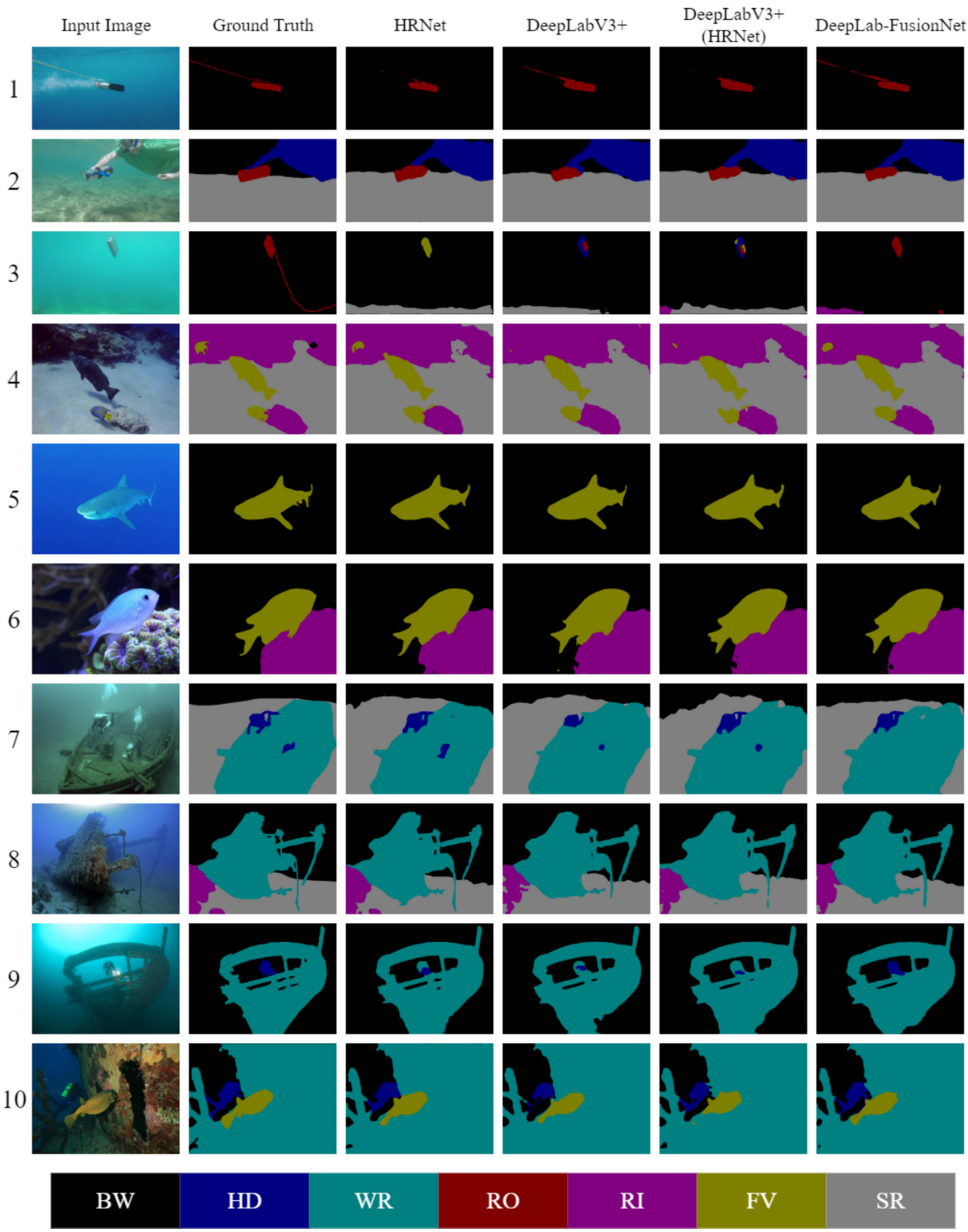

Fig. 9 Visualization comparison of classic segmentation models on SUIM dataset

Table 4 Comparison of individual class object IoU with classic models on SUIM dataset

Model	Backbone	BW	HD	WR	RO	RI	FV	SR
FCN-8S [10]	ResNet	86.3%	71.9%	53.2%	68.5%	70.2%	58.6%	58.8%
U-Net [11]	VGG	87.4%	67.7%	44.8%	58.2%	67.2%	56.5%	61.3%
LRASPP [27]	MobileNetV3	87.2%	74.1%	**65.6%**	68.6%	**72.2%**	64.9%	**63.3%**
HRNet [24]	HRNet w32	**87.9%**	**78.6%**	63.3%	70.1%	68.7%	63.7%	59.0%
DeepLabV3 [15]	Xception	85.5%	75.2%	58.2%	67.3%	71.5%	63.9%	58.1%
DeepLabV3+ [15]	HRNet w32	87.7%	75.9%	52.3%	70.8%	69.3%	62.9%	60.0%
DeepLab-FusionNet	Ours	87.4%	76.7%	64.1%	**75.9%**	71.4%	**66.3%**	60.8%

fitting. For large objects, the network needs to capture the overall shape and contour rather than focusing on too many detailed features. A simplified network can concentrate more on the main features of large objects, thereby improving segmentation performance. This is the reason why, in Table 5, LRASPP slightly outperforms our model on the Accuracy metric.

(2) Analysis of Performance on Three Datasets Compared to Lightweight Models and Transformer-based Models

In Table 6, we present the performance comparison of our model with several other models. These include the lightweight models BiSeNet [37] and PP-LiteSeg [38], as well as the transformer-based models Segmenter [39], SegFormer [40], and TopFormer-B [41]. The comparisons were conducted on the SUIM, UIIS, and TrashCan datasets. The best and second-best results on the UIIS dataset are highlighted in bold red and underlined red, respectively. The results on the TrashCan dataset are highlighted in blue.

As shown in Table 6, the speed of our model ranked third among the compared models, following only BiSeNet and Segmenter. In the prediction results on the three datasets, our model outperformed BiSeNet in all metrics except for Accuracy. BiSeNet, as the fastest lightweight model in Table 6, achieved the best Accuracy results on all three datasets, similar to the LRASPP results shown in Table 5. The UIIS and TrashCan datasets, similar to the SUIM dataset, had a large proportion of the Background class, resulting in an unbalanced class distribution. BiSeNet's advantage in the Accuracy metric further validated that extremely lightweight models were more effective at segmenting large target objects but performed poorly in segmenting details and small target objects. Our model outperformed Segmenter in all metrics on the UIIS dataset. On the TrashCan dataset, our model performed better than Segmenter in mIoU, mPrecision, and Accuracy. This indicates that among the fast models, our model demonstrated better overall performance in underwater target segmentation.

In Table 6, no single model achieved the best performance across all datasets and metrics. Each model has its own strengths and characteristics. Among all the models compared, our model achieves a balance between speed and accuracy, fulfilling the requirements for real-time underwater operation while maintaining good segmentation performance in complex underwater environments. On the UIIS dataset, our model achieved the highest mRecall of 67.6%, while the mIoU was only 0.1% lower than the best, and it attained the second-best result in Accuracy metric. Compared to the SUIM dataset, our model shows better comparative performance on the UIIS and TrashCan datasets, which have poorer image quality. This demonstrates the effectiveness of our method in more complex underwater environments.

There is still room for improvement in our model under the SUIM dataset conditions when compared to other models in Table 6. The variation in object size within the SUIM dataset is greater than that in the other two datasets. Additionally, the images in this dataset are clearer, and the features such as color and texture are simpler. While the multi-scale fea-

Table 5 Performance comparison with classic model on SUIM dataset

Model	Backbone	mIoU	mPrecision	mRecall	Accuracy	Average Frame Rate
FCN-8S [10]	ResNet	66.8%	81.5%	78.6%	84.1%	119.6 FPS
U-Net [11]	VGG	63.3%	79.4%	75.0%	83.2%	44.1 FPS
LRASPP [27]	MobileNetV3	70.9%	84.4%	81.6%	**86.2%**	**130.7 FPS**
HRNet [24]	HRNet w32	70.2%	83.0%	81.6%	85.2%	29.3 FPS
DeepLabV3+ [15]	Xception	68.5%	81.8%	80.7%	84.7%	25.2 FPS
DeepLabV3+ [15]	HRNet w32	69.1%	82.0%	81.3%	84.8%	22.2 FPS
DeepLab-FusionNet	Ours	**71.8%**	**84.9%**	**82.3%**	85.9%	59.2 FPS

Table 6 Performance comparison with lightweight models and transformer-based models on three datasets

Dataset	Model	backbone	miou	mPrecision	mRecall	Accuracy	Average Frame Rate
	BiSeNet [37]	BiSeNet	69.5%	84.4%	80.3%	**96.6%**	**206 FPS**
	PP-LiteSeg [38]	STDC2	**82.0%**	**90.8%**	**89.4%**	90.3%	55.0 FPS
	Segmenter [39]	ViT-L/16	78.0%	87.3%	88.0%	87.3%	87.9 FPS
	SegFormer [40]	MiT-B0	81.4%	90.6%	88.8%	91.0%	55.5 FPS
	TopFormer-B [41]	TopFormer	80.9%	90.6%	88.4%	90.4%	41.7 FPS
SUIM	DeepLab-FusionNet	Ours	71.8%	84.9%	82.3%	85.9%	59.2 FPS
	BiSeNet	BiSeNet	37.0%	50.0%	50.1%	**94.1%**	**206 FPS**
	PP-LiteSeg	STDC2	50.5%	66.1%	64.1%	76.8%	55.0 FPS
	Segmenter	ViT-L/16	51.5%	64.1%	63.3%	64.4%	87.9 FPS
	SegFormer	MiT-B0	50.0%	63.1%	66.3%	77.3%	55.5 FPS
	TopFormer-B	TopFormer	**51.8%**	**66.2%**	64.1%	77.5%	41.7 FPS
UIIS	DeepLab-FusionNet	Ours	51.7%	65.2%	**67.6%**	80.7%	59.2 FPS
	BiSeNet	BiSeNet	35.0%	68.9%	40.1%	**99.4%**	**206 FPS**
	PP-LiteSeg	STDC2	41.6%	65.6%	52.1%	95.4%	55.0 FPS
	Segmenter	ViT-L/16	38.0%	46.7%	**72.1%**	46.7%	87.9 FPS
	SegFormer	MiT-B0	44.1%	71.0%	52.8%	95.7%	55.5 FPS
	TopFormer-B	TopFormer	**48.1%**	**71.8%**	58.0%	96.0%	41.7 FPS
TrashCan	DeepLab-FusionNet	Ours	43.0%	66.0%	52.9%	95.6%	59.2 FPS

ture fusion module can retain detailed features and improve segmentation performance, it may also cause the model to focus excessively on fine details, leading to overfitting. Consequently, the model may perform poorly compared to those with stronger generalization capabilities when dealing with simpler or less challenging datasets (such as the targets in the SUIM dataset), as these datasets do not provide the complexity required to fully leverage the model's strengths. Moreover, the ASPP module captures multi-scale feature information using convolutions of different scales. However, fixed scales and strides may not effectively capture details across all scales, resulting in reduced performance on datasets where object scale varies greatly.

Although Transformer-based models have higher computational complexity, they have demonstrated significant advantages in underwater object segmentation tasks, particularly in models like SegFormer and TopFormer-B. Compared to pure CNN models such as DeepLab-FusionNet, Transformer-based models capture global dependencies through the self-attention mechanism, rather than focusing solely on local neighborhood information. This global perspective enables the model to dynamically adjust its attention to different scale features based on context, allowing for a better understanding and handling of large-scale variations or complex scenes. This insight is particularly valuable for our research, as it suggests potential future directions involving the integration of Transformer architectures with multi-scale feature fusion for underwater object segmentation tasks.

(3) Impact of Structural Reparameterization on Model Inference Speed

Table 7 presents a comparison of the model inference speed for the DeepLab-FusionNet approach before and after structural reparameterization, where the unit for model parameters is million(M). Compared to the training phase, the DeepLab-FusionNet approach reduces the number of parameters by only 0.03M during the validation phase while the prediction speed improves by 13.8%. It indicates that the structural reparameterization-based DeepLab-FusionNet approach can effectively reduce redundant memory consumption and access costs while maintaining the same prediction accuracy.

Table 7 Comparison of model inference speed before and after structural reparameterization of DeepLab-FusionNet method

Model	Operating Mode	mIoU	Parameters	Average Frame Rate
DeepLab-FusionNet	Training	71.8%	4.74 M	52.0 FPS
DeepLab-FusionNet	Validation	71.8%	**4.71 M**	**59.2 FPS**

Table 8 Results of ablation experiments on SUIM dataset

Experimental groups	Multi-scale feature feature map fusion module	Parallel convolution branch	ASPP module	Deep-shallow level information association module	mIoU	mPrecision	mRecall	Accuracy
1	✓				64.9%	80.5%	77.0%	83.6%
2	✓	✓			65.2%	80.2%	77.4%	83.4%
3	✓		✓		68.9%	83.6%	79.6%	85.3%
4	✓			✓	66.9%	81.7%	78.4%	84.5%
5	✓	✓	✓		69.5%	84.0%	80.2%	85.2%
6	✓	✓		✓	68.1%	82.2%	79.6%	84.5%
7	✓		✓	✓	70.4%	84.4%	80.8%	**86.4%**
8	✓	✓	✓	✓	**71.8%**	**84.9%**	**82.3%**	85.9%

4.3.3 Ablation experiment

In this section, the impact of each module used in the proposed method on the segmentation accuracy of the model was analyzed through ablation experiments. The results of the ablation experiments on the SUIM dataset are shown in Table 8. From the results of the ablation experiments, the effectiveness of the coupled different receptive field structure in the proposed improvement method is demonstrated. Furthermore, it also demonstrates the necessity of the parallel convolution branch, ASPP module, and deep-shallow level information fusion module in the inverted residual convolution module in terms of segmentation performance metrics.

As seen in Table 8, when comparing the models with and without the parallel convolution branch, such as in Groups 7 vs. 8, there is a slight decrease in the Accuracy metric, and in Groups 1 vs. 2, both Accuracy and mPrecision exhibit a slight drop. While the parallel convolution branch improves feature extraction capabilities, it also inevitably introduces overfitting to certain details, slightly reducing generalization ability and leading to a decrease in the Accuracy metric. At the same time, the parallel branch can couple multi-scale receptive field information, addressing the issue of missed detections caused by variations in object scale in underwater environments, thus improving Recall. However, this may also introduce more false positives, as the model may predict more areas as belonging to object classes, which can lead to a slight decline in the mPrecision metric.

5 Conclusion

We proposed a DeepLab-FusionNet method based on Deep LabV3+ for underwater object segmentation in this paper. Firstly, we designed a multi-scale feature map fusion method as the encoding network and utilized an improved inverted residual module based on depth-wise separable convolution as its basic convolutional unit. This method effectively couples multi-scale features and semantic information of the feature maps, thereby improving the segmentation accuracy and capability of detail retention on marine objects. Furthermore, we introduced structural reparameterization techniques in the improved inverted residual module, in order to convert multi-branch convolution to single-branch convolution. Through this approach, this model achieved an average frame rate improvement of 7.2 FPS during the inference phase compared with the training phase without any accuracy loss. Moreover, the ASPP method was integrated into the proposed model to efficiently extract multi-scale receptive field information from the feature maps, which improved the segmentation capability of multi-size objects in the underwater images. Finally, in order to solve the problem of loss of target details and spatial information caused by increasing the depth of convolutional network, we constructed a structure which strengthened the association of semantic information of deep and shallow level in the network.

The advantages of DeepLab-FusionNet method have been shown in the experimental results based on SUIM, UIIS and TrashCan datasets. Visualization on the SUIM dataset revealed that the model performed better in segmenting objects of varying sizes and edge details in marine scenes compared to classic segmentation methods, with particularly outstanding performance in segmenting small to medium-sized targets. Furthermore, the performance on SUIM dataset showed that our model DeepLab-FusionNet could achieve the best mIoU, mPrecision and mRecall compared to classic models. The comparison of performance between lightweight models and transformer-based models across the three datasets demonstrate that our model achieves

a balance between speed and accuracy. The inference speed of proposed model can reach 59.2 FPS, which meets the real-time requirement of underwater devices. Finally, to further verify the positive impacts of each module used in the proposed model, we conducted the ablation experiment on the SUIM dataset. The results showed that DeepLab-FusionNet achieved its optimal performance when all modules are combined, proving the necessity of each module.

However, it is acknowledged that our model needs further enhancement to better handle the segmentation of extremely small-sized targets and to improve semantic understanding in complex scenarios such as overlapping, occluded targets, and challenging lighting conditions. Besides, in the performance comparison of lightweight models and Transformer-based models across the three datasets, the poor performance observed on the SUIM dataset is likely due to overfitting and the limitations of fixed-scale convolutions in the multi- scale feature map fusion module and ASPP module. To address these issues, future improvements will include integrating a Transformer module into the encoder network to enhance global information extraction and semantic understanding, and refining the inverted residual convolution module with methods that adaptively adjust the receptive field, such as the SKAttention method, to enhance attention to critical features and accuracy for extremely small-sized targets.

Acknowledgements This research is currently supported by Guangdong Province Basic and Applied Basic Research Foundation(2022A1515110420), Shenzhen Science and Technology Program(Grant No.RCBS20221008093227028), and National Natural Science Foundation of China(Grant No.12405214). We would like to thank Ming Yang for his participation in improving the manuscript and for his dedicated efforts in collecting the dataset required for new experiments.

Author Contributions **Chengxiang Liu:** Conceptualization, Methodology, Supervision, Writing - Reviewing and Editing, Project administration. **Haoxin Yao:** Software, Visualization, Data curation, Writing-Original Draft. **Wenhui Qiu:** Software, Methodology, Data curation. **Hongyuan Cui:** Supervision, Visualization, Investigation. **Yubin Fang:** Investigation, Validation. **Anqi Xu:** Conceptualization, Formal analysis, Supervision, Writing-Reviewing and Editing, Funding acquisition.

Data availability and access The datasets such as SUIM, UIIS and TrashCan used in this research are available from the reference [21, 35, 36] respectively. All data generated or analysed during this study are included in this published article.

Declarations

Competing interests The authors declare that they have no competing interest to this work.

Ethical and informed consent for data used The authors of the submitted manuscript declare that does not involve any ethical issues.

References

1. Hong L, Wang X, Zhang D (2024) Cfd-based hydrodynamic performance investigation of autonomous underwater vehicles: A survey. Ocean Eng 305:117911
2. Osayi Philip Igbinenikaro OOA, Etukudoh EA (2024) A comparative review of subsea navigation technologies in offshore engineering projects. Int J Front Eng Technol Res 6(2):019–034
3. Hasan K, Ahmad S, Liaf AF, Karimi M, Ahmed T, Shawon MA, Mekhilef S (2024) Oceanic challenges to technological solutions: A review of autonomous underwater vehicle path technologies in biomimicry, control, navigation, and sensing. IEEE Access 12:46202–46231
4. Huy DQ, Sadjoli N, Azam AB, Elhadidi B, Cai Y, Seet G (2023) Object perception in underwater environments: A survey on sensors and sensing methodologies. Ocean Eng 267
5. Li M, Zhang H, Gruen A, Li D (2024) A survey on underwater coral image segmentation based on deep learning. Geo-spatial Inf Sci p 1–25
6. Pergeorelis M, Bazik M, Saponaro P, Kim J, Kambhamettu C (2022) Synthetic data for semantic segmentation in underwater imagery. in OCEANS. Hampton Roads. IEEE 2022:1–6
7. Ji L, Du Y, Dang Y, Gao W, Zhang H (2024) A survey of methods for addressing the challenges of referring image segmentation. Neurocomputing 583:127599
8. Mo Y, Wu Y, Yang X, Liu F, Liao Y (2022) Review the state-of-the-art technologies of semantic segmentation based on deep learning. Neurocomputing 493:626–646
9. Hao S, Zhou Y, Guo Y (2020) A brief survey on semantic segmentation with deep learning. Neurocomputing 406:302–321
10. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation.' in Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3431–3440
11. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. in Medical Image Computing and Computer-Assisted Intervention-MICCAI, 18th International Conference, Munich, Germany, October 5–9, Proceedings, Part III 18. Springer 2015:234–241
12. Wang J, Liu X (2021) Medical image recognition and segmentation of pathological slices of gastric cancer based on deeplab v3+ neural network. Comput Methods Prog Biomed 207:106210
13. Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Patt Anal Mach Intell 40(4):834–848
14. Bai Z, Jing J (2023) Mobile-deeplab: a lightweight pixel segmentation-based method for fabric defect detection. J Intell Manuf
15. Chen L, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. in Proceedings of the European conference on computer vision (ECCV), pp 801–818
16. Zhuang P, Wang Y, Qiao Y (2021) Wildfish++: A comprehensive fish benchmark for multimedia research. IEEE Trans Multimed 23:3603–3617
17. Ditria EM, Connolly RM, Jinks EL, Lopez-Marcano S (2021) Annotated video footage for automated identification and counting of fish in unconstrained seagrass habitats. Front Marine Sci 8
18. Cai L, Chen C, Chai H (2021) Underwater distortion target recognition network (udtrnet) via enhanced image features. Comput Intell Neurosci 2021:1–10

19. Zhang P, Yu H, Li H, Zhang X, Wei S, Tu W, Yang Z, Wu J, Lin Y (2023) Msgnet: multi-source guidance network for fish segmentation in underwater videos. Front Marine Sci 10
20. Martin-Abadal M, Guerrero-Font E, Bonin-Font F, Gonzalez-Cid Y (2018) Deep semantic segmentation in an auv for online posidonia oceanica meadows identification. IEEE Access 6(2018):60956–60967
21. Islam MJ, Edge C, Xiao Y, Luo P, Mehtaz M, Morse C, Enan SS, Sattar J (2020) Semantic segmentation of underwater imagery: Dataset and benchmark. in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 1769–1776
22. Nezla N, Haridas TM, Supriya M (2021) Semantic segmentation of underwater images using unet architecture based deep convolutional encoder decoder model. in 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol 1. IEEE, pp 28–33
23. Zhou J, Yang T, Zhang W (2023) Underwater vision enhancement technologies: a comprehensive review, challenges, and recent trends. Appl Intell 53(3):3594–3621
24. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5693–5703
25. Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X et al (2020) Deep high-resolution representation learning for visual recognition. IEEE Trans Patt Anal Mach Intell 43(10):3349–3364
26. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. in Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
27. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for mobilenetv3. in Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324
28. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. in Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
29. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. in Proc of the AAAI Conf Artif Intell 31(1)
30. Rahnemoonfar M, Dobbs D (2019) Semantic segmentation of underwater sonar imagery with deep learning. in IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp 9455–9458
31. Tolie HF, Ren J, Elyan E (2024) Dicam: Deep inception and channel-wise attention modules for underwater image enhancement. Neurocomputing 584:127585
32. Liu F, Fang M (2020) Semantic segmentation of underwater images based on improved deeplab. J Marine Sci Eng 8(3):188
33. Jin A, Zeng X (2023) A novel deep learning method for underwater target recognition based on res-dense convolutional neural network with attention mechanism. J Marine Sci Eng 11(1):69
34. Ding X, Zhang X, Ma N, Han J, Ding G, Sun J (2021) Repvgg: Making vgg-style convnets great again. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13733–13742
35. Lian S, Li H, Cong R, Li S, Zhang W, Kwong S (2023) Watermask: Instance segmentation for underwater imagery. in 2023 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE
36. Hong J, Fulton M, Sattar J (2020) Trashcan: A semantically-segmented dataset towards visual detection of marine debris. arXiv:2007.08097
37. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N (2021) Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vis 129(11):3051–3068
38. Peng J, Liu Y, Tang S, Hao Y, Chu L, Chen G, Wu Z, Chen Z, Yu Z, Du Y et al (2022) Pp-liteseg: A superior real-time semantic segmentation model. arXiv:2204.02681
39. Strudel R, Garcia R, Laptev I, Schmid C (2021) Segmenter: Transformer for semantic segmentation. in 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE
40. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P (2021) Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in neural information processing systems 34(2021):12077–12090
41. Zhang W, Huang Z, Luo G, Chen T, Wang X, Liu W, Yu G, Shen C (2022) Topformer: Token pyramid transformer for mobile semantic segmentation. in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com