




LDANet: the laplace-guided detail-constrained asymmetric network for real-time semantic segmentation

Zhifang Zhu¹ · Wenhao Wu¹ · Hongzhou Wang²  · Hengyu Li³ · Yibo He¹ · Yuanjie Liu¹ · Quanguo Lu¹ · Xiaohuang Zhan^{4,5}

Received: 4 May 2023 / Revised: 25 October 2023 / Accepted: 7 November 2023 /

Published online: 27 November 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The current mainstream image semantic segmentation networks often suffer from mis-segmentation, segmentation discontinuity, and high model complexity, which limit their application in real-time processing scenarios. The work established a lightweight neural network model for semantic segmentation to address this issue. The network used a dual-branch strategy to solve low semantic boundary segmentation accuracy in semantic segmentation tasks. The semantic branch applied the characteristics of the deeplabv3+ model structure. Besides, dilated convolutions with different dilation rates in the encoder were used to expand the receptive field of convolutional operations and enhance the ability to capture local features. The boundary refinement branch extracted second-order differential features of the input image through the Laplace operator, and it gradually refined the second-order differential features through a feature refinement extraction module to obtain advanced semantic features. A convolutional block attention module was introduced to filter the features from both the channel and spatial dimensions and finally fused with the semantic branch to achieve constrained segmentation boundary effects. Based on this, a multi-channel attention fusion module was proposed to aggregate features from different stages. Low-resolution features were first up-sampled and then fused with high-resolution features to enhance the spatial information of high-level features. Proposed network's effectiveness was demonstrated through extensive experiments on the MaSTr1325 dataset, the MID dataset, the Camvid dataset, and the PASCAL VOC2012 dataset, with mIoU of 98.1, 73.1, and 81.1% and speeds of 111.40, 100.36, and 111.43 fps on a single NVIDIA RTX 3070 GPU, respectively.

Keywords Semantic segmentation · Attention mechanism · Dual-branch structure · Real-time analysis · Neural network

1 Introduction

Unmanned surface vessels (USVs) equipped with autonomous navigation systems and obstacle avoidance functions have become an important tool for assisting or replacing human work and have attracted widespread attention with increased tasks such as coastal mapping, environmental monitoring, ecological monitoring, and marine rescue. The semantic-level perception recognition of water surface scene targets is required during water surface navigation to achieve the autonomous navigation of unmanned vessels, including target shape, posture, and division of feasible driving areas. Besides, real-time inference decisions are made based on the recognition results. An efficient and lightweight real-time semantic segmentation network should be designed due to the limited size and working mode of unmanned vessels. Thus, the target recognition can be completed with high accuracy in real time. This is also one of the current research hotspots.

Common semantic segmentation networks in the design process consider the design of the network backbone, such as ResNet-101 [17], Xception [8], and HRNet [38], instead of the inference speed and computational cost. These structures, multi-layered, complex, and deep, can extract integrated features from a larger spatial range and have stronger feature expression capabilities. However, these massive backbone models often have high computational complexity and slow inference speeds, which is not suitable for some practical applications requiring low latency and high prediction accuracy from the model. Therefore, how to achieve more accurate segmentation results and ensure a high inference speed has become a challenging task.

Real-time semantic segmentation networks have been proposed recently. ENet [31] achieves high efficiency and accuracy using methods such as depth-wise separable convolution, skip connections, and early downsampling. Also, it has a fast running speed. ERFNet [33] uses a series of effective designs (e.g., an asymmetric encoder and non-bottleneck-1D modules) to improve the accuracy and efficiency of the network. DFANet [25] reduces the parameter quantity based on multi-scale feature propagation. Its feature extraction method uses sub-network aggregation and sub-stage aggregation to promote the interaction and aggregation between features at different levels and reduces the calculation amount. Therefore, accuracy is improved. BiSeNet [46] is a bilateral segmentation network. Two paths are designed to collect spatial and semantic information without affecting the segmentation speed. The spatial path uses more channels and shallower networks to retain rich spatial information and generate high-resolution features, while the context path uses fewer channels and deeper networks for rapid downsampling to obtain sufficient contextual information.

BiSeNetV2 [45] adds bilateral guided aggregation based on BiSeNet to integrate complementary information extracted by the detail and semantic branches. The layer guides the feature response of the detailed branch with the contextual information of the semantic branch, and it captures feature representations of different scales through different scale guides to accurately perform multi-scale information fusion. BiSeNetV3 [37] uses the STDCNet [12] backbone network to remove the time-consuming Spatial Path in BiSeNet. An attention refinement module (ARM) and a feature fusion module (FFM) are added to the original architecture to make BiSeNetV3 perform better than existing results in urban landscapes with traditional edge detection methods. ICEG [16] proposes the ESC module to eliminate boundary ambiguity by separating foreground and background features and using attention masks to reduce uncertain boundaries and eliminate false predictions. The scheme also leverages edge features to adaptively guide

segmentation and enhance feature-level edge information to achieve high-definition edges in segmentation results.

Lightweight convolutional neural networks can be primarily categorized into the following three types to obtain a real-time semantic segmentation algorithm that can be applied in practice: (1) Compression of pre-trained deep networks: A structure sparsity learning method (SSL method) is used to learn a compact structure from a large convolutional neural network [40], which reduces the computational cost. (2) Network quantization-based methods: The Quantized CNN method is used to quantize weights in the convolution and fully connected layers and to minimize the response error of each layer, which accelerates and compresses the CNN [43]. The method has a faster computational speed but poor segmentation performance. (3) Lightweight convolutional neural networks: The design focus of lightweight networks is to obtain a compact network that reduces both the computational cost and memory consumption.

Inspired by the Deeplab v3+ network, the work proposed an efficient real-time semantic segmentation network that balanced accuracy and speed (Fig. 1). The network retained the ASPP module of the Deeplab v3+ network and replaced the backbone network with the lighter Mobilenet v3 network. On this basis, a new decoder structure was designed and consisted of Laplace feature extraction, a multi-path channel attention feature fusion module (CAFEM), a convolutional block attention module (CBAM) [41], and a feature

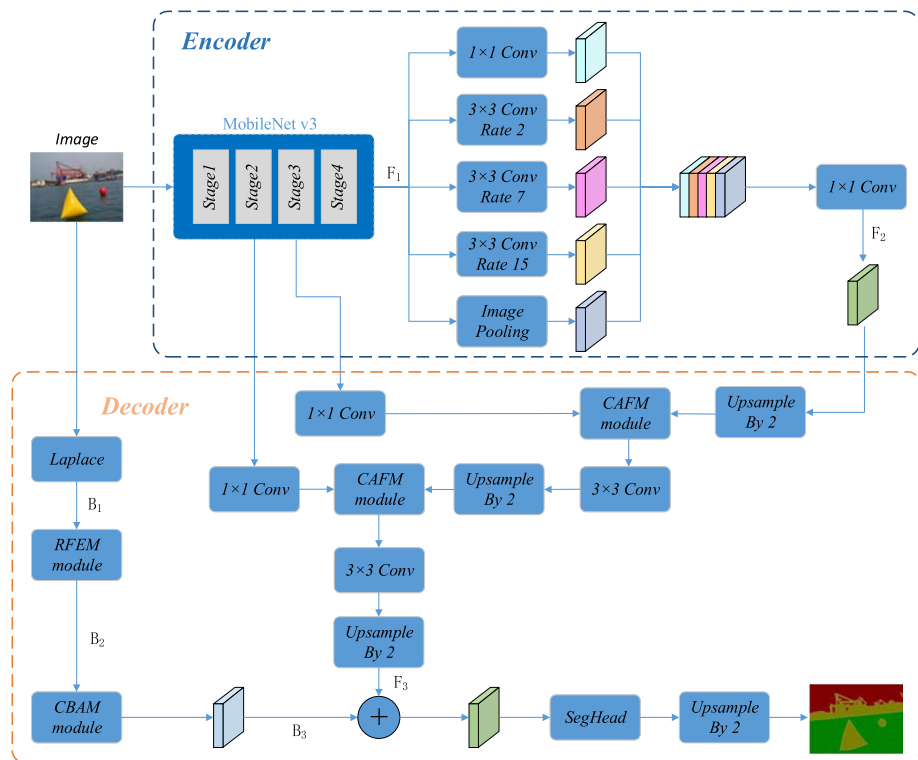


Fig. 1 LDANet’s detailed architecture, The Laplace module is applied to extract second-order differential features in the framework. Three layers of FREM {16, 64, 128} are used for feature refinement. The CAFEM module is used to fuse deep and shallow features

refinement extraction module (FREM). The CAFM fuses the semantic features extracted from the encoding network with low-level features through a channel attention mechanism. The Laplace operator is used to extract second-order differential information from the input image to obtain edge features. The FREM refines the edge features and gradually decomposes abstract features into high-level semantic features. Finally, the CBAM filters useful features at both the channel and spatial levels and removes useless features. Then the resulting features are added back to the final semantic features fused by the CAFM.

The main contributions of the work are summarized as follows:

- (1) LDANet, a lightweight real-time semantic segmentation network, was proposed for the semantic-level perception recognition of water surface scene targets by SUV. The network had a dual-branch structure of a semantic branch and edge refinement branch, which ensured good segmentation performance and maintained inference speed.
- (2) The work proposed an edge refinement branch composed of the Laplace operator, FREM, and CBAM, which constrained the segmentation results of the semantic branch through boundary contour information.
- (3) A multi-path channel attention feature fusion module was designed to suppress unimportant features and enhance important features through a channel attention mechanism. Context information and spatial information were fused.
- (4) LDANet was tested on an integrated dataset consisting of MaSTr1325 and MID, with a mIoU score of 98.1%. Additionally, when tested on the CamVid and PASCAL VOC2012 datasets, LDANet achieved the mIoU scores of 73.1 and 81.1%, respectively. The network achieved an inference speed of 100.36 frames/second with an input image size of 720×960 . Compared to other lightweight algorithms such as DSA, LDANet performs better in terms of segmentation performance.

2 Related work

Deep learning-based image semantic segmentation techniques have been rapidly developing recently. Image classification is the most fundamental task in computer vision, and the demand is also increasing for segmentation accuracy and speed based on image classification networks. This section introduces the development of relevant technologies.

Fully Convolutional Networks (FCN) proposed by Long et al. [29]. solved the semantic segmentation problem in image analysis. The FCN network based on VGGNet replaces fully connected layers with convolutional layers, which promotes traditional convolutional neural networks (CNN). U-Net proposed by Ronneberger et al. [34]. uses an encoder-decoder architecture. The introduction of skip connections fuses the feature maps of the encoder with the corresponding feature maps of the decoder, which addresses the roughness of the FCN network's output segmentation. SegNet proposed by Badrinarayanan et al. [1], similar to U-Net, uses an encoder-decoder structure; however, the encoder part of SegNet uses the first 13 layers of the VGG16 convolutional network, with each encoder layer corresponding to a decoder layer. It enhances the accuracy of image position information.

DeepLabv1 proposed by Chen et al. [5] introduces dilated convolution instead of standard convolution to increase the network's receptive field. A dense conditional random field (CRF) is introduced as post-processing to address blurred edges after image segmentation. Chen et al. [4] added the atrous spatial pyramid pooling (ASPP) module to DeepLabv1 to extract image features as completely as possible by combining pyramid pooling

and atrous convolution. Multiple atrous convolutions with different kernel sizes are used to obtain features at different scales. Chen et al. [6] used the multi-grid strategy to add batch normalization to the ASPP module and remove CRF post-processing, with DeepLabV3 proposed. After that, a decoder module is added based on DeepLabV3 to restore object boundaries caused by parallel hollow convolution [7]. Meanwhile, the Xception model is used for segmentation tasks and the deep separable hollow convolution is applied to ASPP and decoder, which improves the accuracy and speed of the network.

Researchers paid more attention to whether networks could be applied to different real-world scenarios. The efficient residual factorized net (ERFNet) [33] is based on residual connections and depth-wise separable convolutions. The ERFNet replaces the residual module with a non-bottleneck module and uses only 1D asymmetric convolutions internally, which reduces the number of parameters and ensures segmentation accuracy. ELA-Net [44] is a lightweight network based on an asymmetric encoder framework. A novel efficient context guidance strategy (ECG) is used to learn and capture complex contextual semantic information in the encoder, with good results. FEDER [15] model proposes a method for auxiliary edge reconstruction tasks using high-order ODE solvers and ODE-inspired edge reconstruction modules to address the fuzzy boundary problem in the ED challenge. This approach can help generate accurate segmentation results with precise object boundaries.

2.1 Encoder-decoder

The encoder-decoder architecture has been widely used in image segmentation tasks based on deep learning to improve the accuracy of semantic segmentation networks. Symmetric encoder-decoder structures usually include multiple feature extraction and upsampling steps. ResNet [17] and Xception [8] introduce multiple branch structures or connect shallow and deep features based on existing backbone networks with excellent feature extraction capabilities. However, optimizing the network structure increases more parameters, which limits their use in scenarios with limited computational resources. ENet [31] is a lightweight network that balances network accuracy and efficiency using an asymmetric encoder-decoder structure. The encoder has a deeper network structure and a larger receptive field, while the decoder is relatively shallow. The encoder uses skip connections to extract high-level semantic features, and the decoder fine-tunes the feature map details by upsampling encoder output.

BANet [51] adopts a dual-branch encoder-decoder structure, where two encoders process different levels of feature representations. The corresponding decoder combines these features to obtain the final detection and segmentation results. This dual-branch structure can utilize hierarchical features at different stages to improve task performance. OverSegNet [26] achieves good results in image superpixel segmentation. OverSegNet uses a NAS encoder, which is more efficient than existing encoders. Oversplitting is optimized through two steps: first, over-parameterize the encoder with small convolution kernels. Then data-driven network architecture search (NAS) is used to learn and select convolution kernels specific to over-splitting types, which optimizes network performance. Besides, OverSegNet proposes a clustering-specific decoder to calculate the correlation between pixels and superpixels. Since the decoder is optimized for superpixel clustering, the use of skip connections in the structure can significantly improve the accuracy of superpixel clustering without increasing the computational cost.

2.2 Attention mechanism

Attention mechanisms are widely used in neural networks. The more parameters a model has, the stronger its expressive power and the more information it can store in the learning process of neural networks. Attention mechanisms can quickly scan the global information in an image and allocate computational resources to more important tasks in situations where computational resources are limited, which reduces or filters the focus on other information.

There are two types of trainable attention: hard/local attention and soft/global attention. Hard attention relies on parameter updates in reinforcement learning, which hinders model training. Soft attention is probabilistic and can be trained using standard backpropagation without Monte Carlo sampling. Wu et al. [42] proposed a flexible attention model that uses global pooling and deformable convolutions to model spatial and channel attention. A dense gating and collaboration method is introduced to drive spatial and channel attention as dynamic attention. Self-attention mechanisms can establish long-range dependencies within sequences. OCNet [49] and DANet [13] use self-attention mechanisms to obtain relevant contextual information. CCNet [21] proposes a more efficient attention module—the cross-attention module (CCA). A category-consistent loss is introduced to force the CCA to acquire more diverse features.

BiSeNet V2 [45] uses two parallel branches to handle low-resolution and high-resolution features. One branch focuses on global information and the other focuses on local fine-grained information. The results of the two branches are fused using attention mechanisms to improve segmentation accuracy. Qu et al. [30] propose a global attention dual pyramid network (GADPNET), which uses an improved global attention module with a lightweight contextual branch to capture more local information on low-level features. GADPNET captures multi-scale features with a single high-level feature through a pyramid decoder structure. Different low-level multi-scale features are fused under the guidance of the improved global attention module to enhance the multi-scale feature capture of semantic segmentation networks.

2.3 Feature fusion

The fusion of features at different scales is an important approach to improve segmentation performance, and feature fusion structures are also effective for obtaining dense features and restoring image information. RefineNet [27] introduced complex refinement modules at each upsampling stage between the encoder and decoder to extract multiscale features. SENet [20] adopts a new “feature recalibration” strategy. Besides automatically acquiring the importance of each feature channel by learning, it enhances useful features and suppresses less useful features for the current task. The SENet network is effective for large-scale targets but performs poorly for small-scale targets. Yu et al. [48] extended the dense connection idea to develop more deeply aggregated structures and enhance the feature representation capability of the network. DFANet [25], a deep feature aggregation network, includes a backbone network with Xception as its backbone structure, subnet aggregation modules, and sub-stage aggregation modules. The sub-network is aggregated at the network level to combine advanced features, and the sub-stage is aggregated at the stage level to fuse semantic and spatial information. Dai et al. [9] proposed a multi-scale channel attention module (MS-CAM) to better fuse features with inconsistent semantics and scales. Channel attention is extracted through two branches

with different scales to fuse different scale features. They proposed an iterative attention feature fusion module (IAFF), which alternately integrates initial feature fusion with another attention module. Zhang et al. [50] proposed the architecture of the cascade fusion network (CFNet) to improve the performance of dense, detection, or segmentation prediction. CFNet inserts feature integration operations into the backbone network, which enables more parameters to be used for feature fusion and increases the richness of feature fusion.

3 Proposed networks

The work improves the encoder and decoder structures of the Deeplabv3+ network to address the issues in the semantic segmentation of water surface image datasets. Firstly, a multi-path channel attention fusion module and a prior-based boundary feature guidance method are proposed to constrain semantic boundaries. Secondly, the backbone network in the encoder section is modified by adjusting the sampling rates of the dilated convolutions in the ASPP module. These improvements will be detailed in the following sections. Figure 1 shows the network model structure proposed in the work.

MobileNetV3 is first used in the entire network to extract features from the original input and obtain feature F_1 . F_1 is then fed into the ASPP [47] structure, which consists of one 1×1 convolution, three 3×3 dilated convolutions with different dilation rates of 2, 7, and 15 to extract multi-scale contextual information from F_1 , and a global average pooling module. The features extracted from these convolutions are concatenated along the channel dimension and then fused by a 1×1 convolution to obtain high-level feature map F_2 .

High-level feature map F_2 is processed hierarchically from low to high using the multi-scale context aggregation fusion (CAFm) module during the decoding stage. This process gradually incorporates shallow-layer high-resolution images into the feature map to obtain feature F_3 , which contains rich semantic information. Contextual and spatial information are combined to improve segmentation performance throughout the process. Feature F_3 is then upsampled using bilinear interpolation and added with the boundary refinement branch to obtain B_3 . F_3 uses the bilinear interpolation method to perform 2 times upsampling and then is added to B_3 obtained by the boundary refinement branch. Features are fine-tuned through 3×3 convolution to reduce the dimension. This approach enhances the fusion of information from multiple scales, which improves results in semantic segmentation tasks.

The loss function used in the work is the commonly used cross-entropy loss function [36]. The formula of the loss function is

$$L = - \sum_{i=1}^N y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)}) \quad (1)$$

where L represents the training loss; N represents the number of samples; y represents the actual sample label, with a value range of $(0, 1)$; \hat{y} represents the predicted label value, with a value range of $(0, 1)$.

3.1 Encoder network

3.1.1 Backbone network

The larger size of the convolutional kernel means the larger receptive field of the network in CNNs, which results in more precise segmentation results. However, using larger kernels

also increases the number of parameters in the network, which hinders training and leads to a larger network size. The introduction of dilated convolutions solves this problem by enlarging the receptive field of the network without adding more parameters. MobileNetV1 [19] uses depthwise separable convolution to build a lightweight network by transforming standard convolutions into depthwise convolution and pointwise convolution, followed by batch normalization (BN) and ReLU activation in each layer. There is a corresponding convolutional kernel for each input dimension of depthwise convolution. The output feature map of the depthwise convolution is calculated as follows for the same input.

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \quad (2)$$

where \hat{K} is a depthwise convolutional kernel of size $D_K \times D_K \times M$, where the m^{th} filter in \hat{K} is applied to the m^{th} channel in F to produce the m^{th} channel of filtered output feature map \hat{G} .

Although dilated convolution greatly reduces the number of parameters, it cannot integrate multiple dimensions well. Therefore, the feature maps produced by the depthwise convolution should be fused using 1×1 convolution to generate new feature maps. 3×3 depthwise separable convolution requires 8–9 times fewer computations than traditional 3×3 convolution from the perspective of the computational cost. Besides, pooling layers are not used and strides can be used for downsampling tasks when deep neural networks are constructed using depthwise separable convolution modules.

MobileNetV2 [35] introduces the innovative unit of the inverted residual with linear bottleneck, which increased the number of layers and improves the overall accuracy and speed of the network. MobileNetV3 [18] proposes a novel idea of adding a neural network called “Squeeze-and-Excitation” in the core architecture. The core idea is to improve the quality of the network representation by explicitly modeling the interdependencies between convolutional feature channels. Specifically, it automatically learns the importance of each feature channel and uses information to enhance useful features and suppress irrelevant ones for the current task. The stride of the 5th downsampling module is changed to 1 to adapt to the semantic segmentation task and preserve more spatial information. Therefore, the main network performs only 4 downsamplings. To compensate for the reduction in receptive field resulting from this decrease in downsampling operations, we incorporated dilated convolutions with a dilation rate of 2 in the last three bottleneck modules. Table 1 shows the specific structure.

3.1.2 ASPP module

The ASPP module consists of a series of dilated convolutions and global average pooling with different dilation rates, which can extract object multi-scale features. However, improper setting of the parallel dilated convolution dilation rates can easily cause the gridding effect. Ref. [39] finds that continuous use of dilated convolutions with the same dilation rate in the ASPP module causes the convolution kernels to be discontinuous, which leads to the gridding effect (Fig. 2 (a)). Reasonable settings of dilation rates (Fig. 2 (b)) can obtain the contextual information of different scale targets and avoid the loss of relevant information.

$$M_i = \max[M_{i+1} - 2r_i, M_{i+1} - 2(M_{i+1} - r_i), r_i] \quad (3)$$

Table 1 Backbone network of the semantic encoding network used for extracting high-level information, SE is whether to use the attention mechanism; NL is the currently used non-linear activation function; s is the stride

Input	Operator	exp size	#out	SE	NL	s
$480^2 \times 3$	Conv2d	-	16	-	HS	2
$240^2 \times 16$	bneck, 3×3	16	16	-	RE	1
$240^2 \times 16$	bneck, 3×3	64	24	-	RE	2
$120^2 \times 24$	bneck, 3×3	72	24	-	RE	1
$120^2 \times 24$	bneck, 5×5	72	40	✓	RE	2
$60^2 \times 40$	bneck, 5×5	120	40	✓	RE	1
$60^2 \times 40$	bneck, 5×5	120	40	✓	RE	1
$60^2 \times 40$	bneck, 3×3	240	80	-	HS	2
$30^2 \times 80$	bneck, 3×3	200	80	-	HS	1
$30^2 \times 80$	bneck, 3×3	184	80	-	HS	1
$30^2 \times 80$	bneck, 3×3	184	80	-	HS	1
$30^2 \times 80$	bneck, 3×3	480	112	✓	HS	1
$30^2 \times 112$	bneck, 3×3	672	112	✓	HS	1
$30^2 \times 112$	bneck, 5×5	672	160	✓	HS	1
$30^2 \times 160$	bneck, 5×5	960	160	✓	HS	1
$30^2 \times 160$	bneck, 5×5	960	160	✓	HS	1
$30^2 \times 160$	Conv2d	-	960	-	HS	1

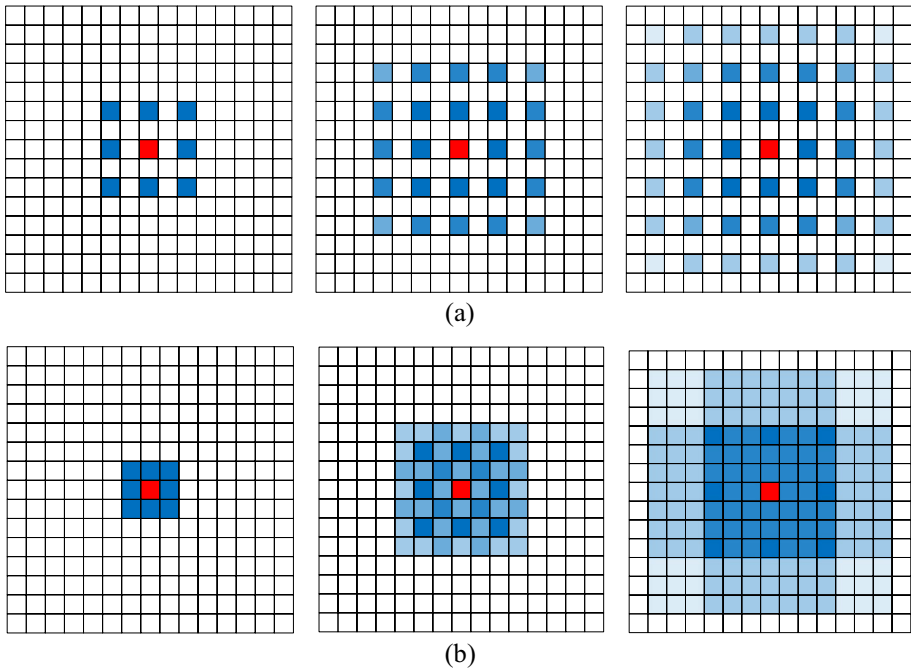


Fig. 2 Grid artifacts and ideal effects of dilated convolution [39]. **a** Grid artifacts in dilated convolution. **b** Ideal effect of dilated convolution

The expansion rate should follow the following principles: 1) The expansion rate should not contain a common divisor greater than 1; otherwise, the grid effect will occur. 2) Suppose that the expansion rates corresponding to N empty convolution kernels with the size of $k * k$ are $[r_1, \dots, r_i, \dots, r_n]$, then Eq. (3) is required to meet that M_2 is less than or equal to k . r_i represents the expansion rate of the i^{th} empty convolution; M_i represents the maximum expansion rate of the i^{th} empty convolution; $M_n = r_n$ by default. Therefore, following the above design principles, the work resets a set of cavity convolution with the expansion rates of 1, 2, 7, and 15 through experiments.

3.2 Decoder network

The decoder's role is to decode the feature information encoded by the encoder, offering pixel-level classification. An excellent encoder can improve model accuracy while maintaining inference speed. To decode feature information better with fewer parameters, a boundary constraint branch based on second-order derivatives and a multi-scale fusion attention mechanism are designed for the decoder, consisting mainly of CAFM, CBAM, and FREM.

The CAFM module filters out useful features through channel attention, enhancing their impact on the model and reducing the influence of irrelevant features on the results. Secondly, we up-dimension the low-level features to avoid them being overwhelmed by high-level features. The Laplace branch is used to constrain the segmentation boundaries, enabling more precise boundary segmentation. Generic semantic segmentation is achieved by gradually up-sampling the results obtained through high-magnification downsampling, which can obtain a larger receptive field but may lead to loss of boundary information. Although feature fusion can be used to supplement detailed information, it has limited effectiveness for fine boundaries. Therefore, we propose boundary constraints that constrain the expansion of semantic information while supplementing boundary information as much as possible without introducing any additional irrelevant information.

3.2.1 Multi-channel attention fusion module

Semantic segmentation networks usually extract semantic information in deep networks with a larger receptive field. However, the input feature map of the deep network loses a lot of detailed information due to multiple downsampling, which affects the final upsampled prediction result. Considering that shallow networks extract features from images with higher resolution and obtain output feature maps rich in detail information, it is crucial to fuse these output feature maps with those from deep networks. However, simply fusing low-level features with high-level features, such as SegNet [1] and U-Net [34], cannot well exploit the advantages of both, which wastes information. If these features can be integrated more appropriately, the result will be improved to some extent. Therefore, a multi-channel attention fusion module (CAFM) is proposed to fuse these features (Fig. 3).

The CAFM calculates the attention vector to guide the fusion of semantic features. It is divided into two parts: compression and excitation. The compression part aims to compress global spatial information, while the excitation part aims to predict the importance of each channel. The multi-channel features are first connected along the channel dimension in the compression part (Fig. 3). The feature map dimension after the connection is $H \times W \times 2C$, where H , W , and C are the height, width, and number of channels of the feature map before the connection, respectively. Then, the feature dimension is compressed from $H \times W \times 2C$

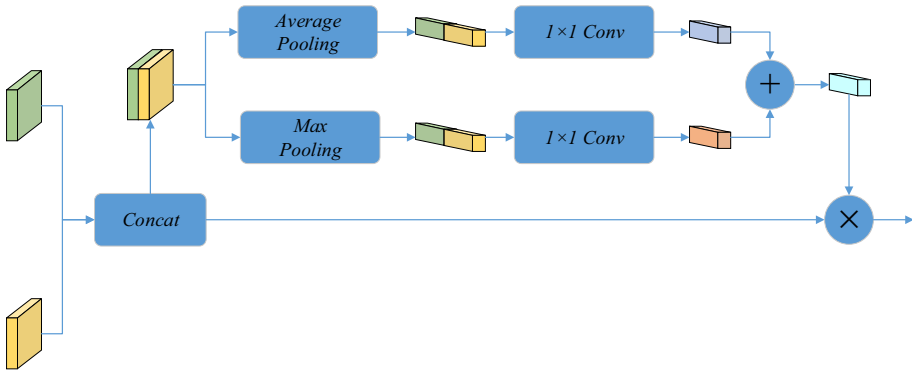


Fig. 3 Overall architecture of the multi-path channel attention feature fusion module

to $1 \times 1 \times 2C$ through global average pooling and global maximum pooling, i.e., $H \times W$ is compressed to 1×1 . Although the dimension is reduced, $H \times W$ is compressed to 1×1 , which can obtain a large receptive field with fewer parameters. The above operations are represented as follows:

$$Z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j) \tag{4}$$

The $1 \times 1 \times C$ feature tensor obtained from the compression part needs to be integrated into a fully connected layer in the excitation part to predict the importance of each channel. Then the importance values are used to weigh the corresponding channels in the feature map. A simple gating mechanism and a sigmoid activation function are used. The operations mentioned above are represented as follows:

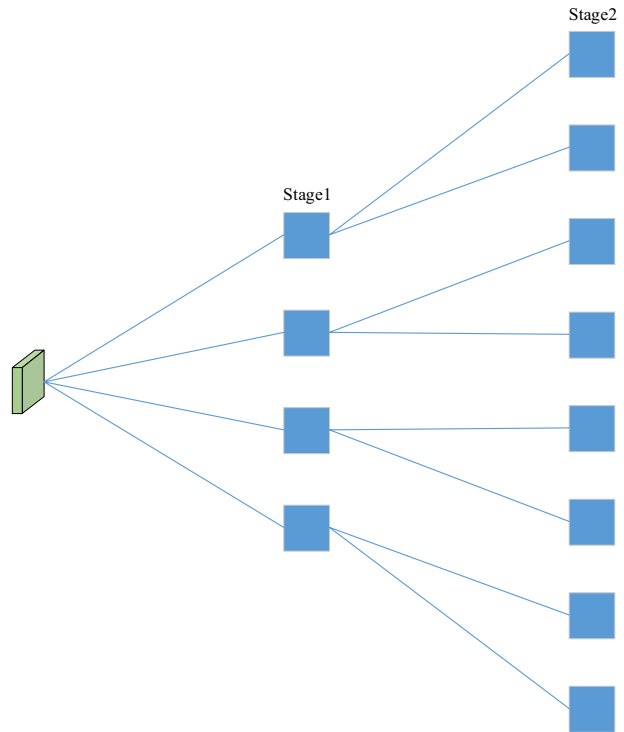
$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{5}$$

where δ is the ReLu function; σ is the Sigmoid function; $W_1 \in R^{\frac{2C}{r} \times 2C}$ and $W_2 \in R^{2C \times \frac{2C}{r}}$.

3.2.2 Boundary refinement branch

This article designs a branch of edge detail constraints based on Laplace. The input image is processed by Laplace operator to obtain the low-level boundary features B_1 of the input image. This feature map contains a lot of second-order differential information that is not related to the segmented object. In order to extract continuous and complete edge information to constrain the edge segmentation of semantic branch, this article designs the feature refinement extraction module (FREM) (Fig. 4) to extract the high-level semantic features in feature B1. This module gradually segments feature B1 from whole to details layer by layer, and each feature map independently extracts its own features again without being affected by other feature maps to obtain the most concise and complete edge feature map. This article uses a three-layer {16, 64, 128} structure of FREM, where the first layer uses a convolutional kernel with a stride of 2 to downsample the input features. The three layers use 7×7 , 5×5 , and 3×3 -sized convolutional kernels to refine the features of B_1 and obtain features B_2 . B_2 then goes through a Convolutional Block Attention Module (CBAM) [41] (Fig. 5) to filter out unwanted noise features and ultimately generate edge constraints. The CBAM module is composed of a channel attention module and a spatial attention

Fig. 4 2-layer {4, 8} structure of the feature refinement extraction module. This model uses a 3-layer {16, 64, 128} structure



module. The channel attention module assigns different weights to each channel to enhance the ability to identify important feature channels. This helps to eliminate noise information that is unrelated to the segmented object. The spatial attention module captures global contextual information to aid in inferring ambiguous pixels and enhancing the continuity of target edges. The channel attention module compresses the feature dimension of the obtained feature map from $H \times W \times 128$ to $1 \times 1 \times 128$. The compressed feature is integrated into a fully connected layer to predict the importance of each channel, and then it is applied to the corresponding channels of B_2 to obtain the feature B'_2 . The spatial attention mechanism compresses the feature dimension of B'_2 from $H \times W \times 128$ to $H \times W \times 1$. A 7×7 convolution is used to capture dependencies between features, enhancing the continuity of edges. Finally, this is applied to the feature F_3 to assist in obtaining more complete and accurate segmentation.

4 Experiments

The MaSTr1325 [2] dataset and MID [28] dataset were used to evaluate the effectiveness of the proposed method in the semantic understanding of the ocean. Besides, the generalization of the proposed method was evaluated in other application scenarios on two famous and challenging datasets (CamVid [3] and PASCAL VOC2012 [11]) which are commonly used for semantic segmentation.

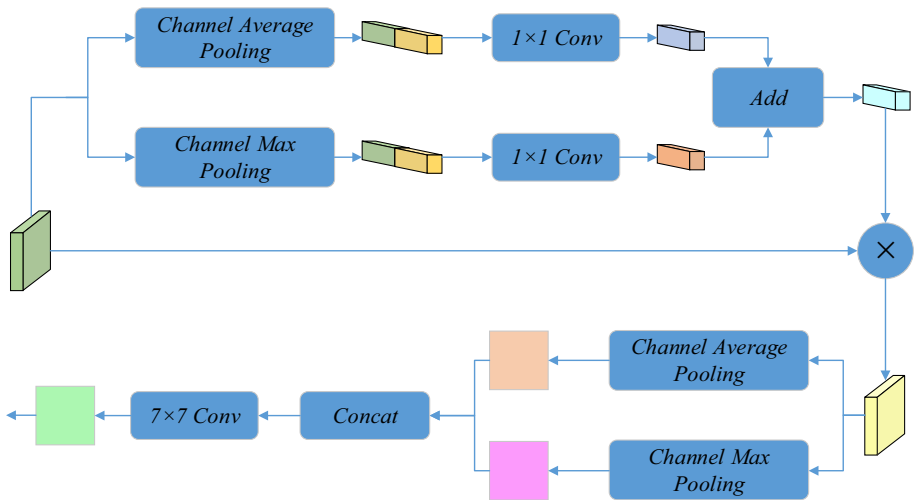


Fig. 5 Overall architecture of the convolutional block attention module including the channel attention module and the spatial attention module

Firstly, detailed information about the datasets and evaluation metrics is introduced in Sect. 4.1, and the implementation details are described in Sect. 4.2. Subsequently, an ablation study on the Camvid validation set is conducted, and the effectiveness of each component in LDANet is explained in Sect. 4.3. Finally, the inference speed, model parameters, and segmentation accuracy of the proposed model are compared with those of some advanced models on VOC 2012 and CamVid datasets in Sect. 4.4. The work follows the common semantic segmentation evaluation metric, Mean Intersection over Union (mIoU).

4.1 Datasets and evaluation metrics

The datasets used in the experiments were MaStr1325 [2] and MID [28], in addition to PASCAL VOC 2012 [11] and CamVid [3] datasets for the generalization evaluation of the network.

MaStr1325 is a new large-scale ocean semantic segmentation training dataset for developing obstacle detection methods for small coastal unmanned surface vehicles (USVs). The dataset contains 1,325 different images that were captured by a real USV for two years with a resolution of 384×512 , and it covers a range of realistic scenarios encountered in coastal surveillance tasks. All images were semantically labeled at the pixel level and synchronized with inertial measurements from onboard sensors. The dataset included three semantic labels and was split into a training set and a test set at a 9:1 ratio.

The MID dataset contained eight video sequences used for marine obstacle detection. There were 2,655 annotated images in the dataset with a resolution of 640×480 , and they were captured by the Shenghai No.8 USV. The obstacles in this dataset were divided into large obstacles (spanning the water's edge) and small obstacles (surrounded by water). The dataset was originally used for horizon segmentation and was pixel-level annotated in the work as obstacles, ocean, and sky for semantic segmentation. The dataset was split into a training set and a test set at a 9:1 ratio.

The CamVid dataset is a road scene dataset captured from driving a car, and it contains 701 images with a resolution of 960×720 . The quality of the images and annotations in this database was lower than that of Cityscapes, which challenges the dataset.

The PASCAL VOC 2012 semantic benchmark contained 20 foreground object classes and one background class. The original dataset had 1,464 and 1,449 images for training and validation, respectively. Additional annotations provided by [14] were also used in the work to enhance the training dataset.

Evaluation Metrics: Two metrics were used to quantitatively and qualitatively evaluate the accuracy performance of the proposed network model in the experiments, including pixel accuracy (PA), mean Intersection over Union (mIoU), parameters, and frames per second (FPS). The PA metric reflected the proportion of correct pixels in the semantic-segmentation image to the total pixels in the image. Equations (6) and (7) present the PA calculation and the mean IoU calculation, respectively.

$$PA = \sum_{i=0}^k \frac{P_{ii}}{\sum_{j=0}^k P_{ij}} \quad (6)$$

$$\text{mean IoU} = \frac{1}{k+1} \sum_{i=0}^k \frac{P_{ii}}{\sum_{j=0}^k P_{ij} + \sum_{j=0}^k P_{ji} - P_{ii}} \quad (7)$$

where k is the number of categories; $k+1$ is the number of categories containing background; p_{ij} is the total number of pixels predicted by categories i to j ; p_{ji} is the total number of pixels predicted by categories j to i ; p_{ii} is the total number of categories correctly predicted by category i . Parameters reflect the size of the network model's running parameter memory (Eq. (8)).

$$\text{Parameters} = (K^2 \times C_i) \times C_0 + C_0 \quad (8)$$

where C_i is the number of input channels of the convolution layer; K is the size of the convolution kernel; H and W are the sizes of the output feature graph, respectively; C_0 is the number of output channels of the convolution layer. FPS is used to reflect the image processing speed of the semantic segmentation model (Eq. (9)).

$$FPS = F \div T \quad (9)$$

where F is the frame number; T is elapsed time and is usually set as 1 s.

4.2 Implementation details

Training: The Adam [23] optimizer and SGD optimizer were combined to train all the models. The “poly” learning-rate update strategy [4] was employed to multiply the learning rate by $\left(1 - \frac{\text{step}}{\text{total_steps}}\right)^{0.9}$ after each step. We scaled the images by a factor of 0.5 to 2 and flipped them horizontally with a probability of 0.5. The images were normalized using mean=[0.485, 0.456, and 0.406] and variance=[0.229, 0.224, and 0.225] and were randomly cropped based on different datasets. The cross-entropy loss was used to train the network.

Setup: All experiments were conducted on CUDA 11.6. The platform used was Windows 10, with an Intel® Core™ i7-11700KF CPU@3.60 GHz CPU and an NVIDIA GeForce RTX3070 GPU.

4.3 Ablation study

Ablation experiments were used to validate the effectiveness of the CBAM module, FREM module, and CAFM module in the CamVid dataset.

4.3.1 CBAM module

The CBAM module improved the channel attention module and spatial attention module to obtain spatial features at the edges of features and provide spatial constraints for the features obtained by the semantic branch.

The experiment with the module removed was performed to verify the effectiveness of the CBAM module, and edge-based refining feature B_2 obtained by FREM was added directly to feature F_3 (Table 2). When the CBAM was removed, the network's mIoU on the test set decreased from 73.1 to 71.4%, and segmentation accuracy decreased by 1.7%. The CBAM module could improve the segmentation ability of the network.

Five experiments (S1, S2, S3, S4, and S5) were performed to find the most appropriate convolution kernel size in the spatial attention module of the CBAM module. The kernel sizes used in the spatial attention module were 3, 5, 9, 11, and 7, respectively. Compared with S5 (the proposed method in the work), the mIoU of S1, S2, S3, and S4 decreased by 0.8, 0.9, 0.8, and 0.9%, respectively (Table 3). Generally, increasing the kernel size of the spatial attention module can enable each weight value to obtain a larger receptive field, which brings a certain improvement. However, a larger receptive field will introduce more noise information.

4.3.2 FREM

The FREM further extracts features from the shallow second-order differential features, which enables it to quickly extract refined edge information.

The FREM was replaced with a fully-connected convolutional layer to test its impact on the network. The mIoU decreased from 73.1 to 71.5% (Table 3), demonstrating that the FREM can quickly extract gradient features to improve network performance.

However, simply introducing the FREM or CBAM alone did not improve segmentation accuracy. When both the FREM and CBAM were removed, the mIoU obtained was 71.5% (Table 3). Using both modules significantly increased the accuracy.

Table 2 Results of ablation experiments on CAFM, Laplace, FREM, and CBAM on the Camvid test set

CAFM	Laplace	FREM	CBAM	Params	fps	mIoU(%)
✓	✓	✓	x	8.77	109.17	71.4
✓	✓	x	✓	9.02	100.28	71.5
✓	✓	x	x	8.87	111.48	71.5
✓	x	x	x	8.77	117.64	70.1
x	✓	✓	✓	7.13	114.19	72.0
✓	✓	✓	✓	7.74	100.36	73.1

Table 3 Experimental results of different Kernel sizes in the spatial attention mechanism in CBAM

Strategy	Kernel size	mIOU(%)
S1	3	72.3
S2	5	72.2
S3	9	72.3
S4	11	72.2
S5	7	73.1

4.3.3 Laplace

The Laplace module extracted second-order differential features from the three channels of the original image (i.e., R, G, and B) to obtain boundary information between different semantics and constrain the prediction results.

The entire edge-based refining branch was removed to verify the effectiveness of the Laplace module. The mIoU decreased from 71.5 to 70.1%, compared to only removing the FREM and CBAM modules. The edge-based branch could constrain the edges of different semantics and improve segmentation accuracy.

4.3.4 CAFM

The CAFM, an efficient and simple feature fusion module, integrates low-level features and high-level features in semantic encoding networks. The work compared element-wise addition, concatenation, and CAFM in terms of performance (Table 4). The experiment showed that feature concatenation had a better inference speed and higher mIoU compared to the element-wise addition of features, but the CAFM method achieved a 3.1% increase in the mIoU at the cost of an inference speed of 13.83 fps. Higher semantic segmentation performance was achieved under real-time constraints.

4.4 Comparison to state-of-the-art methods

First, LDANet was compared with advanced networks on the MaStr1325 and MID datasets (Table 5). Second, LDANet was compared with some methods proposed in recent years on the VOC2012 and CamVid datasets to demonstrate its generalizability.

4.4.1 MaStr1325 and MID datasets

The MaStr1325 and MID datasets were integrated into one dataset, with a ratio of 9:1 between the training set and the test set. Images were uniformly scaled from 0.5 to 2

Table 4 Comparison of three types of feature fusion methods

Fusion method	Params	fps	mIoU
Add	8.04	107.65	67.3
Concatenate	7.16	114.19	70.0
CAFM	7.74	100.36	73.1

Table 5 Segmentation results of the three classes on the integrated dataset of the MaSTr1325 dataset and the MID dataset for the model

Method	Sea	Sky	Land	mIoU	Correct	Params	fps
Deeplab V3 [6]	99.5	98.9	93	97.1	99.3	11.02	167.08
Deeplab V3+[7]	99.5	99.2	94.6	97.8	99.5	11.72	144.24
Ours	99.6	99.3	95.2	98.1	99.5	7.74	111.40

times using 480 as the base, and then they were cropped to a size of 480×480 . The mini-batch size was set to 12 for training images. Figure 6 shows some example results on the dataset, and Table 5 presents the comparison of the IoUs of three classes of LDANet and the Deeplab v3 [6] and Deeplab v3+[7] networks on the dataset. LDANet

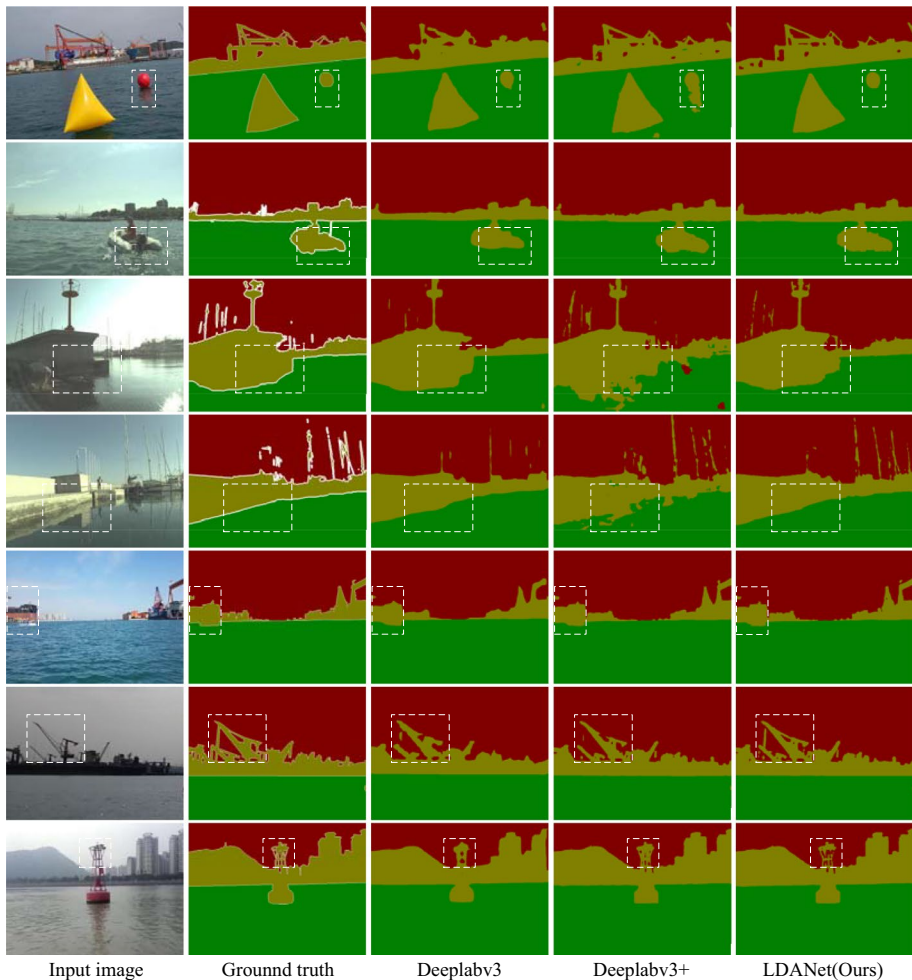


Fig. 6 Visual examples from the test sets of the MaSTr1325 dataset and the MID dataset

is more accurate than Deeplab v3 in edge and small object performance. Even though its fps decreases by 55.68, it still meets real-time requirements. Although Deeplab v3+ has made significant progress in edge detection compared to Deeplab v3, its ability to distinguish sea surface reflections is poor.

When shallow features are introduced, features are not well fused, which causes the influence of useless shallow features on the results. The CAFM module of LDANet addresses this problem and selects shallow features. The edge refinement branch constrains the semantic branch to ensure the continuity of semantic edges for accurate semantic boundaries.

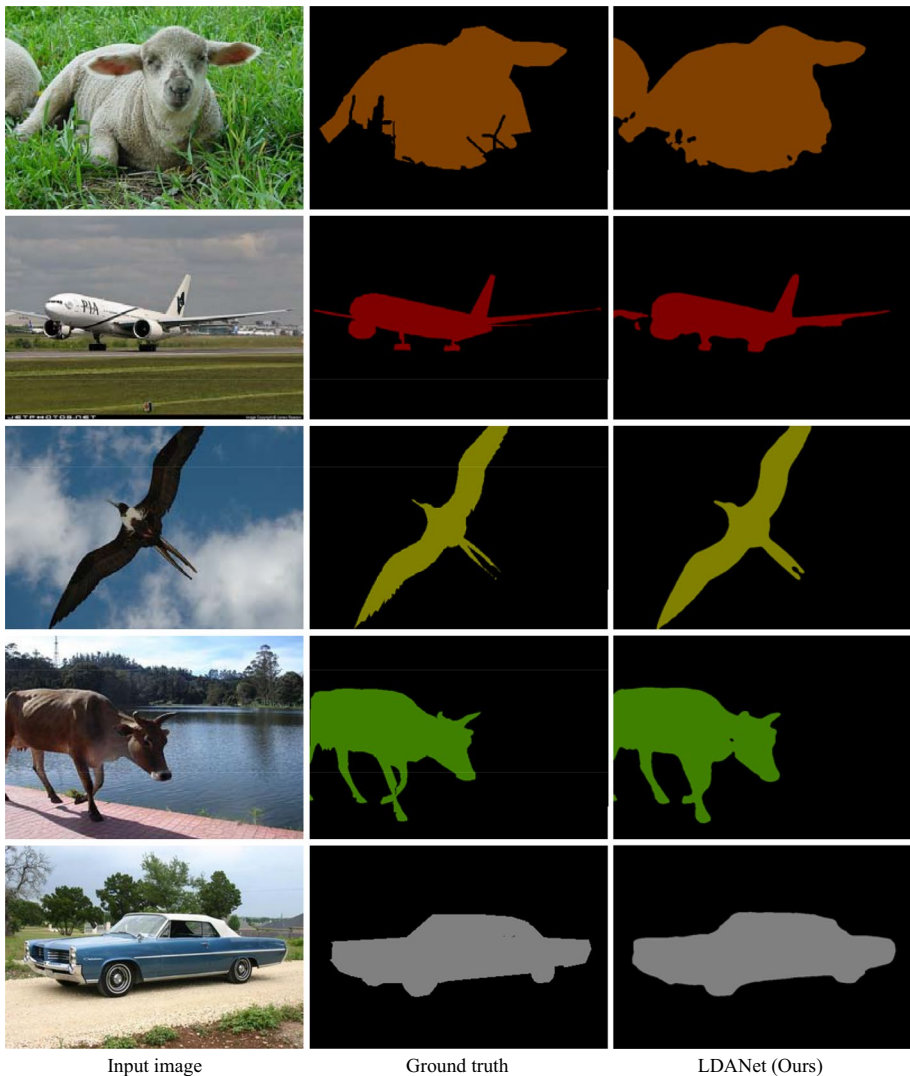


Fig. 7 Visual examples from the PASCAL VOC2012 test set

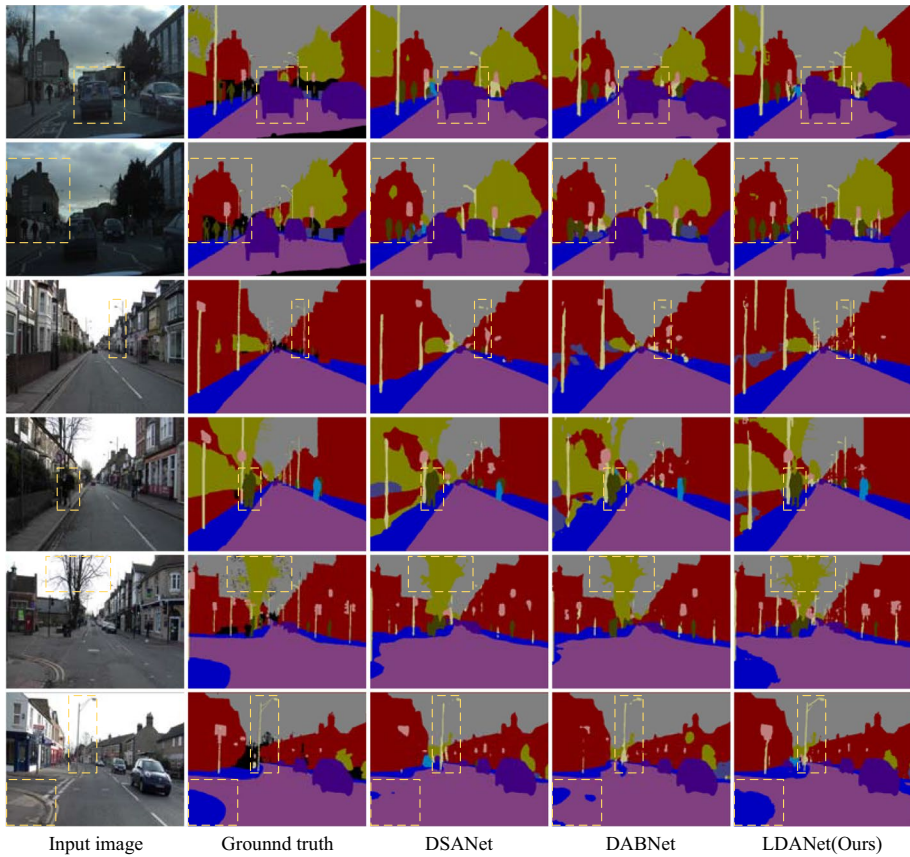


Fig. 8 Visual examples from the Camvid test set

4.4.2 PASCAL VOC2012 and CamVid

The performance of LDANet in the PASCAL VOC2012 and CamVid datasets was tested to verify the generalizability of LDANet in other tasks. The effectiveness of LDANet

Table 6 Segmentation results of 11 classes in the Camvid dataset

Method	Bui	Tre	Sky	Car	Sig	Roa	Ped	Fen	Pol	Sid	Bic	mIoU
SegNet-Basic [1]	80.6	72.0	93	78.5	21.0	94.0	62.5	31.4	36.6	74.0	42.5	46.3
SegNet [1]	88.8	87.3	92.4	82.1	20.5	97.2	57.1	49.3	27.5	84.4	30.7	55.6
ENet [31]	74.7	77.8	95.1	82.4	51.0	95.1	67.2	51.7	35.4	86.7	34.1	51.3
BiSeNet1 [46]	82.2	74.4	91.9	80.8	42.8	93.3	53.8	49.7	25.4	77.3	50.0	65.6
BiSeNet2 [46]	83.0	75.8	92.0	83.7	46.5	94.6	58.8	53.6	31.9	81.4	54.0	68.7
AGLNet [32]	82.6	76.1	91.8	87.0	45.3	95.4	61.5	39.5	39.0	83.1	62.7	69.4
DSANet [10]	84.3	77.8	92.0	86.1	51.1	94.8	62.7	41.8	35.9	82.1	60.1	69.9
Ours	86.8	80.0	93.2	86.8	56.3	94.3	67.2	46.6	44.9	82.9	64.8	73.1

The best performance are highlighted in bold

Table 7 Results of the model in the Camvid dataset and experimental environment

Method	Input size	GPU	Params M	FPS	mIoU(%)	Correctness
SegNet-Basic [1]	360 × 480	Titan XP	1.4	70.0	46.3	82.8
SegNet [1]	360 × 480	Titan XP	29.5	51.0	55.6	88.6
ENet [31]	360 × 480	Titan X	0.36	98.8	51.3	n/a
BiSeNet1 [46]	720 × 960	Titan X	5.80	175.0	65.6	n/a
BiSeNet2 [46]	720 × 960	Titan X	49.00	116.2	68.7	n/a
AGLNet [32]	360 × 480	GTX 1080Ti	1.1	90.1	69.4	n/a
DABNet [24]	360 × 480	GTX 1080Ti	0.76	124.4	66.2	n/a
LRNNet [22]	360 × 480	GTX 1080Ti	0.67	83.0	67.6	n/a
DSANet [10]	360 × 480	GTX 1080	3.47	75.3	69.9	n/a
Ours	720 × 960	RTX 3070	7.74	100.36	73.1	92.9

The best performance are highlighted in bold

was demonstrated by comparing it with realistic semantic segmentation methods.

PASCAL VOC2012: LDANet in the PASCAL VOC2012 dataset was tested, and a mini-batch of 6 and the images were cropped to 512 × 512. The SGD optimizer algorithm was used to train for 300 rounds. A mIoU of 81.1% was achieved at 111.46 fps with the extended dataset. Figure 7 shows the visualization results.

CamVid: LDANet in the CamVid dataset was tested. The mini-batch size was set to 8, and the image size was randomly cropped to 480 × 480. The SGD optimizer algorithm was used to train for 200 rounds. LDANet achieves excellent performance and significantly improves the segmentation of semantic edges (Fig. 8). LDANet can process images with a resolution of 960 × 720 at a speed of 100.36 fps and achieve a mIoU of 73.1% on the CamVid test set. The work selected the pre-training condition, inference speed (fps), parameter quantity, and mIoU on the test set for comparison. Although LDANet's inference speed is not the fastest, LDANet achieved the best results in segmentation accuracy without pre-training. Compared with BiSeNet2 [46], DSANet [10], and AGLNet [32], LDANet was 4.4, 2.2, and 3.7% higher in the mIoU, respectively.

Table 6 shows the IoUs of 11 classes in the Camvid test set. Compared with other networks, LDANet significantly improves several difficult-to-classify classes, which proves the effectiveness of this method. Table 7 presents the experimental environment and results of LDANet and other methods to objectively evaluate the performance of the network model.

5 Conclusions

A new real-time semantic segmentation method was proposed to achieve a balance between speed and accuracy in the work. Efficient semantic segmentation required the fusion of high-level semantic features and detail features. MobileNetV3 was used to extract high-level semantic features from the input, and multi-scale semantic information was obtained through the ASPP structure.

A new efficient decoder network was designed to integrate high-level semantic features and low-level semantic features well and ensure inference speed. It solved the feature loss caused by upsampling. Meanwhile, a boundary constraint branch was established based on second-order differentiation to ensure the segmentation accuracy of semantic boundaries. Specifically, our boundary constraint branch was inserted into other segmentation networks.

The work validated the effectiveness of LDANet in the MaSTr1325 and MID datasets and qualitatively and quantitatively evaluated LDANet in two challenging datasets, CamVid and PASCAL VOC2012. The network was still competitive in other scenarios. Compared with other advanced networks, LDANet significantly improved accuracy, speed, and generalizability.

Funding This work was supported by Major Science and Technology Projects of the Ministry of China Water Resources for the year 2022 (No. SKS2022149); Key Research and Development Plan Projects in Jiangxi Province (No. 20212BBE53028).

Data availability All data included in this study are available upon request by contact with the corresponding author.

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39:2481–2495
2. Bovcon B, Muhovic J, Pers J et al (2019) The MaSTr1325 dataset for training deep USV obstacle detection models. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE
3. Brostow GJ, Fauqueur J, Cipolla R (2009) Semantic object classes in video: A high-definition ground truth database. *Pattern Recogn Lett* 30:88–97
4. Chen L-C, Papandreou G, Kokkinos I et al (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans Pattern Anal Mach Intell* 40:834–848
5. Chen L-C, Papandreou G, Kokkinos I et al (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*. <https://doi.org/10.48550/arXiv.1412.7062>
6. Chen L-C, Papandreou G, Schroff F et al (2017) Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*. <https://doi.org/10.48550/arXiv.1706.05587>
7. Chen L-C, Zhu Y, Papandreou G et al (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 801–818
8. Chollet F (2017) Xception: Deep Learning with Depthwise Separable Convolutions. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1251–1258
9. Dai Y, Gieseke F, Oehmcke S et al (2021) Attentional feature fusion. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp 3560–3569
10. Elhassan M, Huang C, Yang C et al (2021) DSANet: Dilated spatial attention for real-time semantic segmentation in urban street scenes. *Expert Syst Appl* 183:115090
11. Everingham M, Eslami SA, Van Gool L et al (2015) The pascal visual object classes challenge: A retrospective. *Int J Comput Vision* 111:98–136
12. Fan M, Lai S, Huang J et al (2021) Rethinking bisenet for real-time semantic segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 9716–9725

13. Fu J, Liu J, Tian H et al (2019) Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3146–3154
14. Hariharan B, Arbeláez P, Bourdev L et al (2011) Semantic contours from inverse detectors. In: 2011 international conference on computer vision. IEEE, pp 991–998
15. He C, Li K, Zhang Y et al (2023) Camouflaged object detection with feature decomposition and edge reconstruction. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp 22046–22055
16. He C, Li K, Zhang Y, et al (2023) Strategic Preys Make Acute Predators: Enhancing Camouflaged Object Detectors by Generating Camouflaged Objects[J]. arXiv preprint arXiv:2308.03166. <https://doi.org/10.48550/arXiv.2308.03166>
17. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
18. Howard A, Sandler M, Chu G et al (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324
19. Howard AG, Zhu M, Chen B et al (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. <https://doi.org/10.48550/arXiv.1704.04861>
20. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
21. Huang Z, Wang X, Huang L et al (2019) Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 603–612
22. Jiang W, Xie Z, Li Y et al (2020, July) Lrnnnet: A light-weighted network with efficient reduced non-local operation for real-time semantic segmentation. In 2020 IEEE international conference on multimedia & expo workshops (ICMEW), pp 1–6
23. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>
24. Li G, Yun I, Kim J et al (2019) DABNet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. arXiv preprint arXiv:1907.11357. <https://doi.org/10.48550/arXiv.1907.11357>
25. Li H, Xiong P, Fan H et al (2019) Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9522–9531
26. Li P, Ma W (2023) OverSegNet: A convolutional encoder–decoder network for image over-segmentation. *Comput Electr Eng* 107:108610
27. Lin G, Milan A, Shen C et al (2017) Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1925–1934
28. Liu J, Li H, Luo J et al (2021) Efficient obstacle detection based on prior estimation network and spatially constrained mixture model for unmanned surface vehicles. *J Field Robot* 38:212–228
29. 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, pp 3431–3440
30. Ou X, Wang H, Zhang G et al (2023) Semantic segmentation based on double pyramid network with improved global attention mechanism. *Appl Intell* 53:18898–18909. <https://doi.org/10.1007/s10489-023-04463-1>
31. Paszke A, Chaurasia A, Kim S et al (2016) ENet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147. <https://doi.org/10.48550/arXiv.1606.02147>
32. Zhou Q, Wang Y, Fan Y et al (2020) AGLNet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network. *Appl Soft Comput* 96:106682. <https://doi.org/10.1016/j.asoc.2020.106682>
33. Romera E, Alvarez JM, Bergasa LM et al (2017) Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans Intell Transp Syst* 19:263–272
34. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, Springer International Publishing, pp 234–241
35. Sandler M, Howard A, Zhu M et al (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
36. Sze V, Chen Y-H, Yang T-J et al (2017) Efficient processing of deep neural networks: A tutorial and survey. *Proc IEEE* 105:2295–2329
37. Tsai T-H, Tseng Y-W (2023) BiSeNet V3: Bilateral segmentation network with coordinate attention for real-time semantic segmentation. *Neurocomputing* 532:33–42
38. Wang J, Sun K, Cheng T et al (2020) Deep high-resolution representation learning for visual recognition. *IEEE Trans Pattern Anal Mach Intell* 43:3349–3364

39. Wang P, Chen P, Yuan Y et al (2018) Understanding convolution for semantic segmentation. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1451–1460
40. Wen W, Wu C, Wang Y et al (2016) Learning structured sparsity in deep neural networks. *Adv Neural Inf Proces Syst* 29
41. Woo S, Park J, Lee J-Y et al (2018) Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
42. Wu F, Chen F, Jing X-Y et al (2020) Dynamic attention network for semantic segmentation. *Neuro-computing* 384:182–191
43. Wu J, Leng C, Wang Y et al (2016) Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4820–4828
44. Yi Q, Dai G, Shi M et al (2023) ELANet: Effective lightweight attention-guided network for real-time semantic segmentation. *Neural Process Lett* 55:6425–6442. <https://doi.org/10.1007/s11063-023-11145-z>
45. Yu C, Gao C, Wang J et al (2021) Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int J Comput Vision* 129:3051–3068
46. Yu C, Wang J, Peng C et al (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 325–341
47. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122. <https://doi.org/10.48550/arXiv.1511.07122>
48. Yu F, Wang D, Shelhamer E et al (2018) Deep layer aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2403–2412
49. Yuan Y, Huang L, Guo J et al (2018) Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916. <https://doi.org/10.48550/arXiv.1809.00916>
50. Zhang G, Li Z, Li J et al (2023) CFNet: Cascade fusion network for dense prediction. arXiv preprint arXiv:2302.06052. <https://doi.org/10.48550/arXiv.2302.06052>
51. Zhou Q, Qiang Y, Mo Y et al (2022) Banet: Boundary-assistant encoder-decoder network for semantic segmentation. *IEEE Trans Intell Transp Syst* 23:25259–25270

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Zhifang Zhu¹ · Wenhao Wu¹ · Hongzhou Wang²  · Hengyu Li³ · Yibo He¹ · Yuanjie Liu¹ · Quanguo Lu¹ · Xiaohuang Zhan^{4,5}

✉ Hongzhou Wang
15079110193@163.com

¹ Jiangxi Province Key Laboratory of Precision Drive and Control, Nanchang Institute of Technology, Nanchang 330099, China

² College of Intelligent Manufacturing, Jiangxi Technical College of Manufacturing, Nanchang 330095, China

³ School of Mechatronic Engineering and Automation, Shanghai University, 99 Shangda Road, Shanghai, China

⁴ Jiangxi Institute of Mechanical Science, Nanchang 330002, China

⁵ School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China