Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064537

ARTICLE

CGMISeg: Context-Guided Multi-Scale Interactive for Efficient Semantic Segmentation

Ze Wang, Jin Qin, Chuhua Huang* and Yongjun Zhang*

The State Key Laboratory of Public Big Data and College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China

*Corresponding Authors: Chuhua Huang. Email: chhuang@gzu.edu.cn; Yongjun Zhang. Email: zyj6667@126.com

Received: 18 February 2025; Accepted: 19 May 2025; Published: 30 July 2025

ABSTRACT: Semantic segmentation has made significant breakthroughs in various application fields, but achieving both accurate and efficient segmentation with limited computational resources remains a major challenge. To this end, we propose CGMISeg, an efficient semantic segmentation architecture based on a context-guided multi-scale interaction strategy, aiming to significantly reduce computational overhead while maintaining segmentation accuracy. CGMISeg consists of three core components: context-aware attention modulation, feature reconstruction, and cross-information fusion. Context-aware attention modulation is carefully designed to capture key contextual information through channel and spatial attention mechanisms. The feature reconstruction module reconstructs contextual information from different scales, modeling key rectangular areas by capturing critical contextual information in both horizontal and vertical directions, thereby enhancing the focus on foreground features. The cross-information fusion module aims to fuse the reconstructed high-level features with the original low-level features during upsampling, promoting multi-scale interaction and enhancing the model's ability to handle objects at different scales. We extensively evaluated CGMISeg on ADE20K, Cityscapes, and COCO-Stuff, three widely used datasets benchmarks, and the experimental results show that CGMISeg exhibits significant advantages in segmentation performance, computational efficiency, and inference speed, clearly outperforming several mainstream methods, including SegFormer, Feedformer, and SegNext. Specifically, CGMISeg achieves 42.9% mIoU (Mean Intersection over Union) and 15.7 FPS (Frames Per Second) on the ADE20K dataset with 3.8 GFLOPs (Giga Floating-point Operations Per Second), outperforming Feedformer and SegNeXt by 3.7% and 1.8% in mIoU, respectively, while also offering reduced computational complexity and faster inference. CGMISeg strikes an excellent balance between accuracy and efficiency, significantly enhancing both computational and inference performance while maintaining high precision, showcasing exceptional practical value and strong potential for widespread applications.

KEYWORDS: Semantic segmentation; context-aware attention modulation; feature reconstruction; cross-information fusion

1 Introduction

Semantic segmentation is a core task in computer vision and is widely applied in key areas such as autonomous driving, medical image analysis, and remote sensing monitoring. Early methods for semantic segmentation primarily relied on traditional algorithms, including grayscale segmentation, thresholding segmentation, and conditional random fields, among others. However, with ongoing improvements in deep learning techniques, convolutional neural network (CNN)-based approaches, such as FCN [1], DeeplabV3+ [2], and HRNet [3], have made significant progress in segmentation tasks. More recently, the introduction of Transformer-based architectures [4,5] has further boosted segmentation performance. Although progress has been made, most existing approaches still demand considerable computational resources, which hinders their deployment in environments with limited hardware capabilities. Thus, achieving efficient segmentation under such constraints remains a critical challenge.

To address this challenge, recent research has introduced several lightweight, efficient models designed to reduce computational resource consumption while maintaining high segmentation accuracy. These methods typically focus on optimizing model architectures, lowering computational complexity, or employing specific training strategies to enhance inference speed and minimize memory usage. For example, Xu et al. [6] proposed a three-branch architecture that balances speed and performance, allowing real-time semantic segmentation. FeedFormer [7] reengineered the transformer decoder by introducing a feature-enhancing module to replace the self-attention mechanism, achieving high precision and rapid inference. SegNeXt [8] took advantage of multiscale convolutional features to drive spatial attention, demonstrating that simple and efficient convolutional encoders remain highly competitive in both performance and speed.

Although existing lightweight models strike a certain balance between efficiency and accuracy, they fail to fully leverage multi-scale contextual information, limiting their ability to locate foreground objects and optimize boundaries in complex scenes. Furthermore, the lack of effective deep feature synchronization and sharing mechanisms results in an insufficient fusion between low-level details and high-level semantic features, weakening the contextual awareness of multi-scale feature representations and causing boundary-blurring, particularly in the segmentation of complex edges or irregularly shaped objects. In light of these constraints, this paper introduces a Context-Guided Multi-Scale Interaction Semantic Segmentation Network (CGMISeg) to improve segmentation performance while maintaining computational efficiency. CGMISeg incorporates complex contextual relationships and local attention through channel and spatial attention mechanisms, while introducing the designed feature reconstruction and cross-information fusion modules to capture multi-scale contextual dependencies and enable cross-scale interaction. As illustrated in Fig. 1, CGMISeg significantly improves 59 segmentation performance while maintaining low computational complexity, achieving a better 60 trade-off between performance and computational efficiency compared to existing methods. Our main contributions are as follows:

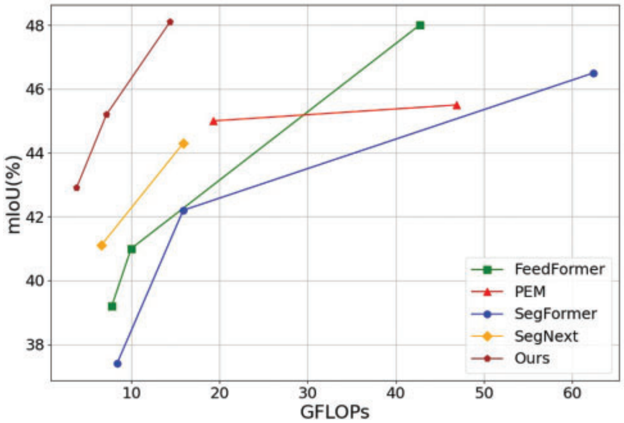

Figure 1: Performance and computational complexity comparison on the ADE20K validation set. Compared to previous methods, our CGMISeg achieves a better balance between performance and computational complexity

1. An efficient context-aware attention modulation is proposed, which dynamically adjusts attention weights in both spatial and channel dimensions. This module captures salient features at different scales, enabling the model to focus more effectively on relevant contexts.

2. A feature reconstruction module is introduced to enhance foreground localization by capturing contextual dependencies across multiple scales.

3. We propose a novel feature fusion method, cross-information fusion, which integrates refined features from the reconstruction module with low-level features from the backbone, enabling effective cross-scale interaction.

4. We construct CGMISeg for 2D semantic segmentation tasks. Compared to current methods, CGMISeg demonstrates superior overall performance, achieving an excellent balance between performance and computational efficiency.

2 Related Work

2.1 Semantic Segmentation

Semantic segmentation is an intensive prediction task in computer vision, aiming to assign a category label to each pixel in an image. The pioneering work of FCN [1] demonstrated that end-to-end pixel-level classification could be achieved using a fully convolutional network, laying the foundation for the application of deep learning in semantic segmentation. Since then, many methods based on convolutional neural network (CNN) have improved FCN from various perspectives. For example, Refs. [3,9] expanded the receptive field by introducing different forms of convolution operations; PSPNet [10] enhanced global scene understanding by collecting multi-scale semantic information; and researchers developed various attention modules [11,12] to improve the model's focus on key features. Recently, transform-based methods have shown tremendous potential. SETR [13] replaces the traditional FCN encoder-decoder structure with a transformer architecture, learning to efficiently capture contextual information through global self-attention. Mask2Former [14] transforms pixel-wise semantic segmentation into mask classification based on the transformer architecture. RTFormer [15] and HRFormer [16] have made further advances in optimizing encoder structures, improving the overall performance of the model. While these approaches have significantly advanced the field and improved task performance, these methods often come with high computational complexity and large parameter counts, which may become bottlenecks in practical applications, limiting their deployment and use on resource-constrained devices.

2.2 Efficient Semantic Segmentation

Efficient semantic segmentation aims to achieve rapid and accurate pixel-level classification in resource-constrained environments, prioritizing the minimization of computational load, parameter numbers, and inference time. CNN-based approaches have been widely adopted for efficient semantic segmentation. These methods often employ specific architectural designs to balance performance and computational cost. For instance, the BiSeNet series [17,18] utilizes a dual-branch architecture and feature fusion strategies to balance performance and efficiency. STDCNet [19] proposes an enhanced framework by rethinking the BiSeNet structure, removing the complex spatial branch and introducing an STDC-guided module to extract multi-scale information. DFANet [20] reduces parameter numbers through sub-network cascading and multi-scale feature propagation, while maintaining a sufficiently large receptive field. SegNeXt [8] employs efficient convolution operators to capture spatial attention, thereby improving segmentation accuracy and optimizing computational efficiency. In parallel, transformer-based methods have emerged as competitive alternatives in efficient semantic segmentation. SeaFormer [21] constructs an efficient backbone network for segmentation by compressing and enhancing an axial transformer, enhancing its cost-effectiveness. SegFormer [22] integrates a transformer architecture with lightweight multi-layer perceptron (MLP) decoders, avoiding the need for complex decoder designs while generating powerful feature representations. Feedformer [7] optimizes computational efficiency by relocating the encoder module to the position of self-attention modules within the decoder. Furthermore, some approaches [23,24] achieve efficient semantic segmentation by reducing learnable parameters and simplifying the training process through pixel-wise clustering or contrastive learning. However, existing methods still encounter challenges in balancing efficiency and accuracy. These challenges can be categorized into three primary areas: (1) Multi-branch architecture issues: Methods like the BiSeNet series, STDCNet, and DFANet enhance receptive fields through multi-branch designs. While effective in expanding receptive fields, the multi-branch structure introduces computational and memory overhead. This overhead hinders further model compression and improvements in inference speed. Furthermore, feature fusion in these architectures faces challenges of semantic inconsistency and scale variations, potentially leading to redundancy or information loss, thus affecting segmentation performance. (2) Attention mechanism limitations: Approaches such as SegNeXt, SeaFormer, and SegFormer leverage lightweight attention mechanisms or Transformers to improve global context understanding. However, the computational cost associated with these mechanisms remains substantial, especially with high-resolution inputs. (3) Contextual imbalance: Many methods struggle to simultaneously capture fine-grained local details and broad global context, which leads to degraded segmentation performance in complex scenes or with indistinct boundaries. To address these issues, we introduce the Context-Aware Attention Modulation (CAAM) and Feature Refinement Module (FRM) to effectively exploit multi-scale contextual dependencies. Furthermore, we propose the Cross Information Fusion Module (CIFM) for efficient cross-layer information integration. Specifically, addressing the multi-branch structure problem, the CIFM module achieves efficient information fusion between different layers. By integrating these lightweight modules, our CGMISeg achieves a balance between performance and computational overhead without requiring extra paths or complex structures.

3 Proposed Methods

Semantic segmentation aims to partition an image into regions with the same semantic features and classify each pixel. Formally, for a given input image $I \in \mathbb{R}^{H \times W \times 3}$, the objective is to generate a segmentation map $S \in \{0, 1, \ldots, K\}^{H \times W}$, where K denotes the number of semantic categories and each element $S(i, j)$ in the matrix represents the category label of the pixel at position (i, j). We follow a classic encoder-decoder architecture to achieve this and design a context-guided multi-scale interactive network for efficient semantic segmentation. This section provides a detailed description of the proposed CGMISeg network structure. Section 3.1 covers the overall architecture, followed by Sections 3.2 and 3.3 discuss contextual feature reconstruction and multi-scale feature interaction.

3.1 Overall Architecture

The overall structure of CGMISeg is shown in Fig. 2a. The backbone network first extracts features from each input image, generating feature maps at scales of 1/4, 1/8, 1/16, and 1/32. The feature maps at scales of 1/8, 1/16, and 1/32 are then fed into the Contextual Feature Reconstruction module, which enhances the overall understanding of the scene context through context-aware attention modulation and utilizes a feature reconstruction module to generate reconstructed features that capture multi-scale context as well as axial global context information. The reconstructed features are subsequently fused with shallow features from the backbone via the Cross-Information Fusion module, thereby enhancing the model's ability to capture fine-grained details and strengthening its overall semantic representation.

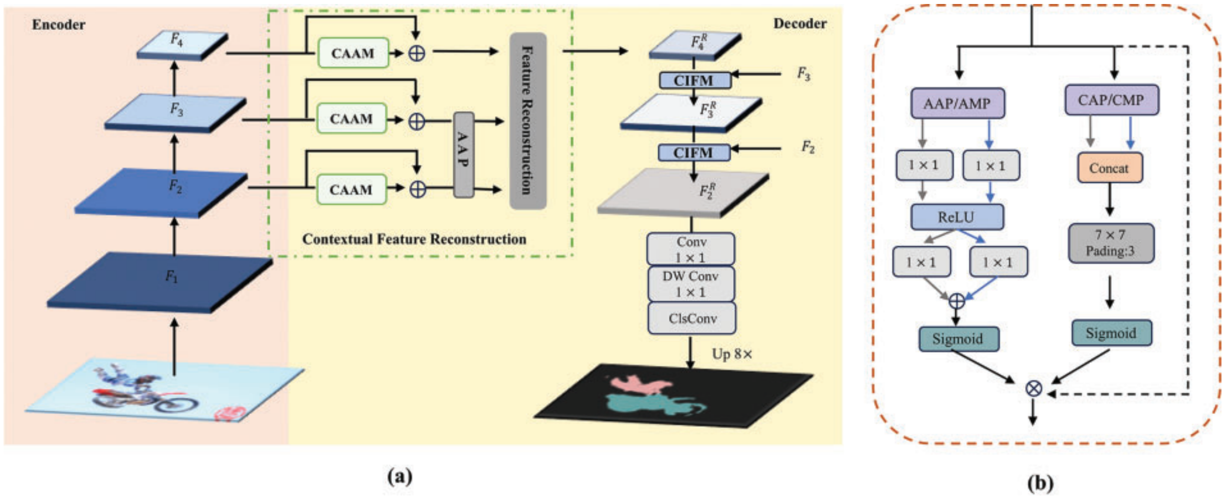

Figure 2: **(a)** The overall architecture of CGIMSeg. **(b)** Illustration of the proposed context-aware attention modulation. Features from different stages, denoted as $[F_2, F_3, F_4]$, are fed into the contextual feature reconstruction module to capture multi-scale contextual information. The reconstructed features are then interactively fused with shallow features from the backbone network via the CIFM, and the outermost reconstructed feature F_2^R is ultimately used to generate the segmentation prediction

3.2 *Contextual Feature Reconstruction*

In semantic segmentation tasks, contextual information always plays a crucial role. Previous studies [25,26] have shown that effectively utilizing contextual information can significantly improve segmentation performance, as it helps the model better understand the scene structure and the relationships between objects, thereby enhancing the accuracy of target object localization and differentiation. In this study, we have designed a contextual feature reconstruction module to capture global contextual information and enhance the focus on foreground features. This module consists of two core components: Context-Aware Attention Modulation (CAAM) and the Feature Reconstruction Module (FRM).

3.2.1 Context-Aware Attention Modulation

Given that the features extracted by the backbone network have strong locality, lack global understanding of the scene context and effective modeling of relationships between local features, we introduce Context-Aware Attention Modulation (CAAM) to enhance feature representation in both channel and spatial dimensions. In Fig. 2b, we present the specific design of CAAM. Specifically, for the four stages of features $F_i \in \mathbb{R}^{H_i \times W_i \times C_i}$, where $i \in \{1, 2, 3, 4\}$, extracted by the encoder through stepwise downsampling, we discard F_1 and select $[F_2, F_3, F_4]$ as input. In CAAM, the input features are processed through two parallel branches. In the first branch, spatially adaptive average and max pooling are employed to distill the most informative spatial features from the input. Each pooled feature is then projected through a 1×1 convolution into a lower-dimensional space:

$$\begin{aligned} F'_{\text{avg},i} &= \text{Proj}(AAP(F_i)) \\ F'_{\text{max},i} &= \text{Proj}(AMP(F_i)) \end{aligned} \tag{1}$$

where $i \in \{2, 3, 4\}$, AAP and AMP represent adaptive average pooling and adaptive max pooling operations, respectively. Proj is a linear transformation implemented by a 1 × 1 convolution, which aims to reduce the channel dimension of the input features to 1/r of the original channel size. Following common channel attention settings [27,28], we set the reduction factor r to 16 to reduce both computational parameters and complexity.

Subsequently, the features are passed through a ReLU activation function and then restored to the original channel dimension via another 1 × 1 convolution. The two restored feature maps are then summed and passed through a Sigmoid activation function to compute the importance weight of each channel:

$$\alpha = \sigma\left(C_1\left(\mathrm{ReLU}\left(F'_{\mathrm{avg},i}\right)\right) + C_2\left(\mathrm{ReLU}\left(F'_{\mathrm{max},i}\right)\right)\right) \tag{2}$$

where C_1 and C_2 represent two independent 1×1 convolutions, and σ denotes the Sigmoid activation function.

In the second branch, global contextual information is extracted by applying both average pooling and max pooling across the channel dimension of the input features. The resulting feature maps are concatenated along the channel dimension and passed through a 7 × 7 convolution, which enhances local context awareness and adjusts the channel dimension to match that of the original input. Finally, we apply a Sigmoid activation function to generate the spatial attention weights:

$$\beta = \sigma(C_{7\times7}([CAP(F_i), CMP(F_i)])) \tag{3}$$

where *CMP* and *CAP* represent the max and average pooling operations performed along the channel dimension, respectively.

After obtaining the channel attention weight α and spatial attention weight β, the context features are computed as:

$$F_i^c = \alpha \odot F_i \odot \beta + F_i \tag{4}$$

where $\odot$ represents element-wise multiplication under the broadcasting mechanism.

3.2.2 Feature Reconstruction Module

After applying context-aware attention modulation, we obtain the contextual features $F_i^c \in \mathbb{R}^{H_i \times W_i \times C_i}$, where $i \in \{2, 3, 4\}$. Subsequently, we introduce a Feature Reconstruction Module (FRM), which not only captures multi-scale contextual information and enhances attention to foreground features but also adopts a lightweight design to address the constraints of limited computational resources. Fig. 3 depicts a graphical representation of FRM.

Specifically, adaptive average pooling is initially used on the low-level features F_2^c and F_3^c, resizing them to align with the resolution of the high-level feature F_4^c:

$$F_i^{c'} = AAP(F_i^c, \mathrm{output\_size}), \quad i = 2, 3 \tag{5}$$

where output_size represents the resolution of the feature F_4^c, denoted as $H_4 \times W_4$. Then, $F_2^{c'}$, $F_3^{c'}$, and F_4^c are fed into the feature reconstruction module. These three different-level features are mapped to a higher-dimensional space through a channel embedding operation. Within this high-dimensional space, the two low-level features are first aggregated through summation and then activated to produce attention weights, which guide the integration with high-level features. This fusion process provides more spatial details to the high-level features, compensating for their lower resolution and lack of detail, thereby enabling the model to accurately identify object contours and details. The computation process can be expressed by the following formulas:

$$(F_2^{c'}, F_3^{c'}, F_4^c) \rightarrow CE(.) \rightarrow (\tilde{F}_2, \tilde{F}_3, \tilde{F}_4)$$

$$F_4' = \tilde{F}_4 \odot \frac{1}{6}\mathrm{ReLU6}(\tilde{F}_2 + \tilde{F}_3) \tag{6}$$

where *CE(.)* denotes the operation of mapping the channel dimension to a higher-dimensional space. Then, a 1×1 convolution is employed to restore F_4' to its initial channel dimension. For the adjusted features, horizontal average pooling P_h and vertical average pooling P_v are applied to capture global context from both directions, resulting in horizontal and vertical vectors. These two vectors are then combined through broadcasting to generate an integrated feature F_4^*. Under the influence of the broadcasting mechanism, the foreground features of the horizontal and vertical vectors are fused, forming a rectangular perception area that enhances the focus on the foreground features. The following formulas illustrate the computation process:

$$\begin{aligned} F_4'' &= \mathrm{Conv}_{1\times1}(F_4') \\ F_4^* &= P_h(F_4'') \oplus P_v(F_4'') \end{aligned} \tag{7}$$

Then, F_4^* is processed through a 3×3 convolution to extract richer local details. Batch normalization and a multi-layer perceptron (MLP) are then applied to refine the features further. The final reconstructed feature F_4^R is obtained by incorporating a residual connection, defined as:

$$F_4^R = \mathrm{MLP}(\mathrm{BN}(\mathrm{Conv}_{3\times3}(F_4^*))) + F_4^c \tag{8}$$

where F_R is the reconstructed feature output by FRM, and MLP refers to a multilayer perceptron consisting of two layers of 1×1 convolutions.

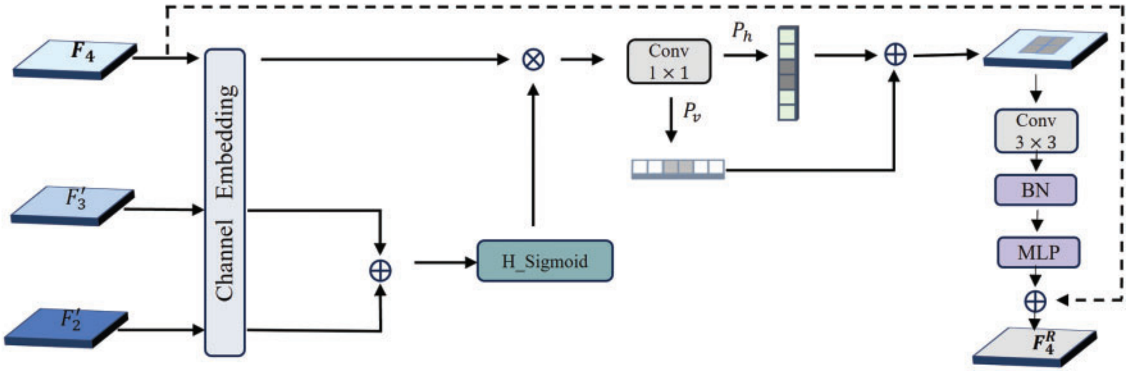

Figure 3: Illustration of the feature reconstruction module (FRM)

3.3 Multi-Scale Feature Interactive

In the contextual feature reconstruction module, we obtain the reconstructed feature F_4^R through context-aware attention modulation and the feature reconstruction module. At this stage, the reconstructed feature is more focused on foreground features under the guidance of contextual information, better reflecting global semantic information. However, directly using this feature for subsequent decoding operations may lead to a loss of foreground feature details or insufficient local information. To further enhance the model's capability, fusing features from different scales is crucial. In light of this, we propose a Cross-Information Fusion Module (CIFM) to facilitate multi-scale feature interaction. The module first performs an initial cross-scale fusion by aligning and adding high-level and low-level features. Subsequently, it extracts local details and global contextual information through separate local and global branches. Next, the local and global features are fused again, and the contribution of low-level and high-level features is dynamically adjusted based on a spatial importance map, ultimately generating the fused feature. In this section, we will detail the design of CIFM and explain how it transforms the reconstructed feature into a pixel-level output containing semantic information.

3.3.1 Cross-Information Fusion Module

The cross-information fusion module is designed to enable effective interaction between features from neighboring stages. As shown in Fig. 4, the module fuses features from high-level and low-level scales. Precisely, we first adjust the high-level features to the same dimensions as the low-level features through convolution and upsampling. Then, the high-level features are directly added to the low-level features for initial feature fusion:

$$\begin{aligned} F_h' &= \mathrm{Up}(\mathrm{Conv}_{1\times1}(F_h)) \\ F_I &= F_l + F_h' \end{aligned} \tag{9}$$

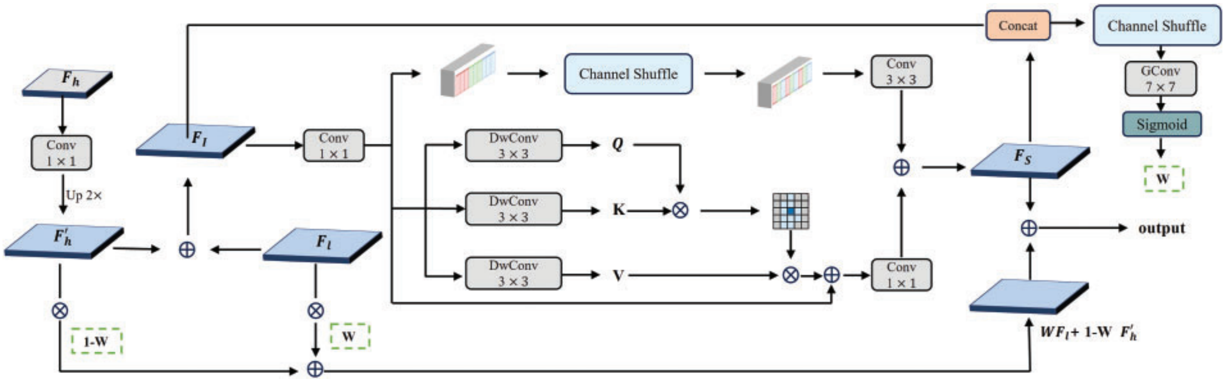

Figure 4: Illustration of the proposed cross-information fusion module (CIFM)

Subsequently, the initial fused feature F_I undergoes a 1 × 1 convolution for channel embedding, adjusting the number of channels to twice its original value to enhance the feature representation ability. The adjusted feature F_I' is then fed into the global and local branches to capture global information and local details, respectively.

In the local branch, the input feature first undergoes a channel shuffle operation, which divides the feature into multiple groups along the channel dimension. Each group is then processed with depthwise separable convolutions to mix and fuse intragroup channel information. After that, a 3 × 3 convolution extracts local features and restores the number of channels to match F_I. The computation of the local branch is formulated as:

$$F_{local} = \mathrm{Conv}_{3\times3}(\mathrm{CS}(F_I)) \tag{10}$$

where F_{local} is the output feature of the local branch, and CS represents the channel shuffle operation.

In the global branch, for the input features, three separate 3 × 3 depthwise separable convolutions are used to generate the Q, K, and V matrices, respectively. The attention map is computed through the interaction between Q and K, and then normalized by Softmax to produce the attention distribution, which is used to weight the information in the V matrix. The result is then combined with the feature F_I' via residual connection, and the original channel dimension is restored using a 1 × 1 convolution. The output of the global branch is defined as:

$$F_{global} = \mathrm{Conv}_{1\times1}\left(V \cdot \mathrm{Softmax}\left(\frac{QK^T}{\gamma}\right)\right) + F_I' \tag{11}$$

where F_{global} denotes the output feature of the global branch, and γ is a learnable scaling parameter used to adjust the magnitude of matrix multiplication.

After obtaining the outputs from both local and global branches, the two features are summed to obtain the secondary fused feature F_S. Subsequently, the resulting secondary fused feature F_S is concatenated with the initial fused feature F_I along the channel dimension. Then, a channel shuffle operation is applied to interleave and reorder the channels of both features, followed by a 7 × 7 grouped convolution to extract inter-channel contextual relationships while restoring the channel dimension to match that of F_S. Finally, a Sigmoid activation function is employed to generate the spatial importance map. The specific computational formula is as follows:

$$W = \sigma(\mathrm{GC}_{7\times7}(\mathrm{CS}([F_I, F_S]))) \tag{12}$$

where σ denotes the Sigmoid activation function, and $\mathrm{GC}_{7\times7}$ represents the 7 × 7 grouped convolution operation.

After obtaining the spatial importance map, we dynamically adjust the contributions of F_l and F_h' based on the spatial importance map, thereby better balancing the fusion of local details and global semantic information. Finally, the final fused feature can be computed as follows:

$$F_{fused} = F_S + W \cdot F_l + (1 - W) \cdot F_h' \tag{13}$$

where F_{fused} denotes the final fused feature output by CIFM, and W serves as the weight coefficient that enables adaptive adjustment of the fusion ratio between low-level and high-level features at different spatial locations, thereby optimizing the feature fusion performance.

3.3.2 Feature Aggregation

Following the contextual feature reconstruction, the reconstructed deep features F_4^R are obtained. We then employ the proposed cross-information fusion module (CIFM) to aggregate the reconstructed features with the shallow features from the backbone network, constructing a feature pyramid network. CIFM fuses information from different scales by inputting the reconstructed deep features and the shallow features from the previous stage of the backbone network for feature fusion, transmitting global semantic information from the deep layers to the shallow layers. Meanwhile, the output of CIFM serves as the reconstructed features for the next stage. The computation process is as follows:

$$F_i^R = \mathrm{CIFM}(F_i, F_{(i+1)}^R), \quad i = 2, 3 \tag{14}$$

In the upsampling feature aggregation stage, we still do not consider the shallowest feature F_1 extracted by the backbone network. This is because the shallowest features typically have higher spatial resolution but are weak in semantic information. Therefore, directly using them for cross-scale feature fusion may introduce noise or overfitting risks to the final prediction results. Instead, we prioritize using intermediate and deep features with richer semantic information, which have aggregated substantial context and help improve the model's segmentation accuracy. Finally, we use the highest-resolution reconstructed feature F_2^R to compute the final prediction. F_2^R undergoes information fusion through feature embedding and a depthwise separable convolution. The fused features are then processed through a classification convolution and upsampling operation to generate the final segmentation map. The computation process is expressed as follows:

$$M = \mathrm{Up}\left(\mathrm{ClassConv}\left(\mathrm{DWconv}\left(CE(F_2^R)\right)\right)\right) \tag{15}$$

where $CE(.)$ is the same as above, representing channel embedding, which adjusts the number of channels using a 1 × 1 convolution layer. DWconv represents the 1 × 1 depthwise separable convolution. ClassConv is the classification convolution, whose output channel number corresponds to the number of segmentation classes. Up represents the upsampling operation. M represents the predicted segmentation mask, which is used during training to compare with the ground truth mask in order to calculate the loss and guide the model's optimization. The calculation formula is as follows:

$$\mathcal{L}_{\mathrm{train}} = \frac{1}{B}\sum \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\ell_{\mathrm{ce}}(M_{ij}, Y_{ij}) \tag{16}$$

where B represents the number of images in the training batch, ℓ_{ce} is the cross-entropy loss, with Y denoting the ground-truth annotations.

4 Experiments

4.1 Experimental Setup

Datasets. We performed extensive experiments on three widely used segmentation datasets–ADE20K [29], Cityscapes [30], and COCO-Stuff [31]–to assess the performance of our CGMISeg. ADE20K contains over 20,000 images from various environments, covering 150 categories, including 100 object categories and 50 background categories. Cityscapes is a segmentation dataset for urban scenes, containing 5000 finely annotated images across 19 categories. COCO-Stuff is built upon the COCO dataset, with pixel-level annotations for 164K images, and includes 172 categories.

Implementation Details. We implemented the code based on the publicly available mmsegmentation library [32] and used EfficientFormerV2 [33] as the encoder backbone. The model variants CGMISeg-T, CGMISeg-B, and CGMISeg-L correspond to EfficientFormerV2-S1, EfficientFormerV2-S2, and EfficientFormerV2-L, respectively. We employed the AdamW optimizer and a Poly learning rate adjustment strategy during training. The feature embedding dimension for CGMISeg-T is 128, while for CGMISeg-B and CGMISeg-L, it is 256. For training, we adopt dataset-specific hyperparameter configurations. On the ADE20K dataset, the learning rate is set to 0.00012, with input images resized to 512 × 512, a batch size of 16, and weight decay of 0.05. The model is trained for a total of 160,000 iterations. For Cityscapes, we use a learning rate of 0.0007 with image dimensions of 1024 × 1024, maintaining the same batch size and weight decay as ADE20K, and training proceeds for 90,000 iterations. On the COCO-Stuff benchmark, the learning rate is adjusted to 0.0001, the weight decay is set to 0.01, input images are sized at 1024 × 512, and a smaller batch size of 8 is used. The training on this dataset is conducted over 10,000 iterations.

4.2 Comparisons with the State-of-the-Art Methods

In this section, we compare our method with various state-of-the-art models on the ADE20K, Cityscapes, and COCO-Stuff. We report the model's performance in three different variants, namely the tiny model, the base model, and the large model. The results are presented in Table 1. For the tiny model, our proposed CGMISeg-T achieves segmentation performance of 42.9% and 41.7% mIoU on ADE20K and COCO-Stuff, respectively, with only 6.3M parameters and 3.8 GFLOPs of computation. For the base model, CGMISeg-B improves upon the popular PEM-STDC2, achieving a 0.2% improvement on the ADE20K dataset (45.2% vs. 45.0%) and a 0.4% improvement on COCO-Stuff (42.6% vs. 42.2%) compared to SegNeXt-S. Although CGMISeg-B has a slightly lower segmentation performance of 80.7% compared to SegNeXt-S's 81.3% on the Cityscapes dataset, CGMISeg-B has fewer parameters and only half the computational cost of SegNeXt-S. Furthermore, for the large-scale model, CGMISeg-L outperforms the previous methods on all three datasets, with mIoU scores of 48.1%, 82.3%, and 45.8%, respectively. CGMISeg-L offers better efficiency in terms of both parameter count and computational cost, with only 27M parameters and computational costs of 14.4 and 121.2 GFLOPs for two different input sizes, which is significantly lower than other models of similar scales. These advantages enable CGMISeg-L to achieve high accuracy with improved computational efficiency and reduced resource consumption, making it ideal for resource-limited applications.

Table 1: Comparison with state-of-the-art methods is performed on the ADE20K, cityscapes, and COCO-Stuff benchmarks. The computational cost in GFLOPs is measured using input resolutions of 512 × 512 for ADE20K and COCO-Stuff, and 2048 × 1024 for cityscapes

Model	**Backbone**	**Params (M)**	**ADE20K**		**Cityscapes**		**COCO-Stuff**	
			GFLOPs	**mIOU**	**GFLOPs**	**mIOU**	**GFLOPs**	**mIOU**
DeepLabV3+ [2]	MobileNetV2	15.4	69.4	34.0	–	–	–	–
Segformer-B0 [22]	MiT-B0	3.8	8.4	37.4	125.5	76.2	8.4	35.6
FeedFormer-B0 [7]	MiT-B0	4.5	7.8	39.2	107.4	77.9	–	–
SegNeXt-T [8]	MSCAN-T	4.3	6.6	41.1	50.5	79.8	6.6	38.7
CGMISeg-T	EFV2-S1	6.3	3.8	42.9	32.8	79.8	3.8	41.7
HRFormer-S [16]	MobileNetV2	13.5	109.5	44.0	835.7	80.0	109.5	37.9
Segformer-B1 [22]	MiT-B1	13.7	15.9	42.2	243.7	78.5	15.9	40.2
SegNeXt-S [8]	MSCAN-S	13.9	15.9	44.3	124.6	81.3	15.9	42.2
FeedFormer-LVT [7]	LVT	4.60	10.0	41.0	124.6	78.6	–	–
PEM-STDC2 [34]	STDC2	21.0	19.3	45.0	118.0	79.0	–	–
CGMISeg-B	EFV2-S2	13.3	7.2	45.2	62.2	80.7	13.3	42.6
Segformer-B2 [22]	MiT-B2	27.5	62.4	46.5	717.1	81.0	62.4	44.6
LRFormer-T [35]	LR-Former	13.0	17.0	46.7	122.0	80.7	17.0	43.9
Mask2Former [14]	Swin-T	47.0	74.0	47.7	–	82.1	–	–
FeedFormer-B2 [7]	MiT-B2	29.1	42.7	48.0	522.7	81.5	–	–
PEM-R50 [34]	ResNet50	35.6	46.9	45.5	240.0	79.9	–	–
CGMISeg-L	EFV2-L	27.0	14.4	48.1	121.2	82.3	13.3	45.8

In addition to comparisons in segmentation performance, parameter count, and computational complexity, we also focus on the inference speed and latency of the proposed lightweight model CGMISeg-T to evaluate its practical potential for deployment on resource-constrained devices. As shown in Table 2, without using any dedicated software or hardware acceleration, CGMISeg-T achieves an inference latency of 34.9 ms and an inference speed of 15.7 FPS when processing 512 × 512 images on a single RTX 2080Ti GPU, outperforming the other four lightweight methods. Under the same testing conditions, DeepLabV3+, Segformer-B0, FeedFormer-B0, and SegNeXt-T all exhibit either higher latency or lower frame rates. For instance, DeepLabV3+ shows an average inference latency of 103.5 ms and a frame rate of 7.2 FPS for the same input size. Although Segformer-B0, FeedFormer-B0, and SegNeXt-T have slightly fewer parameters than CGMISeg-T, they still lag behind in inference speed. In contrast, CGMISeg-T maintains competitive segmentation accuracy while effectively balancing computational resource consumption and inference efficiency, demonstrating superior inference performance and hardware friendliness. These results verify its potential for deployment in edge devices and real-time applications.

Table 2: Comparison with other methods in terms of inference latency and speed

Method	Backbone	Latency (ms)↓	Speed (FPS)↑
DeepLabV3+ [2]	MobileNetV2	103.5	7.2
Segformer-B0 [22]	MiT-B0	57.0	11.0
FeedFormer-B0 [7]	MiT-B0	63.9	9.6
SegNeXt-T [8]	MSCAN-T	44.8	13.5
CGMISeg-T	EFV2-S1	34.9	15.7

4.3 Ablation Studies

Effectiveness of the proposed modules. In this section, we conducted ablation experiments to evaluate the contributions of each component in the proposed method. Table 3 presents the ablation results for the three main components of CGMISeg: CAAM, FRM, and CIFM. The baseline model, shown in the first row of the table, does not include any additional components. In this baseline, we replaced CIFM with a direct upsampling operation of the deep features, followed by element-wise addition to the shallow features, while omitting both CAAM and FRM. As illustrated in the first row, in the absence of additional modules, the final prediction is generated by directly upsampling the deepest features and merging them with early-layer representations, resulting in a segmentation performance of 39.74% mIoU. The baseline model has 5.80 M parameters and a computational complexity of 3.47 GFLOPs. In the second row, after introducing CAAM for context information extraction, the performance improved by 1.30%. CAAM is lightweight, adding only 0.1 M parameters and less than 0.01 GFLOPs to the baseline model. The third row shows the performance after adding FRM to the baseline model. FRM reconstructs features from three stages of the encoder, and the reconstructed features are used for upsampling. This results in a 2.01% performance improvement over the baseline, demonstrating FRM's effectiveness in enhancing multi-scale feature fusion and context information capture. FRM adds only 0.3 M parameters and 0.1 GFLOPs to the baseline, highlighting its balance between performance gains and computational cost. When both CAAM and FRM are combined, the performance improves by 2.67% compared to the baseline. Finally, when CAAM, FRM, and CIFM are all introduced, the model achieves 42.94% mIoU, a significant improvement of 3.2% over the baseline, with 6.26 M parameters and 3.84 GFLOPs. The results indicate that the proposed method yields notable performance gains with minimal additional model complexity. By incorporating lightweight modules, it effectively balances accuracy and computational efficiency.

Table 3: Ablation analysis of CAAM, FRM, and CIFM modules

CAAM	FRM	CIFM	mIoU	GFLOPs	Params
×	×	×	39.74	3.47	5.80 M
✓	×	×	41.04	3.47	5.81 M
×	✓	×	41.75	3.57	6.10 M
✓	✓	×	42.41	3.57	6.11 M
✓	✓	✓	42.94	3.84	6.26 M

The effectiveness of kernel size in contextual feature reconstruction. To evaluate how kernel size influences contextual feature reconstruction, we conducted a set of experiments exploring its effect on overall model performance. First, we explored different kernel sizes for calculating spatial feature attention weights in the CAAM, as shown in Table 4. Increasing the kernel size initially improved segmentation performance, with the best performance achieved using a 7 × 7 kernel. Thereafter, further increasing the kernel size did not significantly improve performance and instead led to a decrease in performance. This may be due to excessive feature fusion caused by larger kernels, which weakened the ability to capture local features. Therefore, we ultimately chose a 7 × 7 kernel to process the fused features, enhancing the local contextual relationships between the features. Additionally, we investigated the effectiveness of kernel size in the feature reconstruction module. Table 5 shows that the model performed best when a 3 × 3 kernel was used to process the integrated features F_4^*. Our analysis suggests that the integrated features obtained through broadcasting already contain global contextual information from horizontal and vertical directions. In this case, using a 3 × 3 kernel effectively combines this global context with local details while maintaining sensitivity to local information. Larger kernels tend to introduce excessive redundant information, reducing the ability to capture fine details and increasing computational overhead. On the other hand, smaller 1 × 1 kernels only operate along the channel dimension and cannot effectively capture spatial information, resulting in poor fusion of local details and global context. Therefore, the 3 × 3 kernel is the ideal choice for this structure.

Table 4: Convolution kernel selection for spatial feature attention calculation in context-aware attention modulation

Kernel size	mIoU (%)
None	41.87
3 × 3	42.51
5 × 5	42.83
7 × 7	42.94
9 × 9	42.59

Table 5: Convolution kernel selection for processing integrated features in the feature reconstruction module

Kernel size	mIoU (%)
None	41.27
1 × 1	42.13
3 × 3	42.94
5 × 5	42.68
7 × 7	42.61

Comparison of performance under different backbones. To validate the performance of our method under different backbone networks, we conducted experiments in two aspects. First, we evaluated the performance of CGMISeg under various lightweight backbone networks to verify the superiority of EfficientFormerV2 as the backbone. As shown in Table 6, we selected backbone networks such as Mix Transformer (MiT) [22], MobileNetV2 [36], MSCAN [8], STDC2 [19], and EfficientFormerV2 (EFV2) [33], all of which have similar computational complexity and parameter scale, ensuring a fair comparison of their impact on the model's performance. The results show significant differences in segmentation performance under different backbone networks. When using MiT-B1 as the backbone, CGMISeg achieved a segmentation performance of 43.24%. In contrast, when using MobileNetV2 and STDC2 as the backbone networks, the performance of CGMISeg increased to 44.30% and 44.68%, respectively. These results indicate that the choice of backbone network plays a key role in determining the final performance of the model. Notably, EfficientFormerV2-S2 achieved the best performance among the five lightweight backbone networks. Specifically, when EfficientFormerV2-S2 was used as the backbone, CGMISeg achieved a segmentation performance of 45.23% with a complexity of only 7.2 GFLOPs. Compared to the other four backbone networks, EfficientFormerV2-S2 exhibited significant advantages in both performance and computational complexity, demonstrating its superiority among several common lightweight backbone networks. Next, we also compared the performance of CGMISeg with larger-scale backbone networks and further analyzed its advantages compared to other methods. Table 7 compares the's performance of CGMISeg with MSCAN-L and MiT-B5 backbone networks with other advanced methods. It can be seen that, when MSCAN-L was used as the backbone, CGMISeg outperformed SegNext by 0.86% in terms of performance while reducing computational complexity by 6.55 GFLOPs. When using MiT-B5 as the backbone, CGMISeg also performed better than SegFormer and FeedFormer. Its segmentation performance was improved by 1.23% and 1.09% compared to SegFormer and FeedFormer, respectively, while also having lower parameter counts and computational complexity than these two methods. This indicates that our method improves accuracy and maintains lower computational resource consumption, offering stronger application potential.

Table 6: Performance comparison under different backbones settings

Method	Backbone	mIoU	GFLOPs	Params (M)
CGMISeg	MobileNetV2	44.30	65.43	**12.95**
	MiT-B1	43.24	15.23	17.17
	MSCAN-S	44.87	14.55	12.71
	STDC2	44.68	18.43	16.72
	EFV2-S2	**45.23**	**7.21**	13.28

Table 7: Performance comparison with other methods under the same backbone configuration

Method	Backbone	mIoU	GFLOPs	Params (M)
SegNext	MSCAN-L	51.01	70.04	48.92
CGMISeg	MSCAN-L	51.87	63.49	52.73
Segformer	MiT-B5	51.83	183.34	84.72
Feedformer	MiT-B5	51.97	187.85	90.17
CGMISeg	MiT-B5	53.06	176.32	73.44

4.4 Qualitative Results

To intuitively evaluate the superiority of CGMISeg, visual segmentation results on the ADE20K and Cityscapes test sets are presented in Figs. 5 and 6, respectively, in comparison with two established lightweight semantic segmentation networks, Segformer and SegNext. To facilitate detailed comparison, key regions exhibiting performance variations are highlighted with yellow bounding boxes. Fig. 5 illustrates the segmentation results on the ADE20K dataset. As shown in the first row, Segformer misidentifies pixels on the desktop as part of the adjacent sofa. Similarly, SegNext exhibits category confusion in the highlighted region, misclassifying the floor area. Conversely, CGMISeg avoids these errors, yielding more accurate segmentation boundaries. In the second and third rows, CGMISeg demonstrates enhanced recognition accuracy and superior boundary preservation for objects such as baskets, washing machines, and tables within the highlighted regions, noticeably outperforming Segformer and SegNext. These results underscore its robustness and ability to delineate fine-grained details within complex scenes. To further assess the generalization capabilities of CGMISeg in urban street environments, Fig. 6 presents visual segmentation results on the Cityscapes dataset. Multiple challenging regions are highlighted. As observed, CGMISeg exhibits superior performance on targets such as traffic signs, motorcycles, sidewalks, and fences compared to Segformer and SegNext. For instance, in the first row, Segformer demonstrates misclassification between grass and sidewalks, incorrectly labeling portions of the sidewalk as grass and misidentifying distant motorcycles as cars, leading to category confusion. Concurrently, SegNext exhibits blurry segmentation around the boundaries of traffic signs, failing to accurately capture object contours and compromising the structural integrity of the scene. In contrast, CGMISeg accurately differentiates between sidewalks and grass areas while successfully preserving the contours of motorcycles and traffic signs, showcasing enhanced segmentation precision and improved boundary retention.

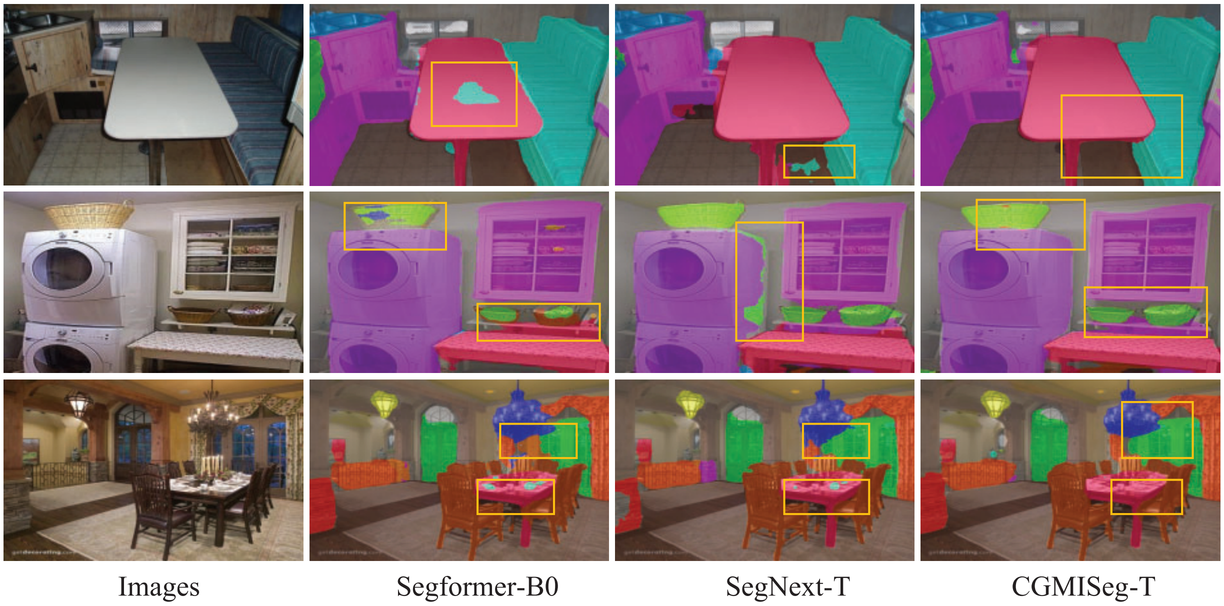

Figure 5: Qualitative comparison of CGMISeg-T on the ADE20K dataset

In addition to evaluations on the standard ADE20K and Cityscapes test sets, we further conducted visual tests of CGMISeg on real-world street view images to assess its generalization capability in complex real-world scenarios. Fig. 7 presents a comparison of segmentation results among CGMISeg, Segformer, and Feedformer on multiple real street view images, using model weights trained on the Cityscapes dataset. As shown in the yellow-highlighted regions, CGMISeg demonstrates higher accuracy and clearer boundary delineation for key targets such as fences, buildings, roads, and sidewalks compared to the other two methods. This further validates CGMISeg's superior structural awareness and category discrimination ability when faced with realistic and complex environments. Benefiting from its context-aware mechanism and multi-scale feature fusion strategy, CGMISeg effectively mitigates category confusion in regions with overlapping semantics, improves boundary completeness, and preserves fine-grained details, demonstrating strong potential for real-world deployment.

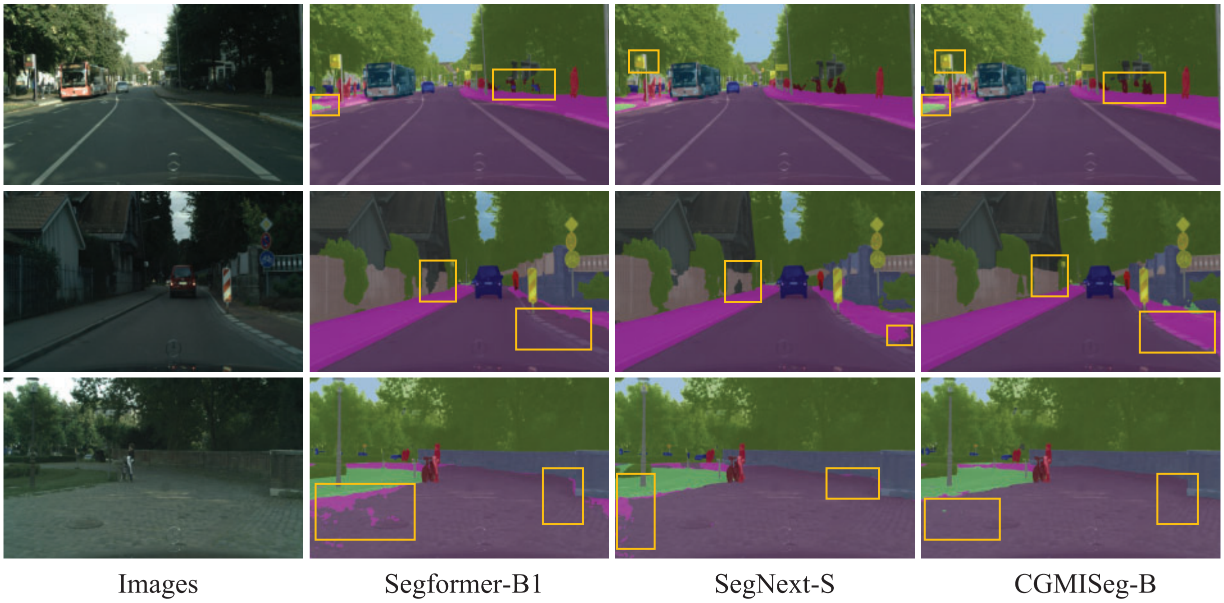

Figure 6: Qualitative comparison of CGMISeg-B on the cityscapes dataset

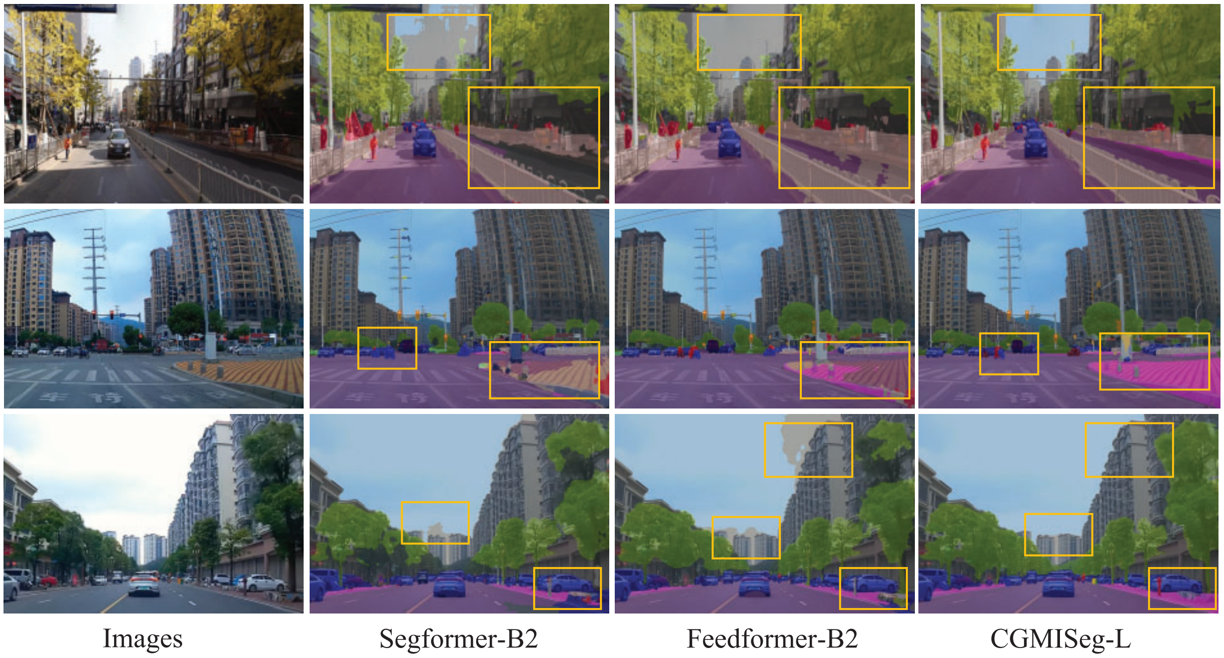

Figure 7: Qualitative comparison of CGMISeg-L on real-world street-view images

5 Limitations and Future Work

Although CGMISeg demonstrates notable advances in computational efficiency and lightweight design, certain limitations warrant further investigation. First, the model's reliance on GPU (Graphics Processing Unit) inference presents obstacles for real-time deployment on resource-constrained embedded systems. To address this, future research will focus on architectural optimizations, specifically employing more efficient attention mechanisms and post-training quantization methods. Post-training quantization can mitigate the GPU dependency by reducing the model's memory footprint and computational demands, thereby enabling deployment on devices with limited processing capabilities. Second, experiments focus exclusively on publicly available natural scene datasets. While promising, the model's generalization capabilities require further validation across cross-domain applications in medical and geospatial imaging. Furthermore, the model's performance degrades under challenging conditions characterized by high inter-class similarity, significant scale variations, extreme lighting, or heavy occlusions. This is particularly evident in boundary region delineation and small object segmentation, areas where accuracy requires improvement. Future work will also explore dynamic computation allocation and conditional execution mechanisms to improve robustness in these complex scenarios. Finally, while we aim to facilitate real-world deployment, future work is needed to investigate broader hardware compatibility.

6 Conclusion

In this paper, we propose CGMISeg, a context-guided multi-scale interactive semantic segmentation network that offers excellent performance for semantic segmentation. CGMISeg consists of three key modules: CAAM, FRM, and CIFM. Firstly, the CAAM dynamically adjusts attention weights in both spatial and channel dimensions to capture rich global contextual information. Secondly, FRM enhances the model's focus on foreground features through multi-scale contextual information fusion and rectangular region modeling. Finally, CIFM facilitates efficient multi-scale feature interaction during the upsampling process, leveraging feature information from each layer of the encoder to optimize boundary details and semantic consistency. Comprehensive evaluations on three benchmark datasets–ADE20K, Cityscapes, and COCO-Stuff–demonstrate that CGMISeg delivers outstanding segmentation performance, achieving an optimal balance between computational cost and segmentation accuracy compared with existing methods.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Natural Science Foundation of China (62162007) and the Guizhou Provincial Basic Research Program (Natural Science) (No. QianKeHeJiChu-ZK[2024]YiBan079).

Author Contributions: Ze Wang: Research design, manuscript drafting. Jin Qin: Data analysis and interpretation, manuscript revision, and methodology. Chuhua Huang: Technical and financial support. Yongjun Zhang: Project supervision and final manuscript revision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The public datasets used in the research are all available and have been cited in the references.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Long Jonathan, Shelhamer Evan, Darrell Trevor. Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell. 2017;39(4):640–51. doi:10.1109/TPAMI.2016.2572683.
2. Chen L, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer Vision–ECCV 2018 (ECCV 2018). Cham, Switzerland: Springer; 2018. p. 833–51. doi: 10.1007/978-3-030-01234-2_49.
3. Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, et al. Deep high-resolution representation learning for visual recognition. IEEE Trans Pattern Anal Mach Intell. 2020;43(10):3349–64. doi:10.1109/TPAMI.2020.2983686.

4. Cheng Bowen, Schwing Alex, Kirillov Alexander. Per-pixel classification is not all you need for semantic segmentation. Adv Neural Inf Process Syst. 2021;34:17864–75. doi:10.5555/3540261.3541628.
5. Thisanke H, Deshan C, Chamith K, Seneviratne S, Vidanaarachchi R, Herath D. Semantic segmentation using vision transformers: a survey. Eng Appl Artif Intell. 2023;126(4):106669. doi:10.1016/j.engappai.2023.106669.
6. Xu JC, Xiong ZX, Bhattacharyya SP. PIDNet: a real-time semantic segmentation network inspired by PID controllers. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023; Vancouver, BC, Canada. p. 19529–39. doi:10.1109/CVPR52729.2023.01871.
7. Shim J, Yu H, Kong K, Kang SJ. FeedFormer: revisiting transformer decoder for efficient semantic segmentation. Proc AAAI Conf Artif Intell. 2023;37(2):2263–71. doi:10.1609/aaai.v37i2.25321.
8. Guo Meng-Hao, Lu C, Hou Q, Liu Z, Cheng M, Hu S. SegNeXt: rethinking convolutional attention design for semantic segmentation. In: NIPS'22: 36th International Conference on Neural Information Processing Systems; 2022 Nov 28–Dec 9; New Orleans, LA, USA. p. 1140–56. doi:10.5555/3600270.3600354.
9. Chen Liang-Chieh. Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587. 2017. doi:10.48550/arXiv.1706.05587.
10. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017; Honolulu, HI, USA. p. 6230–9. doi:10.1109/CVPR.2017.660.
11. Zhong Z, Lin Z, Bidart R, Hu X, Daya IB, Li Z, et al. Squeeze-and-attention networks for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020; Seattle, WA, USA. p. 13062–71. doi:10.1109/CVPR42600.2020.01308.
12. Huang Z, Wang X, Huang L, Huang C, Wei Y, Liu W. CCNet: criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019; Seoul, Republic of Korea. p. 603–12. doi:10.1109/ICCV.2019.00069.
13. Zheng Sixiao, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021; Nashville, TN, USA. p. 6877–86. doi:10.1109/CVPR46437.2021.00681.
14. Cheng B, Misra I, Schwing AG, Kirillov A, Girdhar R. Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022; New Orleans, LA, USA. p. 1280–9. doi:10.1109/CVPR52688.2022.00135.
15. Wang J, Gou C, Wu Q, Feng H, Han J, Ding E, et al. RTFormer: efficient design for real-time semantic segmentation with transformer. In: NIPS'22: 36th International Conference on Neural Information Processing Systems; 2022 Nov 28–Dec 9; New Orleans, LA, USA. p. 7423–36. doi:10.5555/3600270.3600809.
16. Yuan Y, Fu R, Huang L, Lin W, Zhang C, Chen X, et al. HRFormer: high-resolution vision transformer for dense predict. Adv Neural Inf Process Syst. 2021;34:7281–93. doi:10.5555/3540261.3540818.
17. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N. BiSeNet: bilateral segmentation network for real-time semantic segmentation. In: Computer Vision–ECCV 2018 (ECCV 2018). Cham, Switzerland: Springer; 2018. p. 334–49. doi:10.1007/978-3-030-01261-8_20.
18. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N. BiSeNet V2: bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vis. 2021;129(11):3051–68. doi:10.1007/s11263-021-01515-2.
19. Fan M, Lai S, Huang J, Wei X, Chai Z, Luo J, et al. Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021; Nashville, TN, USA. p. 9711–20. doi:10.1109/CVPR46437.2021.00959.
20. Li H, Xiong P, Fan H, Sun J. DFANet: deep feature aggregation for real-time semantic segmentation. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019; Long Beach, CA, USA. p. 9514–23. doi:10.1109/CVPR.2019.00975.
21. Wan Q, Huang Z, Lu J, Yu G, Zhang L. SeaFormer: squeeze-enhanced axial transformer for mobile semantic segmentation. In: The Eleventh International Conference on Learning Representations; 2023 May 1–5; Kigali, Rwanda. doi:10.1007/s11263-025-02345-2.
22. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P. SegFormer: simple and efficient design for semantic segmentation with transformers. Adv Neural Inf Process Syst. 2021;34:12077–90. doi:10.5555/3540261.3541185.

23. Liang J, Zhou T, Liu D, Wang W. Clustseg: clustering for universal segmentation. In: Proceedings of the 40th International Conference on MachineLearning, ICML'23; 2023 Jul 23–29; Honolulu, HI, USA. p. 20787–809. doi:10.5555/3618408.3619265.
24. Wang W, Zhou T, Yu F, Dai J, Konukoglu E, Gool LV. Exploring cross-image pixel contrast for semantic segmentation. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision; 2021; Montreal, QC, Canada. p. 7283–93. doi:10.1109/ICCV48922.2021.00721.
25. Yuan YH, Chen XL, Wang JD. Object-contextual representations for semantic segmentation. In: Computer Vision-ECCV 2020: 16th European Conference; 2020 Aug 23–28; Glasgow, UK. p. 173–90. doi:10.1007/978-3-030-58539-6_11.
26. Shi M, Lin S, Yi Q, Weng J, Luo A, Zhou Y. Lightweight context-aware network using partial-channel transformation for real-time semantic segmentation. IEEE Trans Intell Transp Syst. 2024;25(7):7401–16. doi:10.1109/TITS.2023.3348631.
27. Cao Y, Xu J, Lin S, Wei F, Hu H. Gcnet: non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops; 2019; Seoul, Republic of Korea. p. 1971–80. doi:10.1109/ICCVW.2019.00246.
28. Hou Qibin, Zhou Daquan, Feng Jiashi. Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021; Nashville, TN, USA. p. 13713–22. doi:10.1109/CVPR46437.2021.01350.
29. Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, Torralba A. Scene parsing through ADE20K dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017; Honolulu, HI, USA. p. 5122–30. doi:10.1109/CVPR.2017.544.
30. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, et al. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; Las Vegas, NV, USA. p. 3213–23. doi:10.1109/CVPR.2016.350.
31. Caesar Holger, Uijlings Jasper, Ferrari Vittorio. Coco-stuff: thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. p. 1209–18. doi:10.1109/CVPR.2018.00132.
32. Contributors MMS. MMSegmentation: OpenMMLab semantic segmentation toolbox and Benchmark. 2020." 2023 [Internet] [cited 2025 May 18]. Available from: https://github.com/open-mmlab/mmsegmentation.
33. Li Y, Hu J, Wen Y, Evangelidis G, Salahi K, Wang Y, et al. Rethinking vision transformers for mobilenet size and speed. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023; Paris, France. p. 16843–54. doi:10.1109/ICCV51070.2023.01549.
34. Cavagnero N, Rosi G, Cuttano C, Pistilli F, Ciccone M, Averta G, et al. PEM: prototype-based efficient mask-former for image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024; Seattle, WA, USA. p. 15804–13. doi:10.1109/CVPR52733.2024.01496.
35. Wu Y, Zhang S, Liu Y, Zhang L, Zhan X, Zhou D, et al. Low-resolution self-attention for semantic segmentation. arXiv:2310.05026. 2023. doi:10.48550/arXiv.2310.05026.
36. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. p. 4510–20. doi:10.1109/CVPR.2018.00474.