Academic Journal of Applied Sciences
ISSN: 3105-0549 (Print), ISSN: 3105-0557 (Online) | Vol. 1, No. 1, 2026

SFENet: A Real-Time Semantic Segmentation Network Based on Selective State Models and Frequency-Edge Enhancement

Kaiyu Zhou*

School of Software, Henan Polytechnic University, 2001 Century Avenue, Jiaozuo 454000, China
*** Corresponding author**

Abstract: Existing real-time semantic segmentation models have achieved notable progress. However, they still suffer from feature redundancy and insufficient capability in capturing spatial details. To address these issues, this paper proposes a real-time semantic segmentation network based on selective state models and frequency-domain spatial detail enhancement. Specifically, a Selective Redundancy Suppression Module (SRSM) is designed based on a selective scanning mechanism. This module retains informative features while suppressing redundant ones through selective scanning. In addition, a Frequency Edge Enhancement Module (FEEM) is constructed. It combines the Fast Fourier Transform (FFT) with an attention mechanism to enhance high-frequency edge information. SFENet is evaluated on two datasets. On the Cityscapes dataset, it achieves 77.3% mIoU, outperforming the baseline by 1.4% mIoU. On the CamVid dataset, it achieves 77.1% mIoU. Experimental results demonstrate that SFENet effectively preserves critical semantic features and accurately segments object boundaries.

Keywords: Real-Time Semantic Segmentation; Selective Scanning; Edge Enhancement.

1. Introduction

Semantic segmentation achieves precise object delineation by performing pixel-level classification, and it has significant application value in fields such as intelligent transportation and smart healthcare. Early semantic segmentation networks were primarily built upon Convolutional Neural Networks (CNNs). Since the introduction of Fully Convolutional Networks (FCNs) [1], a variety of semantic segmentation models have been proposed, including SegNet [2], the DeepLab series (V1 [3], V2 [4], V3 [5], V3+ [6]), RefineNet[7], and PSPNet [8]. These methods mainly improve segmentation performance by enhancing feature fusion, enlarging the receptive field, and strengthening global contextual modeling. However, they typically focus on improving accuracy while paying less attention to inference speed and computational efficiency. As a result, they may not achieve satisfactory performance in practical applications.

With the increasing demands of practical applications, model efficiency and computational cost have gradually become key concerns in semantic segmentation research. To address this, several lightweight networks have been proposed to improve processing speed. For example, ENet [9] and ICNet[10] achieve real-time segmentation to a certain extent, while CGNet[11] further reduces computational complexity. However, in pursuing higher efficiency, these methods often inevitably lead to a decline in segmentation accuracy.

In recent years, to balance accuracy and efficiency, researchers have begun to explore real-time semantic segmentation methods based on dual-branch or multi-branch architectures. These approaches process spatial details and semantic information in separate branches, thereby significantly improving inference efficiency while maintaining accuracy. BiSeNet [12] first introduced a structure that combines a spatial branch and a semantic branch, achieving a good balance between speed and performance. Building upon this design, STDC [13] simplifies the architecture by removing certain branches to further improve inference speed. DDRNet [14] adopts a dual-resolution branch design and enhances feature representation through multiple bidirectional fusion operations, improving segmentation accuracy while maintaining high efficiency. RDRNet [15], based on DDRNet, introduces a re-parameterization strategy, employing a multi-path structure during training and converting it into a single-path structure during inference, thereby effectively improving runtime efficiency. PIDNet [16] constructs a three-branch framework consisting of a detail branch, a context branch, and a boundary branch. This design preserves fine-grained details while enhancing contextual modeling, further improving segmentation performance.

Although the aforementioned methods achieve a certain balance between speed and accuracy, they inevitably introduce redundant information during feature fusion. Meanwhile, their ability to capture boundary details remains limited, making it difficult to fully represent fine-grained structures.

To address the above issues, this paper proposes a real-time semantic segmentation network based on selective state models and frequency-edge enhancement, termed SFENet. Inspired by the aforementioned dual-branch architectures, SFENet is also constructed using a dual-branch framework. To mitigate information redundancy during feature fusion, a selective scanning mechanism is introduced to perform selective feature extraction. To improve the representation of boundary details, a frequency edge enhancement module is designed by incorporating the Fourier transform and an attention mechanism:

(1) A Selective Redundancy Suppression Module (SRSM) is proposed. Based on Spatial Mamba, geometric transformations are incorporated, and the original processing path is divided into multiple shorter paths. This design preserves the advantages of multi-path modeling while

reducing computational cost and alleviating the forgetting issue caused by excessively long sequences.

(2) A Frequency Edge Enhancement Module (FEEM) is constructed. It employs frequency decomposition and attention-based enhancement to strengthen boundary information. This design significantly enhances the high-frequency edge components in feature representations, thereby improving the clarity of segmentation boundaries.

(3) Extensive experiments are conducted on the Cityscapes and CamVid datasets to validate the effectiveness of SFENet. On multiple A100 GPUs, SFENet achieves an impressive 77.3% mIoU and a fast inference speed of 110.4 FPS on the Cityscapes dataset. On the CamVid test set, it attains 77.1% mIoU and an inference speed of 135.6 FPS. These results demonstrate that SFENet achieves strong performance in real-time semantic segmentation.

2. Related Work

(1) Dual-branch network structure

Existing studies have shown that multi-branch architectures are effective in enhancing global contextual modeling. Among them, DDRNet is a representative approach. It constructs a dual-resolution branch architecture, where the high-resolution branch preserves spatial detail information, while the low-resolution branch focuses on extracting semantic features. Building upon this design, RDRNet further introduces a re-parameterization strategy to improve inference efficiency. The dual-branch semantic segmentation framework adopted by RDRNet, as shown in Fig. 1, consists of a Detail Branch and a Semantic Branch. Both branches are composed of stacked re-parameterization modules, which combine 3×3 and 1×1 convolutional kernels. In addition, a re-parameterizable pyramid pooling module (RPPM) is designed at the end of the network to enhance multi-scale feature extraction. Furthermore, an auxiliary segmentation head is introduced to further optimize overall performance.

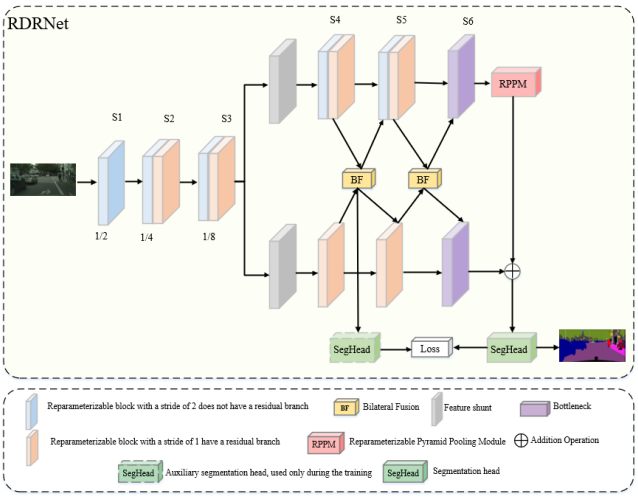

Figure 1. Overall architecture of RDRNet

From an overall perspective, the input image first undergoes a series of feature extraction operations and is progressively downsampled to 1/2, 1/4, and 1/8 of the original resolution. Once the feature map at 1/8 scale is obtained, it is fed simultaneously into the spatial branch and the semantic branch for further processing. In the semantic branch, feature maps are progressively downsampled through re-parameterization modules at stages S4, S5, and S6, reducing their resolutions to 1/16, 1/32, and 1/64 of the original image, respectively. In contrast, the spatial branch maintains a resolution of 1/8 throughout, in order to preserve rich spatial details. To enable effective interaction between spatial and semantic information, fusion modules are introduced at stages S4 to S6, where features from both branches are integrated multiple times. After completing stage S6, the feature map at 1/64 resolution is fed into the RPPM module for multi-scale feature modeling. Subsequently, the output of the RPPM is upsampled by a factor of 8 and fused with the output of the spatial branch. The fused features are then passed to the segmentation head to generate the final predictions. Meanwhile, an auxiliary segmentation head is incorporated to provide additional supervision during training, thereby improving overall segmentation performance.

(2) VMamba

Traditional semantic segmentation models based on Convolutional Neural Networks (CNNs) and Transformers each exhibit certain limitations. Moreover, during feature fusion, integrating information from multiple receptive fields often leads to feature redundancy. With the introduction of VMamba[17], it becomes possible to perform long-range dependency modeling with relatively low computational cost while alleviating feature redundancy. VMamba extends state space models, originally designed for sequential data modeling, to the visual domain. To balance the ordered nature of one-dimensional sequence scanning with the spatial structure of two-dimensional image data, VMamba introduces a two-dimensional selective scanning module, namely Selective Scan 2D (SS2D). This module traverses image features along four distinct paths, enabling ordered modeling of two-dimensional data and facilitating effective global information aggregation. The overall architecture of VMamba, as illustrated in Fig. 2, consists of four stages and mainly includes two key components: a downsampling module and a Visual State Space (VSS) module.

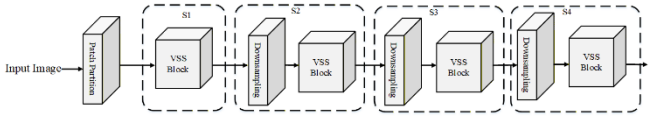

Figure 2. Structure of the VMamba block

3. Method

In this section, we first introduce the overall architecture of SFENet, followed by detailed descriptions of the SRSM and FEEM modules.

3.1. Overall Architecture of SFENet

Based on a dual-branch architecture, this paper proposes a real-time semantic segmentation network, termed SFENet, which integrates selective state modeling and frequency-edge enhancement. The overall architecture is illustrated in Figure. 3. The network consists of six stages, denoted as S1–S6. Stages S1–S3 are primarily composed of residual blocks and are responsible for initial feature extraction. In stages S4–S6, the backbone is divided into two substructures: a semantic branch and a spatial branch. In the semantic branch, these stages incorporate two types of residual blocks along with Bottleneck Blocks (BB) to enhance high-level semantic representation. In contrast, the spatial branch adopts a uniform residual block structure and maintains the feature map resolution, thereby preserving rich fine-grained spatial details. To facilitate effective interaction between semantic

and spatial information, a Bilateral Fusion (BF) module is introduced between the two branches. In stage S6, both branches further incorporate Bottleneck Blocks (BB) to enhance feature representation. To address the issue of feature redundancy during fusion, a Selective Redundancy Suppression Module (SRSM) is introduced between stages S4 and S5. This module selectively retains informative features while suppressing redundant ones. In addition, considering the importance of high-frequency edge information in semantic segmentation, a Frequency Edge Enhancement Module (FEEM) is introduced after the Reparameterizable Pyramid Pooling Module (RPPM) to strengthen edge feature representation. Furthermore, following the design of the baseline network, an auxiliary segmentation head is added alongside the main segmentation head to provide additional supervision during training, thereby improving overall segmentation performance.

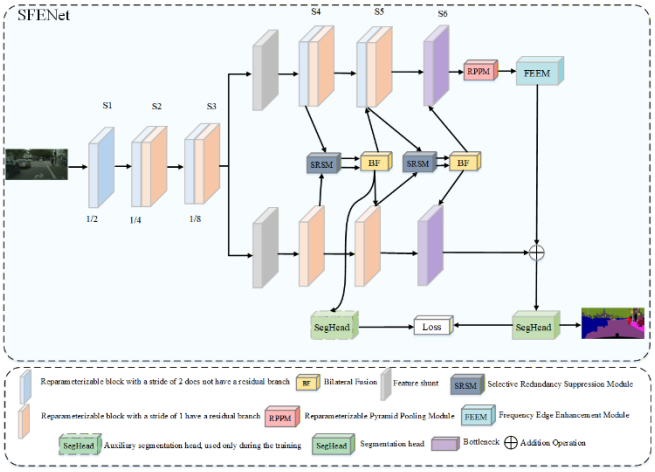

Figure 3. Overall architecture of SFENet

The input image first passes through stages S1–S3 for rapid feature extraction, producing feature maps with resolutions of 1/2, 1/4, and 1/8 of the original image, respectively. The feature map at 1/8 resolution is then fed into two branches, namely the Detail Branch and the Semantic Branch, for further processing. In the Semantic Branch, stages S4 and S5 adopt Reparameterizable Blocks (RB) with a stride of 2, progressively reducing the feature map resolution to 1/16 and 1/32 of the original image. For feature maps at these two scales, they are first passed through the Selective Redundancy Suppression Module (SRSM) for feature selection, and then fed into the Bilateral Fusion (BF) module for feature integration. At stage S6, a Bottleneck Block (BB) with a stride of 2 is applied to further downsample the feature map to 1/64 of the original resolution. The resulting feature map is then input into the Reparameterizable Pyramid Pooling Module (RPPM) to capture deeper and richer semantic information. After processing by the RPPM, the outputs of the semantic branch and the spatial branch are fused (via element-wise addition). The fused features are then upsampled by a factor of 8 to restore spatial resolution and passed to the segmentation head to generate the final prediction. Meanwhile, an auxiliary segmentation head is introduced during training to enhance feature learning and improve overall network performance. The detailed parameter configuration of the network is provided in Table 1, where RB denotes the Reparameterizable Block and BB denotes the Bottleneck Block.

Table 1. Detailed structure of SFENet.

Stage	Semantic Branch				Spatial-Details Branch			
	OPR	S	R	C	OPR	S	R	C
Input				3				3
S1	RB	2	1	32	RB	2	1	32
S2	RB	2 1	1 4	32	RB	2 1	1 4	32
S3	RB	2 1	1 3	64	RB	2 1	1 3	64
S4	RB	2 1	1 5	128	RB	1	4	64
S5	RB	2 1	1 5	256	RB	1	4	64
S6	BB	2	1	512	BB	1	1	128

3.2. Selective Redundancy Suppression Module (SRSM)

The selective state scanning mechanism in state space models (Mamba) demonstrates strong capability in feature selection. Inspired by this, the Selective Redundancy Suppression Module (SRSM) is designed to alleviate feature redundancy during feature fusion. The overall structure of SRSM is illustrated in Figure 4. The module mainly consists of multi-path scanning, state updating, and state fusion. By controlling parameters that regulate the retention of historical states and the scope of representation, the module enhances important features while suppressing redundant ones.

The workflow of SRSM is as follows. First, the input feature map is flattened into a sequence, converting the original two-dimensional representation into a one-dimensional sequence to facilitate subsequent sequence modeling. Next, selective state updates are performed on the flattened sequence, during which historical states are dynamically selected. Finally, the outputs from multiple scanning paths are fused to obtain the final representation.

The above process can be formulated as follows:

Feature Flattening. For the input feature $X \in \mathbb{R}^{B\times H\times W\times d}$, the two-dimensional feature map is transformed into a one-dimensional feature representation:

$$X \in R^{B\times H\times W\times d}, X_f = Reshape(X) \in R^{B\times d\times L}, \quad L = H \times W \tag{1}$$

Dynamic Parameter Generation. Dynamic parameters are generated from the input features, including the step size, input projection, and output projection:

$$\Delta_t = Linear_{\Delta}(x_t), B_t = Linear_B(x_t), C_t = Linear_C(x_t) \tag{2}$$

Selective State Update. The selective state update mechanism dynamically regulates historical information:

$$h_t = \overline{A_t} h_{t-1} + \overline{B_t} \tag{3}$$

Continuous-to-Discrete State Mapping. To ensure stability on discrete sequences, the continuous state-space parameters are discretized:

$$\overline{A_t} = exp(\Delta_t A), \overline{B_t} = (\Delta_t A)^{-1}(exp(\Delta_t A) - I) \bullet \Delta_t B_t \tag{4}$$

State-to-Output Mapping. After updating the hidden states, the output features are obtained through a linear mapping:

$$y_t = C_t h_t \tag{5}$$

Multi-Path Selective Modeling. To enhance the representational capacity of the model, multi-path scanning is introduced:

$$y^{(v)} = \{y_1^{(v)}, y_2^{(v)}, y_3^{(v)}, \ldots, y_L^{(v)}\}, v = 1,2,3,\ldots,V \tag{6}$$

Feature Fusion. The outputs from multiple paths are fused to obtain the final representation:

$$y = Merge\left(y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(V)}\right) \quad (7)$$

Here, B denotes the batch size, while H and W represent the spatial dimensions, and d denotes the number of channels. Δ_t represents the time step. B_t and C_t are used for input projection and output projection, respectively. denotes the current hidden state. $\bar{A}_t$ controls the retention ratio of historical states, while $\bar{B}_t$ regulates the injection strength of the current input. A is a learnable state matrix, and I denotes the identity matrix. Merge$(\cdot)$ represents the fusion operation.

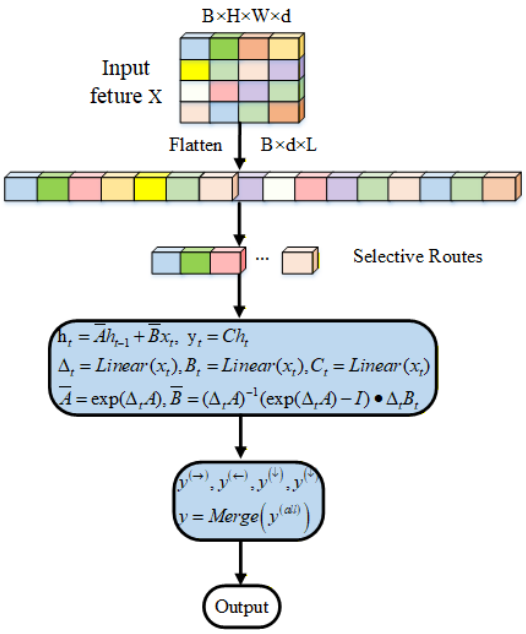

Figure 4. Detailed Structure of the SRSM

3.3. Frequency Edge Enhancement Module (FEEM)

To address the insufficient boundary modeling of traditional convolutions and conventional attention modules, the FEEM module is proposed. Its overall architecture is illustrated in Figure 5. FEEM consists of two main components: a frequency-domain enhancement branch and a spatial residual main branch. The spatial residual main branch is primarily used to preserve the original semantic information, while the frequency-domain enhancement branch is designed to extract and selectively emphasize high-frequency edge information. The main workflow is described as follows.

First, the Fast Fourier Transform (FFT) is applied to map the input features from the spatial domain to the frequency domain, where high-frequency components associated with edges are identified. Next, an attention mechanism is employed to enhance these high-frequency features. Finally, the enhanced frequency-domain information is transformed back to the spatial domain and fused with the original features via residual connections, thereby achieving edge enhancement. The above process can be expressed as follows:

The input features are first fed into the module and mapped to the frequency domain via the Fourier Transform:

$$X \in R^{B\times C\times H\times W}, \quad Z = f(X) \quad (8)$$

After entering the frequency domain, the spectrum is decomposed into amplitude and phase components:

$$A = |Z|, P = \angle Z \quad (9)$$

After spectral decomposition, high-frequency components are selectively filtered to extract edge-related information while suppressing low-frequency components:

$$A_h = M_h \odot A \quad (10)$$

After extracting the high-frequency information, channel attention is applied to enhance these features:

$$W_c = \sigma(MLP(GAP(A_h)))A_h' = W_c \odot A_h \quad (11)$$

After channel-wise selection, the high-frequency features are further refined through spatial enhancement:

$$W_s = \sigma(Conv_{3\times3}(A_h'))\hat{A}_h = (1+W) \odot A_h' \quad (12)$$

The enhanced features are then recombined with the original phase to reconstruct the frequency-domain representation:

$$\hat{Z}_h = \hat{A}_h \bullet e^{jP} \quad (13)$$

The enhanced high-frequency components in the frequency domain are transformed back into the conventional two-dimensional feature map representation:

$$F_h^{spa} = f^{-1}(\hat{Z}_h) \quad (14)$$

A 1×1 convolution is applied for channel alignment, and the result is fused with the original input via a residual connection to obtain the final output:

$$Y = X + Conv_{1\times1}(F_h^{spa}) \quad (15)$$

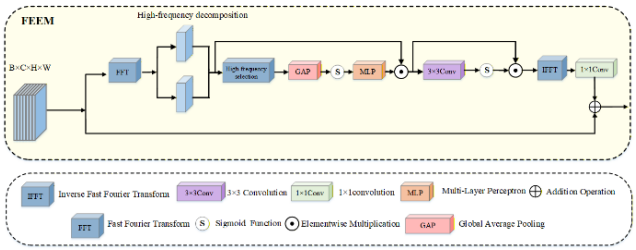

Figure 5. Detailed architecture of FEEM

4. Experiments

4.1. Datasets

In this study, two representative semantic segmentation datasets, Cityscapes and CamVid, are selected to evaluate the proposed model. Example samples are shown in Figures 6 and 7. Cityscapes is primarily designed for urban road scene understanding. It contains approximately 5,000 high-resolution street-view images with a resolution of 1024×2048, covering 19 semantic categories. According to the official split, the dataset includes 2,975 images for training, 500 for validation, and 1,525 for testing. CamVid also focuses on traffic scene analysis and consists of 701 street-view images captured in road environments in Cambridge, UK. The images have a resolution of 720×960 and are annotated with 11 semantic categories. The dataset is divided into 367 training images, 101 validation images, and 233 test images. These two datasets differ in urban scenarios, category definitions, and annotation protocols, enabling a comprehensive evaluation of the model's segmentation performance and generalization ability from multiple perspectives.

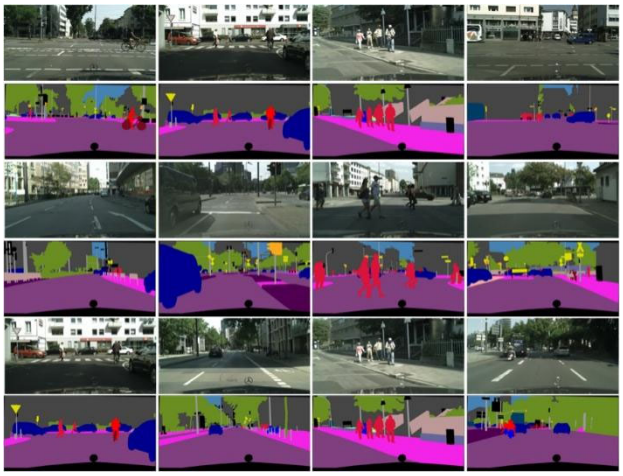

Figure 6. Example images from the Cityscapes dataset

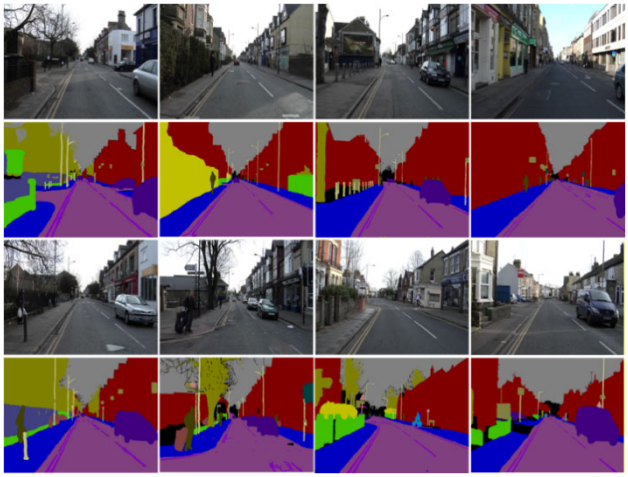

Figure 7. Example images from the CamVid dataset

4.2. Implementation Details and Evaluation

Training Settings. In this study, model training was conducted on the Cityscapes and CamVid datasets, respectively. All experiments were implemented within the MMSegmentation framework, and no pretrained weights from ImageNet were used during training. The detailed training hyperparameter settings are listed in Table 2.

Table 2. Detailed training settings

Dataset	Cityscapes	CamVid
Optimizer	SGD (Momentum=0.9)	
Weight decay	1×10^{-4}	
Initial learning rate	0.01	
Learning rate policy	Poly (power=0.9)	
Batch size	6 (on 2 A100 GPUs, 12 on 1 GPU)	
Loss function	OhemCrossEntropy	
Train Iterations	120000 iterations	7800 iterations
Crop Size	1024 × 1024	960 × 720
Image Resize Range	[0.5 ×, 2.0 ×] of original	

Inference Settings. The detailed configurations for the model inference stage are presented in Table 3. The inference procedures and speed evaluations on the three datasets are conducted under a unified software environment, which consists of CUDA 11.3, PyTorch 1.12.1, MMSegmentation 1.0.0, and Anaconda.

Table 3. Detailed inference settings

Dataset	Cityscapes	CamVid
Batch size	1 on 1 A100 GPU	
Crop Size	1024 × 2048	960 × 720

Evaluation. During the model evaluation stage, Intersection over Union (IoU) and mean Intersection over Union (mIoU) are commonly used to measure the accuracy of segmentation results. In addition, to assess the computational efficiency of the model, GFLOPs (billions of floating-point operations) and the number of model parameters are also introduced as reference metrics.

Intersection over Union (IoU) is used to quantify the degree of overlap between the predicted results and the ground truth. It is defined as the ratio of their intersection to their union. The calculation is given as follows:

$$IoU = \frac{TP}{TP+FP+FN} \tag{16}$$

Here, TP (True Positive) denotes the number of pixels correctly classified as the target class; FP (False Positive) denotes the number of pixels incorrectly predicted as the target class; and FN (False Negative) denotes the number of pixels that belong to the target class but are not correctly identified. A higher IoU value indicates better segmentation performance of the model.

Mean Intersection over Union (mIoU) is an extension of IoU. defined as the average IoU across all classes. Its calculation is given as follows:

$$mIoU = \frac{1}{k}\sum_{i=0}^{k-1}\frac{TP}{TP_i+FP_i+FN_i} \tag{17}$$

Here, $k$denotes the total number of classes. For the i-th class, TP, FP, and FN represent the numbers of correctly predicted pixels, misclassified pixels, and missed pixels, respectively. Therefore, mIoU provides an overall measure of model performance in multi-class segmentation tasks.

4.3. Comparison with State-of-the-Art Methods

4.3.1. Comparison on Cityscapes Dataset

Table 4 presents a comparison of SFENet with other state-of-the-art methods on the Cityscapes dataset. The evaluated approaches include CNN-based real-time segmentation models such as ERFNet[20], ICNet[10], Fast-SCNN[21], MSFNet[22], SwiftNetRN-18[23], SwiftNetRN-18 ens[23], CGNet[11], BiSeNetV1[12], BiSeNetV2[24], STDC1[13], STDC2[13],HyperSeg-M[25], PP-LiteSeg-T2[26], DDRNet-23-Slim[14], PIDNet-S[16], and RDRNet-S-Simple[15]. The table reports key information for each method, including input image resolution, inference speed (FPS), number of parameters, and whether ImageNet pre-trained weights are used during training. From the results in Table 4, SFENet shows competitive performance. It achieves 77.3% mIoU at an inference speed of 110.4 FPS. Compared with other advanced real-time semantic segmentation networks, SFENet attains competitive segmentation performance without increasing computational cost and while maintaining comparable inference speed. Compared with earlier real-time models such as BiSeNet and STDC, SFENet shows clear advantages. BiSeNet preserves high-resolution feature maps to retain fine details, which leads to higher computational cost. STDC removes the spatial branch to improve efficiency, but this results in a drop in accuracy. In contrast, SFENet achieves a better balance between accuracy and computational cost.

When compared with the baseline model RDRNet-S-Simple, SFENet exhibits only a slight reduction in inference speed, but surpasses it in segmentation accuracy due to enhanced boundary modeling. Compared with PIDNet-S, which adopts a three-branch architecture and thus incurs higher computational cost and lower inference speed, SFENet achieves better segmentation accuracy. This improvement is mainly attributed to the introduction of the Mamba selective scanning mechanism during feature fusion, which effectively enhances feature representation.

Table 4. Comparison with state-of-the-art methods on the Cityscapes dataset.

Method	FPS	Params	ImageNet	mIoU (%)
ERFNet[20]	33.1	2.1M	×	72.5
ICNet[10]	101.1	24.9M	×	71.6
Fast-SCNN[21]	192.0	1.4M	×	71.0
MSFNet[22]	41.0	-	×	-
SwiftNetRN-18[23]	39.9	11.8M	✓	75.4
SwiftNetRN-18 ens[23]	18.4	24.7M	✓	-
CGNet[11]	54.9	0.5M	×	68.3
BiSeNetV1[12]	65.9	13.3M	✓	74.4
BiSeNetV2[24]	74.4	3.4M	×	73.6
STDC1[13]	98.4	8.3M	×	71.8
STDC2[13]	74.7	12.3M	×	74.9
HyperSeg-M[25]	59.1	10.1M	✓	76.2
PP-LiteSeg-T2[26]	96.0	-	✓	76.0
DDRNet-23-Slim[14]	129.4	5.7M	×	75.9
DDRNet-23[14]	54.6	20.3M	×	78.0
PIDNet-S[16]	101.5	7.7M	×	76.3
RDRNet-S-Simple[15]	131.2	7.2M	×	75.9
SFENet	110.4	8.6M	×	77.3

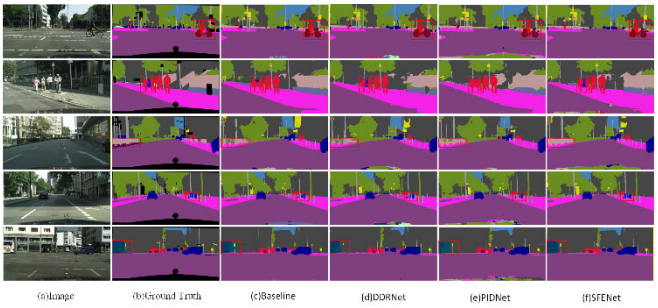

Figure 8. Visual comparison with state-of-the-art methods on the Cityscapes dataset.

Figure 8 presents a visual comparison of SFENet on the Cityscapes dataset. The figure includes the original images, ground-truth labels, and the segmentation outputs of four networks: the baseline model, DDRNet, PIDNet, and SFENet. The red rectangles highlight regions where performance improvements are observed. As shown in Figure 8, SFENet achieves higher segmentation quality in boundary regions, particularly in complex edge scenarios and in accurately delineating the contours of objects such as pedestrians and buses. In contrast, PIDNet and DDRNet exhibit limitations in boundary modeling, with some regions showing missing or incomplete edges. Moreover, in cases involving occlusion or overlapping objects, SFENet maintains more reliable segmentation results. This advantage is mainly attributed to the introduced selective scanning mechanism, which effectively enhances critical feature representations while suppressing redundant and interfering information.

4.3.2. Comparison on CamVid Dataset

Table 5 presents a comparative analysis of SFENet on the CamVid dataset, using several mainstream real-time semantic segmentation methods as references, including ERFNet[20], ICNet[10], Fast-SCNN[21], CGNet[11], BiSeNetV1[12], BiSeNetV2[24], STDC1[13], STDC2[13], DDRNet-23-Slim[14], PIDNet-S[16], and RDRNet-S-Simple[15]. The comparison focuses on two aspects: inference speed and mean Intersection over Union (mIoU). In addition, the table indicates whether each method utilizes ImageNet pre-trained weights. As shown in Table 5, SFENet achieves superior segmentation performance compared with state-of-the-art methods. Although the introduction of additional functional modules leads to a slight reduction in inference speed compared with RDRNet-S-Simple, SFENet attains a notable improvement in segmentation accuracy. Compared with DDRNet-23-Slim, SFENet demonstrates better segmentation performance under comparable inference speed. Furthermore, relative to earlier methods such as BiSeNet and STDC, SFENet benefits from a more efficient network architecture, enhanced edge modeling, and the incorporation of a feature selection mechanism during feature fusion, which collectively improve the overall segmentation performance.

Table 5. Comparison with state-of-the-art methods on the CamVid dataset

Method	ImageNet	FPS	mIoU (%)
ERFNet[20]	×	71.2	72.4
ICNet[10]	×	173.7	65.7
Fast-SCNN[21]	×	260.6	66.2
CGNet[11]	×	99.5	69.4
BiSeNetV1[12]	✓	140.2	73.3
BiSeNetV2[24]	×	156.9	75.8
STDC1[13]	×	170.2	70.2
STDC2[13]	×	114	71.4
DDRNet-23-Slim[14]	×	166.7	76.2
PIDNet-S[16]	×	124.9	77.0
RDRNet-S-Simple[15]	×	175.1	76.1
SFENet	×	135.6	77.1

Figure 9 presents a visual comparison of segmentation results on the CamVid dataset among SFENet, the advanced networks DDRNet and PIDNet, and a baseline model. Specifically, Figure 9(a) and Figure 9(b) correspond to the original images and ground-truth labels, respectively. From the results, it can be observed that PIDNet and DDRNet exhibit incomplete segmentation of environmental boundaries in complex road-edge scenarios. In contrast, SFENet produces clearer and more complete segmentation along guardrails and human boundaries, owing to the incorporation of the frequency-based edge enhancement module. Furthermore, in cases where pedestrians overlap with walls, DDRNet and PIDNet suffer from incomplete segmentation and category misclassification, whereas these

issues are alleviated in SFENet. This advantage primarily stems from the introduced Mamba selective scanning mechanism, which suppresses redundant information while preserving critical feature representations.

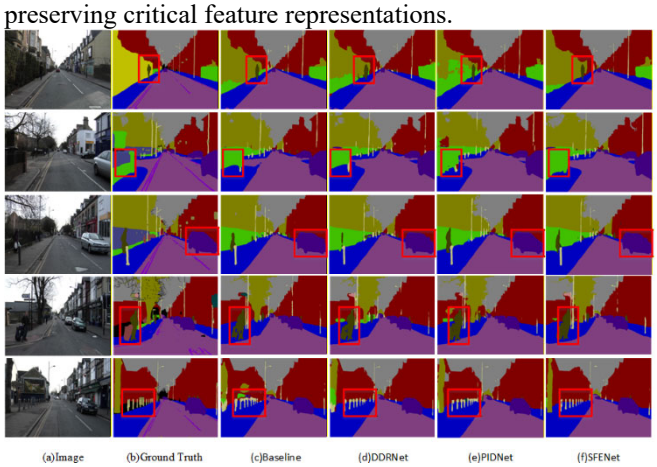

Figure 9. Visual comparison with state-of-the-art methods on the CamVid dataset

5. Conclusion

This paper addresses the issues of information redundancy in feature fusion and insufficient attention to boundary details by proposing a real-time semantic segmentation network, termed SFENet, which integrates selective state modeling with frequency-based edge enhancement. The method introduces the selective scanning mechanism from Mamba to effectively alleviate redundancy during the feature fusion stage. In addition, to compensate for the limitations of previous approaches in boundary modeling, a combination of frequency decomposition and attention enhancement is employed, thereby improving the completeness and clarity of segmentation results in edge regions. Comparative experiments with several state-of-the-art methods demonstrate the superior performance of SFENet in semantic segmentation tasks. However, there remains room for improvement in global feature modeling and the utilization of frequency information. Future work will focus on incorporating more powerful global modeling strategies and further exploring and enhancing frequency-domain features.

References

[1] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

[2] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.

[3] Chen L C. Semantic image segmentation with deep convolutional nets and fully connected CRFs[J]. arXiv preprint arXiv:1412.7062, 2014.

[4] Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 834-848.

[5] Chen L C. Rethinking atrous convolution for semantic image segmentation[J]. arXiv preprint arXiv:1706.05587, 2017.

[6] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.

[7] Lin G, Milan A, Shen C, et al. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1925-1934.

[8] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2881-2890.

[9] Paszke A, Chaurasia A, Kim S, et al. Enet: A deep neural network architecture for real-time semantic segmentation[J]. arXiv preprint arXiv:1606.02147, 2016.

[10] Zhao H, Qi X, Shen X, et al. Icnet for real-time semantic segmentation on high-resolution images[C]//Proceedings of the European conference on computer vision (ECCV).

[11] Wu T, Tang S, Zhang R, et al. Cgnet: A light-weight context guided network for semantic segmentation[J]. IEEE Transactions on Image Processing, 2020, 30: 1169-1179.

[12] Yu C, Wang J, Peng C,etal.Bisenet:Bilateral segmentation network for real-time semantic segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 325-341.

[13] Fan M, Lai S, Huang J, et al. Rethinking bisenet for real-time semantic segmentation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 9716-9725.

[14] Hong Y, Pan H, Sun W, et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes[J]. arxiv preprint arxiv:2101.06085, 2021.

[15] Yang G, Wang Y, Shi D. Reparameterizable Dual-Resolution Network for Real-time Semantic Segmentation[J]. arxiv preprint arxiv:2406.12496, 2024.

[16] XU J, XIONG Z, BHATTACHARYYA S P.PIDNet: A real-time semantic segmentation network inspired by PID controllers[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 19529-19539.

[17] Liu Y, Tian Y, Zhao Y, et al. Vmamba: Visual state space model[J]. Advances in neural information processing systems, 2024, 37: 103031-103063.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213– 3223.

[19] G. J. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” Pattern recognition letters, vol. 30, no. 2, pp. 88–97, 2009.

[20] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, “Erfnet: Efficient residual factorized convnet for real-time semantic segmentation,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 263–272, 2017.

[21] R. P. Poudel, S. Liwicki, and R. Cipolla, “Fast-scnn: Fast semantic segmentation network,” arXiv preprint arXiv: 1902. 04502, 2019.

[22] H. Si, Z. Zhang, F. Lv, G. Yu, and F. Lu, “Real-time semantic segmentation via multiply spatial fusion network,” arXiv preprint arXiv:1911.07217, 2019.

[23] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, “In defense of pretrained imagenet architectures for real-time semantic segmentation of road-driving images,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 607–12 616.

[24] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation,” International Journal of Computer Vision, vol. 129, pp. 3051–3068, 2021.

[25] Y. Nirkin, L. Wolf, and T. Hassner, “Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4061–4070.

[26] J. Peng, Y. Liu, S. Tang, Y. Hao, L. Chu, G. Chen, Z. Wu, Z. Chen, Z. Yu, Y. Du et al., “Pp-liteseg: A superior real-time semantic segmentation model,” arXiv preprint arXiv:2204.02681, 2022.