WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 5, Number 3, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v5n3.03
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Real-Time Semantic Segmentation: A Comprehensive Review and Future Perspectives

Meng Gao *

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, P R China

*Corresponding Author: gaomeng@home.hpu.edu.cn

ABSTRACT

As a fundamental task in image perception and understanding, semantic segmentation has been extensively applied across various domains, including medical image processing, scene analysis, autonomous driving perception, and intelligent video analytics. Practical implementations often prioritize real-time semantic segmentation due to constraints in computational resources, interaction requirements, and cost considerations. To facilitate researchers' efficient comprehension of algorithmic design and applications in this field, this paper conducts a comprehensive review and analysis of deep learning-based real-time semantic segmentation methods. Specifically, 1) Fundamental concepts, application scenarios, and challenges of semantic segmentation and its real-time variant are introduced. 2) Essential techniques and design paradigms for real-time semantic segmentation algorithms are systematically elucidated. 3) State-of-the-art real-time semantic segmentation approaches are thoroughly categorized and summarized. 4) Practical application scenarios of real-time semantic segmentation are discussed. 5) A complete evaluation framework with standardized metrics is established. 6) Conclusions are drawn along with critical analysis of remaining challenges while proposing insightful perspectives for future research directions.

KEYWORDS

Semantic segmentation; Image perception; Real-time

1. INTRODUCTION

Semantic segmentation, as a fundamental pixel-level perception and understanding task in computer vision, aims to assign category labels to each pixel in an input image. This task plays a pivotal role in diverse real-world applications, including medical image processing [1–3], robotic vision [4], remote sensing classification [5–7], augmented reality [8], autonomous driving [9–11], and intelligent video analytics [12].

Prior to the advent of deep learning, semantic segmentation primarily relied on traditional image segmentation methods such as region-based and boundary-based techniques, including Otsu's method [13], K-means clustering [14], watershed algorithms [15], region growing [16], active contours [17], graph cuts [18], conditional random fields (CRFs) [19], and Markov random fields (MRFs) [20]. The emergence of deep learning significantly enhanced segmentation performance, catalyzing advancements in practical applications. The introduction of the Fully Convolutional Network (FCN) [21] marked a paradigm shift, establishing convolutional neural networks (CNNs) [22] as the dominant framework for image segmentation.

Despite the breakthroughs achieved by FCN and subsequent CNN-based approaches, semantic segmentation faces inherent challenges: recovering information lost during downsampling and

effectively leveraging multi-scale features and long-range contextual dependencies to improve accuracy. To address these challenges, researchers have proposed diverse strategies for capturing contextual information, such as expanding receptive fields, multi-scale feature fusion, and self-attention mechanisms. These efforts have led to three main research directions: 1) advanced backbone architectures (e.g., VGG [23], GoogLeNet [24], ResNet [25], HRNet [26]), 2) context modeling frameworks (e.g., U-Net [27], PSPNet [28], RefineNet [29]), and 3) attention-enhanced CNN designs.

The introduction of Vision Transformers (ViT) [30] by Dosovitskiy et al. revolutionized the field by integrating Transformer mechanisms from natural language processing into computer vision. Pioneering works like SETR [31] demonstrated ViT's potential for segmentation, while PVT [32] incorporated pyramid structures into Transformer-based models. SegFormer [33] further proposed an efficient multi-scale Transformer architecture, and Swin Transformer [34] emerged as a state-of-the-art backbone across vision tasks, surpassing CNN counterparts. Although these methods achieve remarkable accuracy, they incur substantial computational overhead—particularly due to the quadratic complexity of self-attention mechanisms relative to image resolution—resulting in prohibitive inference latency for real-time applications.

Real-world scenarios such as mobile computing, autonomous driving, and human-computer interaction demand real-time segmentation capabilities. To address this need, researchers have developed efficient deep learning architectures that balance accuracy and speed. A real-time semantic segmentation network is typically defined as one achieving ≥30 FPS (the minimum frame rate for smooth video perception) on target hardware. However, maintaining this performance while preserving high-resolution spatial details and multi-scale context remains challenging: high-resolution features increase computational costs, while complex context fusion modules introduce latency. The core pursuit in this field lies in optimizing the speed-accuracy trade-off through efficient spatial information preservation and context modeling.

Furthermore, model size and memory footprint are critical considerations for deployment on resource-constrained edge devices. Optimizing storage and computational efficiency for such platforms represents another key research direction.

This survey is structured as follows:

(1) Introduction: Defines real-time semantic segmentation, discusses applications, and outlines challenges.

(2) Fundamentals: Introduces essential techniques for efficient network design, including model compression, lightweight CNN modules, and efficient Transformer components.

(3) Method Taxonomy: Categorizes state-of-the-art real-time segmentation networks by architectural paradigms.

(4) Application scenario: Part of the application field of real-time semantic segmentation is shown.

(5) Evaluation Framework: Presents benchmark datasets, metrics, and comparative analysis of existing methods.

(6) Conclusions and Future Directions: Summarizes research progress, analyzes persistent challenges, and proposes potential research trajectories.

2. PRIOR KNOWLEDGE

2.1. Model Compression Techniques

Model compression is one of the important means to realize real-time semantic segmentation, aiming to improve the reasoning efficiency of models by reducing their number of parameters, computation and memory occupation, and to ensure their efficient operation in environments with limited

computational resources, such as embedded devices or mobile devices. Common model compression methods include weight pruning, quantization, and knowledge distillation.

2.1.1. Weight Pruning

Weight pruning reduces network parameters and computational costs by removing less critical weights (typically those with smaller absolute values) from neural networks. This technique not only decreases model storage requirements but also accelerates inference, particularly when supported by hardware-optimized sparse computation. The standard pruning workflow follows an iterative train-prune-finetune cycle: 1) training a full network, 2) pruning redundant weights, and 3) fine-tuning to recover performance.

Han et al. (2015) pioneered a magnitude-based pruning method for fully connected and convolutional layers, achieving over 90% parameter reduction with negligible accuracy loss [35]. Li et al. (2017) further advanced the field by proposing gradient-guided pruning, which dynamically eliminates redundant connections through neuron importance evaluation [36]. These studies demonstrate pruning's effectiveness in model compression and inference acceleration.

2.1.2. Quantization

Quantization converts model weights and activations from high-precision data types (e.g., 32-bit floating points) to lower-precision representations (e.g., 8-bit integers or below). This technique reduces memory footprint and computational complexity while accelerating inference, particularly when leveraging hardware-optimized low-precision arithmetic. Common strategies involve mapping full-precision values to discrete integer levels through uniform/non-uniform quantization schemes.

Courbariaux et al. (2015) pioneered extreme quantization with BinaryConnect, which binarizes network weights into {-1, +1} values, achieving 32× storage compression and efficient bitwise operations without significant accuracy degradation [37]. Zhang et al. (2018) advanced this field through LQ-Nets, introducing learnable quantization parameters that jointly optimize network accuracy and computational efficiency in low-bitwidth regimes [38].

2.1.3. Knowledge Distillation

Knowledge distillation enables knowledge transfer from a complex, high-capacity teacher model to a compact student model. This paradigm enhances student performance by encouraging mimicry of the teacher's probabilistic outputs (soft labels) rather than solely relying on ground-truth labels, thereby inheriting the teacher's generalization capabilities while maintaining smaller model footprints.

Hinton et al. (2015) pioneered this framework, demonstrating that distilled students could achieve comparable performance to teachers with significantly reduced parameters [39]. Romero et al. (2015) advanced the approach through FitNets, introducing intermediate feature-level distillation that aligns hidden representations between deep teachers and shallow students for enhanced knowledge transfer [40].

2.1.4. Low-Rank Decomposition

Low-rank decomposition techniques reduce the number of parameters by decomposing weight matrices into multiple smaller matrices. This approach is commonly applied to the convolutional or fully connected layers of convolutional neural networks (CNNs). By leveraging low-rank decomposition, the model's parameter count is significantly reduced while preserving most of its representational capacity. Such methods are particularly suitable for network architectures with a large number of parameters, enabling a reduction in model size without substantial performance degradation.

Tai et al. (2016) proposed a low-rank regularization method that constrains the convolutional kernels of CNNs, enforcing the weight matrices in convolutional layers to adopt a low-rank structure. This effectively reduces computational complexity and memory requirements [41]. Similarly, Sainath et

al. (2017) introduced a low-rank matrix decomposition method based on singular value decomposition (SVD) to enhance the efficiency of deep convolutional neural networks, further minimizing both the number of parameters and computational cost [42].

2.1.5. Weight Sharing

Weight sharing reduces parameter count by reusing identical weights across multiple network connections. This technique achieves inherent parameter efficiency in convolutional neural networks (CNNs), where convolutional layers inherently employ weight sharing through their filter kernels – a design that simultaneously minimizes storage requirements and accelerates inference through hardware-friendly memory access patterns.

Zhang et al. (2016) formalized this concept by developing cross-layer weight sharing, where multiple convolutional layers strategically reuse identical weight matrices. This architecture-level optimization significantly reduces both redundant computations and memory footprint in CNNs [43].

2.2. Lightweight Convolutional Neural Networks

Lightweight CNNs are designed to enhance inference speed by reducing the number of parameters, computational cost, and memory footprint while maintaining model accuracy as much as possible. These networks achieve efficiency through optimized architectures and computationally efficient convolution operations, significantly lowering the overall complexity.

2.2.1. Depthwise Separable Convolution

Depthwise separable convolution is one of the most fundamental and widely used techniques in lightweight CNN design. Traditional convolution operations perform computations across all input feature map channels, whereas depthwise separable convolution decomposes this process into two steps. First, depthwise convolution applies independent convolution operations to each input channel. Then, pointwise convolution (1×1 convolution) is used to mix the outputs across all channels. This decomposition significantly reduces computational cost and parameter count while maintaining strong representational capacity.

This concept was first introduced in MobileNet by Chollet (2017), where depthwise separable convolutions reduced computational cost by over 90% compared to standard convolutions while preserving high classification accuracy [44]. This architecture demonstrated the feasibility of efficient deep learning models on resource-constrained devices, making it a key representative of lightweight CNNs.

2.2.2. Neural Architecture Search (NAS)

Neural Architecture Search (NAS) is an automated approach for designing optimal network architectures. Unlike traditional manually designed networks, NAS explores the architecture search space to identify the best-performing structure under computational constraints. The introduction of NAS has greatly facilitated the development of lightweight networks, particularly in scenarios requiring a balance between model accuracy and inference speed.

Liu et al. (2018) proposed NASNet, an architecture designed using NAS to automatically discover efficient CNN structures. Experimental results demonstrated that NASNet achieved superior performance across multiple computer vision tasks while significantly reducing computational complexity [45]. By eliminating the need for manual tuning, NAS enables the automated discovery of efficient architectures, expanding the possibilities for practical applications of lightweight CNNs.

2.3. Efficient Transformer Modules

Since its introduction by Vaswani et al. (2017), the Transformer architecture has become a fundamental tool in both natural language processing (NLP) and computer vision (CV) tasks. With

its strong global modeling capability, the Transformer excels at capturing long-range dependencies. However, its quadratic computational complexity concerning sequence length poses significant challenges, particularly when handling high-resolution images or large-scale data. Therefore, designing efficient Transformer variants that maintain high performance while reducing computational and memory costs has become a crucial research focus.

To address this challenge, researchers have proposed various efficient Transformer modules, which optimize self-attention mechanisms, introduce lightweight network architectures, and reduce computational complexity. This section discusses several key approaches, including local attention mechanisms, sparse attention mechanisms, and hybrid architectures.

2.3.1. Local Attention Mechanism

Local self-attention is a variant designed to reduce computational complexity. In the standard Transformer, self-attention operations have a complexity of $O(N^2)$, where N is the sequence length. In vision tasks, where image resolutions are high, computing global self-attention can be computationally expensive. Local self-attention mitigates this issue by restricting attention computations to localized regions of the input feature map, thereby reducing computational cost.

In Linformer (Wang et al., 2020), researchers introduced a local self-attention approach based on low-rank decomposition. By assuming that the self-attention matrix has a low-rank structure, the method significantly reduces memory and computational overhead when processing long sequences while maintaining performance comparable to standard Transformers in various NLP tasks [46].

2.3.2. Sparse Attention Mechanism

Sparse self-attention reduces computational cost by sparsifying the self-attention matrix. Instead of computing attention scores for all token pairs, sparse self-attention selectively samples and computes attention only for a subset of important positions. This approach effectively lowers computational and memory requirements, making it particularly suitable for processing long sequences.

Longformer (Beltagy et al., 2020) is a representative model that employs a sparse self-attention mechanism. It introduces a sliding window attention approach, where each token attends only to a local context window instead of the entire sequence. This significantly reduces the computational burden associated with full self-attention while maintaining high efficiency in processing long texts. Longformer has demonstrated outstanding performance in multiple NLP tasks [47].

2.3.3. Hybrid Convolution-Attention Architectures

Hybrid architectures integrate convolutional inductive biases with Transformer global modeling, combining efficiency and expressiveness. These designs leverage CNN's local feature extraction capabilities to reduce attention computation costs.

Wu et al. (2021) introduced CvT (Convolutional Vision Transformer), which hierarchically embeds convolutional tokenization within Transformer blocks. This architecture achieves 20% faster inference than pure Transformers on ImageNet while maintaining accuracy, demonstrating effective synergy between convolutional and attention operations [48].

3. CLASSIFICATION OF METHODS

The development of real-time semantic segmentation methods mainly centers around different network frameworks, among which Convolutional Neural Network (CNN), Transformer, and hybrid frameworks combining the advantages of both have become the main direction of current research.

3.1. Real-Time Semantic Segmentation Network Based on CNN

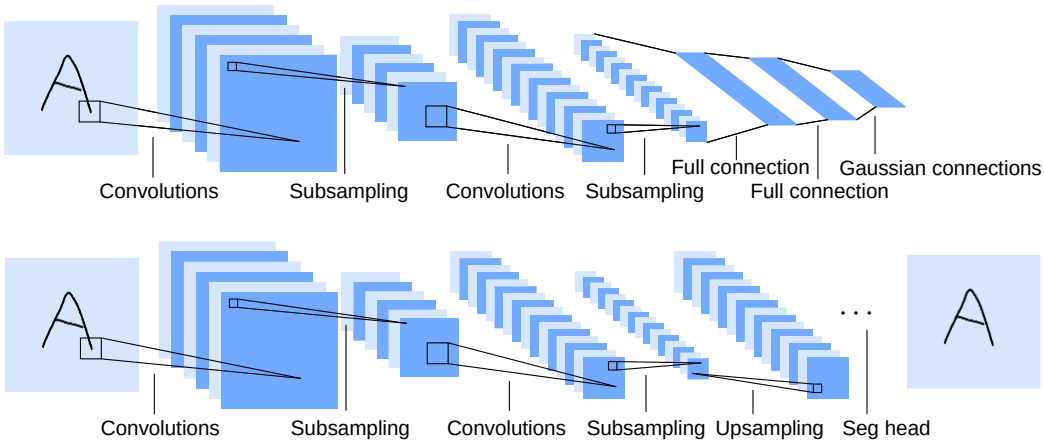

Figure 1. Comparison of traditional CNN classification model and real-time semantic segmentation network

In the early stages, real-time semantic segmentation networks were predominantly based on CNNs, and to this day, they remain a crucial direction for research and practical applications. CNNs extract local image features effectively through convolutional and pooling operations. In real-time semantic segmentation, networks such as Enet [49], ICNet [50], DABNet [51], DFANet [52], ESPNetv2 [53], BiSeNetv2 [54], and Fast-SCNN [55] are all designed based on CNN architectures. As shown in Figure 1, traditional CNN-based classification models generate global class probabilities through fully connected layers, whereas real-time semantic segmentation networks progressively recover spatial resolution via upsampling operations (e.g., transposed convolutions) and output pixel-wise semantic predictions through a segmentation head.

Among these models, ENet is one of the earliest lightweight semantic segmentation networks, designed with a bottleneck structure to reduce computational complexity. By removing traditional fully connected and max pooling layers, ENet significantly improves inference speed. ICNet introduces a multi-resolution cascade structure that enables efficient inference on high-resolution images while leveraging low-resolution branches to reduce computational costs, making it well-suited for autonomous driving applications. DABNet enhances feature representation through Dual Attention Blocks (DABs), which incorporate spatial and channel attention, thereby improving segmentation accuracy without compromising computational efficiency. DFANet adopts a multi-path cascade structure, utilizing shared feature extractors to minimize redundant computations and employing feature aggregation modules to enhance segmentation performance. This architecture is specifically optimized for high-efficiency autonomous driving applications.

ESPNetv2 further refines the ESPNet structure by incorporating more efficient separable convolutions and group convolutions, enabling lower computational complexity and faster inference on lightweight devices. BiSeNetv2 enhances its architecture by introducing a Feature Refinement Module (FRM) to refine spatial path (SP) and context path (CP) features, achieving an improved balance between inference speed and segmentation accuracy. Fast-SCNN integrates a Fast Downsampling Path with a fine-grained upsampling structure, making it well-suited for efficient real-time segmentation on mobile and embedded devices.

Beyond these mainstream models, several novel lightweight architectures have been proposed. Shi et al. introduced the Lightweight Context-Aware Network (LCNet) [56], which employs a partial channel transformation strategy to reduce computational latency. It incorporates a three-branch context aggregation module to expand the receptive field and a dual-attention decoder to improve pixel-level prediction accuracy. This network demonstrates high inference speed and superior segmentation performance in mobile scenarios, highlighting its practical applicability. Song et al.

proposed the Atrous Network (ANet) [57], specifically designed for semantic segmentation. ANet employs Atrous Blocks (A-blocks) to construct an efficient backbone and integrates a lightweight decoder to restore spatial details. Peng et al. introduced the Hierarchical Semantic-Aware Network (HSNet) [58], which incorporates a Hierarchical Feature Refinement Module (HFRM) to recover spatial details and a Cross-Scale Pyramid Fusion Module (CPFM) to integrate local and global contextual information, thereby improving segmentation accuracy. Xue et al. [59] developed an efficient semantic segmentation method that combines channel attention and spatial attention mechanisms. The model leverages HFRM and a Hierarchical Feature Fusion Module (HFFM) to compensate for spatial information loss during downsampling while integrating multi-level features to expand the receptive field.

These advancements demonstrate significant progress in real-time semantic segmentation, particularly in optimizing computational efficiency and segmentation precision for resource-constrained environments.

3.2. Real-time Semantic Segmentation Network based on Hybrid Framework

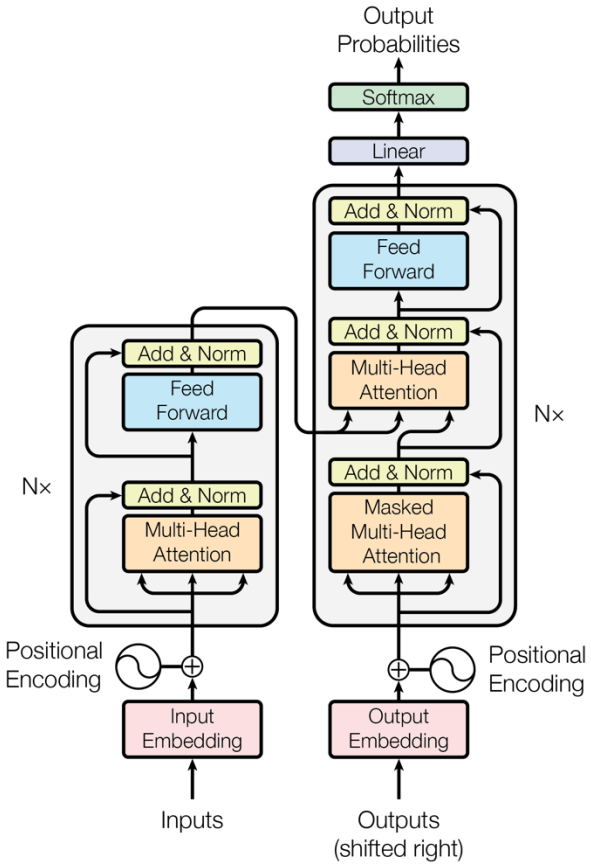

Figure 2. Model architecture of Transformer

Transformer-based semantic segmentation architectures, such as ViT , divide an input image into N patches and model global dependencies through multi-head self-attention (MHSA), as illustrated in Figure 2. The computational complexity of this operation can be expressed as:

$$\mathcal{O}_{\text{Transformer}} = O(N^2 d) \tag{1}$$

Where $N = HW/P^2$ represents the sequence length (H, W denote image resolution, and P is the patch size), and d is the feature dimension. When processing high-resolution inputs (e.g., $H = W = 512$), the N^2 complexity results in a significant increase in memory consumption, limiting the feasibility of real-time deployment. Although global attention mechanisms effectively capture long-

range contextual dependencies, the trade-off between computational cost and high-resolution requirements remains a key challenge for applying Transformers in mobile semantic segmentation tasks.

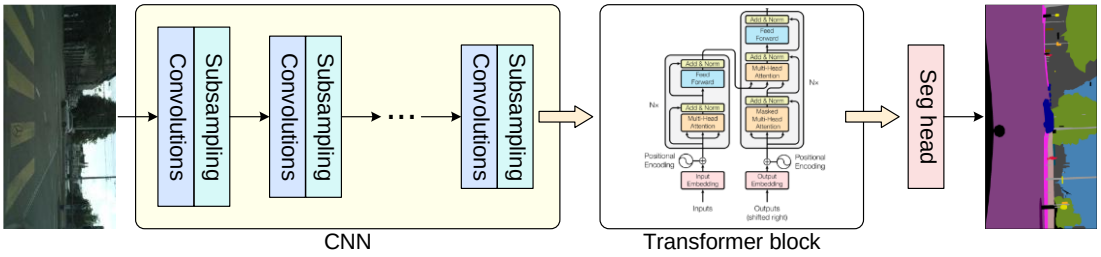

Figure 3. Hybrid architecture design

To balance performance and efficiency, recent studies have adopted hybrid architectures, as illustrated in Figure 3. These designs utilize CNN-based local convolutions in the encoder stage to efficiently extract low-level spatial details (shallow feature maps) while integrating lightweight Transformer modules (e.g., axial attention, window attention) in the decoder stage to capture global semantic relationships (deep feature heatmaps).

For example, SegFormer adopts a multi-scale hybrid structure by first downsampling via a CNN, followed by feature fusion and semantic encoding using a Transformer, and finally upsampling to generate segmentation results. Chen et al. proposed TransUNet [60], a medical image segmentation model that combines the strengths of Transformers and U-Net. In TransUNet, the Transformer extracts global contextual information, while the U-Net structure restores local spatial details, thereby improving segmentation accuracy. Such hybrid models leverage hierarchical feature interactions, reducing computational cost while significantly enhancing segmentation boundary precision and semantic consistency in complex scenarios.

Building upon the advantages of hybrid architectures, recent research has focused on further optimizing these frameworks for real-time semantic segmentation tasks. Key advancements include:

Parallel Local-Global Feature Extraction: Multi-scale feature interactions are enhanced by jointly extracting local and global features, ensuring complementary information representation.

Lightweight Attention Mechanisms: Techniques such as dynamic token sparsification and cross-scale window aggregation improve efficiency while maintaining the ability to model local details (e.g., object boundaries) and global context (e.g., scene topology).

Ge et al. proposed EdgeFormer [61], an efficient edge-device Transformer that introduces parameter-efficient strategies and layer-adaptive mechanisms to enhance model performance under constrained computational and memory conditions. The authors also released EdgeLM, the first publicly available edge-side pretrained seq2seq model, facilitating efficient fine-tuning for real-world applications. Vasu et al. introduced FastViT [62], a hybrid vision Transformer that integrates RepMixer (token mixing operators) and large-kernel convolutions, effectively reducing memory access costs while improving accuracy. Cui et al. proposed FFTNet [63], a feature fusion network that combines CNNs and Transformers for semantic segmentation. FFTNet incorporates a Feature Alignment Module (FAM) to refine spatial details, leverages the Transformer structure for enhanced pixel representation, and employs a Pyramid Convolutional Pooling Module (PCPM) to compress and enrich feature maps while reducing computational overhead.

Hybrid architectures that integrate CNNs and Transformers leverage the strengths of both frameworks, making them a promising direction for real-time semantic segmentation networks. However, achieving an optimal fusion of features and maintaining a balance between computational complexity and segmentation performance remain open research challenges that warrant further investigation.

4. APPLICATION SCENARIOS

4.1. Autonomous Driving

In autonomous driving, real-time semantic segmentation plays a critical role in environment perception by accurately identifying roads, vehicles, pedestrians, traffic signs, and other key objects. This enables autonomous vehicles to make informed decisions in dynamic traffic environments. Many real-time segmentation networks, such as ENet, ICNet, and BiSeNetv2, have been trained and evaluated on datasets like Cityscapes, specifically designed to meet the requirements of autonomous driving. These networks must perform real-time inference on in-vehicle computing platforms, while also adapting to complex road conditions and varying weather scenarios.

Beyond segmentation accuracy, model reliability and robustness are crucial factors in autonomous driving applications, ensuring safe vehicle operation in diverse and challenging environments. Therefore, real-time semantic segmentation networks for autonomous driving must achieve an optimal balance between computational efficiency, robustness, and precision, making them a fundamental component of advanced driver assistance systems (ADAS) and fully autonomous vehicles.

4.2. Mobile Devices

Real-time semantic segmentation is also widely applied in mobile devices such as smartphones and tablets, where computational resources and battery life are inherently limited. Consequently, segmentation networks deployed on mobile platforms must be lightweight and energy-efficient.

Mobile-optimized architectures, including MobileNet and ShuffleNet, have been extensively utilized in mobile image segmentation. These networks leverage techniques such as depthwise separable convolutions to significantly reduce the number of parameters and computational complexity. Additionally, model compression and quantization techniques are frequently employed to further minimize memory usage and power consumption, making real-time segmentation feasible on resource-constrained devices.

Real-time semantic segmentation on mobile platforms is widely applied in image editing, augmented reality (AR), and intelligent camera applications, enhancing user experiences by providing advanced real-time visual processing capabilities.

4.3. Industrial Inspection

In industrial inspection, real-time semantic segmentation is essential for defect detection, component recognition, and quality control. For example, in printed circuit board (PCB) inspection, segmentation networks can accurately distinguish individual electronic components and detect short circuits, open circuits, and other defects in real time.

In such scenarios, high accuracy and reliability are critical, as industrial environments introduce various challenges, such as lighting variations, occlusions, and background noise. To enhance adaptability, deep learning-based segmentation models often integrate domain-specific prior knowledge and data augmentation techniques, improving segmentation accuracy and robustness in industrial settings.

By enabling precise and automated defect detection, real-time semantic segmentation plays a pivotal role in improving manufacturing efficiency, reducing costs, and ensuring product quality across various industrial applications.

5. EVALUATION SYSTEM

5.1. Relevant Datasets

5.1.1. Cityscapes

Cityscapes [64] is a widely used benchmark dataset for urban street scene understanding, primarily designed for autonomous driving research. It consists of street-view images captured from 50 different cities, covering various weather conditions and scene types. The dataset is divided into a training set (2,975 images), a validation set (500 images), and a test set (1,525 images).

Cityscapes provides high-quality pixel-level annotations for 19 semantic classes, including roads, buildings, pedestrians, vehicles, and other urban elements. Its fine-grained annotations and high spatial resolution make it an essential dataset for training and evaluating semantic segmentation models in complex urban environments.

5.1.2. CamVid

CamVid [65, 66] is an early video-based semantic segmentation dataset, primarily derived from dashcam footage. It contains 701 images, split into a training set (367 images), a validation set (101 images), and a test set (233 images).

CamVid provides pixel-wise annotations for 11 semantic categories, including sky, road, vehicles, and pedestrians. Although it features fewer categories compared to other datasets, its video-sequence characteristics make it particularly valuable for studying temporal consistency and dynamic scene understanding in real-world driving scenarios.

5.1.3. Pascal VOC 2012

Pascal VOC 2012 [67] is a widely recognized benchmark dataset for computer vision tasks, including semantic segmentation. It contains 20 object categories, covering a diverse range of human, animal, vehicle, and indoor object classes encountered in everyday scenes. The dataset is divided into a training set (1,464 images), a validation set (1,449 images), and a test set (1,456 images).

Pascal VOC 2012 is known for its high-quality annotations and broad applicability across multiple vision tasks, including object detection and image classification. The overlap between Pascal VOC 2012 and other vision datasets facilitates multi-task learning and cross-domain generalization, making it a valuable resource for developing robust segmentation models.

5.1.4. ADE20K

ADE20K [68, 69] is a large-scale scene parsing dataset designed for scene understanding and semantic segmentation. It contains over 20,000 images, with a training set (20,210 images), a validation set (2,000 images), and a test set (3,352 images).

ADE20K provides pixel-level annotations for 150 semantic categories, covering natural landscapes, urban environments, and indoor scenes. Its diverse category distribution and complex scene compositions make it a critical benchmark for studying semantic segmentation in real-world, multi-object environments. ADE20K is particularly valuable for developing models capable of handling highly cluttered and contextually rich scenarios, which are common in practical applications.

5.2. Evaluation Metrics

5.2.1. Pixel Accuracy (PA)

Pixel Accuracy (PA) is a straightforward and intuitive metric that measures the proportion of correctly predicted pixels over the total number of pixels in the dataset. It is computed as follows:

$$PA = \frac{\sum_{i=0}^{C} p_{ii}}{\sum_{i=0}^{C} \sum_{j=0}^{C} p_{ij}} \quad (2)$$

Where C represents the total number of classes, and p_{ij} denotes the number of pixels belonging to the ground truth class i that are predicted as class j.

While PA provides an overall measure of segmentation accuracy, it can be misleading in datasets with class imbalance, as it tends to favor dominant classes while underrepresenting minority categories.

5.2.2. Mean Pixel Accuracy (MPA)

Mean Pixel Accuracy (MPA) computes PA for each class individually and then averages the values across all classes:

$$MPA = \frac{1}{C} \sum_{i=0}^{C} \frac{p_{ii}}{\sum_{j=0}^{C} p_{ij}} \quad (3)$$

MPA mitigates the class imbalance issue by evaluating segmentation accuracy on a per-class basis, ensuring that underrepresented categories contribute equally to the final performance metric.

5.2.3. Intersection over Union (IoU)

Intersection over Union (IoU), also known as the Jaccard Index, is one of the most widely used metrics for semantic segmentation. It quantifies the overlap between the predicted segmentation mask and the ground truth. The IoU for each class is calculated as:

$$IoU_i = \frac{p_{ii}}{\sum_{j=0}^{C} p_{ij} + \sum_{j=0}^{C} p_{ji} - p_{ii}} \quad (4)$$

Where p_{ii} is the number of correctly predicted pixels for class iii, while p_{ij} and p_{ji} represent misclassified pixels.

The Mean IoU (mIoU), a widely accepted benchmark for segmentation models, is obtained by averaging the IoU values across all classes:

$$mIoU = \frac{1}{C} \sum_{i=0}^{C} IoU_i \quad (5)$$

mIoU provides a balanced and comprehensive assessment of a model's segmentation accuracy, making it the primary metric in many benchmarking studies.

5.2.4. Frequency-Weighted IoU (FWIoU)

Frequency-Weighted Intersection over Union (FWIoU) considers the occurrence frequency of each class in the dataset to provide a weighted evaluation of segmentation performance. It is defined as:

$$FWIoU = \frac{\sum_{i=0}^{C} f_i IoU_i}{\sum_{i=0}^{C} f_i} \quad (6)$$

Where f_i denotes the occurrence frequency of class i in the dataset. FWIoU reflects the model's practical performance in real-world applications by accounting for the class distribution, ensuring that frequently appearing classes have a greater influence on the final score.

5.2.5. Computational Complexity Metrics: FLOPs & Params

Beyond segmentation accuracy, computational efficiency is crucial for real-time applications. Two key efficiency metrics are:

Floating Point Operations per Second (FLOPs): Measures the number of arithmetic operations required for inference, reflecting computational complexity. Lower FLOPs indicate higher efficiency, which is essential for mobile and embedded applications.

Number of Parameters (Params): Represents the total trainable weights in the model. A lower parameter count typically results in reduced memory footprint and faster inference, making the model more suitable for resource-constrained devices.

Efficient segmentation networks aim to balance accuracy with computational cost, optimizing mIoU while minimizing FLOPs and Params to achieve real-time performance on edge devices.

5.3. Performance Summary of Existing Methods

Table 1. Performance of existing methods on the Cityscapes dataset

Method	Resolution	Params(M)	FPS	Val mIoU(%)	Test mIoU(%)
Enet [49]	1024 × 512	0.4	76.9	-	58.3
ERFNet [70]	640 × 360	-	-	71.3	70.33
BiSeNet [71]	1536 × 768	49.0	65.5	74.8	74.7
BiSeNet V2 [54]	1024 × 512	-	47.3	75.8	75.3
STDC [72]	1536 × 768	22.2	97.0	77.0	76.8
Fast-SCNN [55]	2048 × 1024	1.1	123.5	68.6	68.0
DDRNet [73]	2048 × 1024	20.1	37.1	79.5	79.4
ICNet [50]	2048 × 1024	26.5	30.3	-	69.5
DFANet [52]	1024 × 1024	7.8	100.0	-	71.3
PIDNet [74]	2048 × 1024	36.9	31.1	80.9	80.6
ESPNet [75]	1024 × 512	0.4	113	-	60.3
LETNet [76]	1024 × 512	0.95	150	72.8	-
SCTNet-S-Seg50 [77]	1024 × 512	4.6	160.3	72.8	-
AFFormer-B [78]	2048 × 1024	3.0	22	78.7	-
RTFormer [79]	2048 × 1024	16.8	39.1	79.3	-
SeaFormer [80]	1024 × 512	-	-	77.7	77.5

As shown in Table 1, existing real-time semantic segmentation methods exhibit significant trade-offs among speed (FPS), accuracy (mIoU), and model efficiency (number of parameters).

High-speed models, such as Fast-SCNN (123.5 FPS) and LETNet (150 FPS), achieve real-time performance through highly simplified architectures with minimal parameter counts (≤1.1M). However, their respective mIoU scores of 68.0% and 72.8% indicate that excessive lightweighting compromises semantic understanding.

High-accuracy models, such as PIDNet (80.6% mIoU) and RTFormer (79.3% mIoU), leverage complex architectures (≥16.8M parameters) to enhance segmentation precision. However, their FPS values remain below 40, making them unsuitable for strict real-time applications.

Balanced models, such as STDC (97.0 FPS, 76.8% mIoU) and DFANet (100.0 FPS, 71.3% mIoU), adopt multi-branch feature fusion strategies, achieving a trade-off between speed and accuracy. These models are well-suited for edge computing scenarios.

It is noteworthy that the parameter count does not strictly correlate with performance. For example, SCTNet-S-Seg50 achieves an exceptionally high speed of 160.3 FPS with only 4.6M parameters, highlighting the importance of efficient operator design in real-time segmentation.

Current real-time segmentation models focus on three primary optimization directions: Lightweight Backbone Optimization – e.g., LETNet, which simplifies the network architecture to improve

inference speed. Dynamic Resolution Adjustment – e.g., BiSeNet series, which dynamically balances computational efficiency and feature representation. Hybrid Attention-Convolution Design – e.g., RTFormer, which integrates attention mechanisms with convolutional layers to enhance feature extraction.

Although some recent methods, such as SeaFormer and AFFormer-B, achieve accuracy levels comparable to non-real-time models (e.g., DeepLabV3+), their FPS remains constrained by global attention computations (e.g., AFFormer-B reaches only 22 FPS). To further enhance real-time segmentation models, future studies should explore: Low-rank approximation and sparsification techniques to reduce computational complexity while preserving the global modeling capacity of Transformer-based architectures. Hardware-aware architecture search, tailoring network structures and operations for specific hardware platforms (e.g., NPU, FPGA) to improve real-world efficiency. Multi-task joint optimization, integrating semantic segmentation, edge detection, and depth estimation to improve model generalization and reduce dependency on large-scale annotated datasets. These advancements will be crucial for bridging the gap between real-time efficiency and high-accuracy segmentation in practical applications.

6. CONCLUSION AND FUTURE DIRECTIONS

6.1. Conclusion

Real-time semantic segmentation has become a crucial research area in computer vision, achieving significant progress in recent years. By leveraging model compression techniques, efficient convolutional neural network modules, and optimized Transformer-based architectures, real-time segmentation networks have progressively enhanced segmentation accuracy while maintaining real-time performance.

This paper provides a comprehensive review of real-time semantic segmentation in deep learning, covering fundamental concepts, application scenarios, and key challenges. Additionally, we systematically analyze and categorize existing real-time segmentation methods, introducing commonly used architectural designs and optimization strategies. A complete evaluation framework is also presented, encompassing benchmark datasets, evaluation metrics, and comparative performance analyses across mainstream real-time segmentation approaches.

6.2. Challenges and Future Directions

Despite substantial progress, real-time semantic segmentation still faces several critical challenges:

Balancing Accuracy and Computational Efficiency – Achieving high segmentation accuracy under constrained computational resources remains a fundamental challenge. While many existing methods offer a reasonable trade-off between speed and accuracy, they still struggle with fine-grained segmentation of small objects and intricate details in complex scenes.

Multi-Scale Context and Spatial Detail Fusion – Effectively integrating multi-scale contextual information and spatial details is crucial for improving segmentation performance. Although various multi-scale feature fusion strategies have been explored, designing more efficient context capture mechanisms remains an open research problem.

Computational Complexity of Transformer-Based Models – While Transformer-based segmentation networks excel in global context modeling, their high computational cost limits their applicability in real-time scenarios. Optimizing their architectures and enhancing computational efficiency is a key research direction for enabling broader real-time applications.

Looking forward, the future development of real-time semantic segmentation is expected to evolve in the following directions:

Advanced Model Compression and Hardware Acceleration – The integration of quantization, pruning, and knowledge distillation with hardware-optimized inference techniques (e.g., TPUs, FPGAs, and NPUs) will further enhance real-time efficiency. Moreover, emerging computing paradigms such as neuromorphic computing and quantum computing may provide new opportunities for accelerating real-time segmentation.

Multi-Modal Data Fusion – Combining RGB images with other sensing modalities (e.g., LiDAR, radar, or audio signals) could improve segmentation robustness, particularly in safety-critical applications such as autonomous driving and robotic vision.

Enhancing Model Explainability – Improving the interpretability of real-time segmentation models is essential for increasing transparency and trustworthiness, especially in high-stakes applications like medical image analysis and autonomous navigation.

Task-Specific Model Customization – Developing domain-adaptive and application-specific segmentation models by leveraging prior knowledge and task-specific data characteristics will be key to enhancing real-world deployment and performance.

By addressing these challenges and advancing along these research directions, real-time semantic segmentation can achieve higher efficiency, robustness, and broader applicability across various domains.

REFERENCES

[1] Qureshi I, Yan J, Abbas Q, et al. Medical image segmentation using deep semantic-based methods: A review of techniques, applications and emerging trends [J]. Information Fusion, 2023, 90: 316-352.

[2] Kaur A, Singh Y, Chinagundi B. ResUNet++: a comprehensive improved UNet++ framework for volumetric semantic segmentation of brain tumor MR images [J]. Evolving Systems, 2024, 15(4): 1567-1585.

[3] Lu Y, Li W, Cui Z, et al. Beyond low-dimensional features: Enhancing semi-supervised medical image semantic segmentation with advanced consistency learning techniques [J]. Expert Systems with Applications, 2025, 261: 125456.

[4] Dang T V, Bui N T. Multi-Scale Fully Convolutional Network-Based Semantic Segmentation for Mobile Robot Navigation [J]. Electronics, 2023, 12(3).

[5] Cheng J, Deng C, Su Y, et al. Methods and datasets on semantic segmentation for Unmanned Aerial Vehicle remote sensing images: A review [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2024, 211: 1-34.

[6] Li X, Xu F, Yu A, et al. A Frequency Decoupling Network for Semantic Segmentation of Remote Sensing Images [J]. IEEE Transactions on Geoscience and Remote Sensing, 2025.

[7] Ma X, Wang Z, Hu Y, et al. Kolmogorov-Arnold Network for Remote Sensing Image Semantic Segmentation [J]. arXiv preprint arXiv:2501.07390, 2025.

[8] Voronin V, Semenishchev E, Zelensky A, et al. Real-time deep learning semantic segmentation for 3-D augmented reality [C]//LI M, SHI K, ASGHARI H, et al. Real-time Photonic Measurements, Data Management, and Processing VII: Vol. 12772. SPIE, 2023: 127720L.

[9] Wang Y, Zhang J, Chen Y, et al. An Automated Learning Method of Semantic Segmentation for Train Autonomous Driving Environment Understanding [J]. IEEE Transactions on Industrial Informatics, 2024, 20(4): 6913-6922.

[10] Hao W, Wang J, Lu H. A Real-Time Semantic Segmentation Method Based on Transformer for Autonomous Driving [J]. Computers, Materials & Continua, 2024, 81(3).

[11] Lin F, Lin T, Yao Y, et al. VPA-Net: A visual perception assistance network for 3d lidar semantic segmentation [J]. Pattern Recognition, 2025, 158: 111014.

[12] Wang H, Jiang X, Ren H, et al. Swiftnet: Real-time video object segmentation [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 1296-1305.

[13] Otsu N. A Threshold Selection Method from Gray-Level Histograms [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66.

[14] MacQueen J. Some methods for classification and analysis of multivariate observations [C]//Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press. 1967.

[15] Meyer F, Beucher S. Morphological segmentation [J]. Journal of Visual Communication and Image Representation, 1990, 1(1): 21-46.

[16] Adams R, Bischof L. Seeded region growing [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(6): 641-647.

[17] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models [J]. International Journal of Computer Vision, 1988, 1(4): 321-331.

[18] Boykov Y, Jolly M P. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images [J]. Proceedings of IEEE International Conference on Computer Vision (ICCV), 2001: 105-112.

[19] Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data [J]. Proceedings of the 18th International Conference on Machine Learning (ICML), 2001: 282-289.

[20] Li S Z. Markov random field modeling in image analysis [M]. Springer Science & Business Media, 2009.

[21] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 3431-3440.

[22] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [J]. Advances in neural information processing systems, 2012, 25.

[23] Simonyan K. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014.

[24] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions [C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 1-9.

[25] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016: 770-778.

[26] Sun K, Xiao B, Liu D, et al. Deep High-Resolution Representation Learning for Human Pose Estimation [Z]. (2019).

[27] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation [C]//NAVAB N, HORNEGGER J, WELLS W M, et al. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham: Springer International Publishing, 2015: 234-241.

[28] 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.

[29] Lin G, Milan A, Shen C, et al. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation [J]. CoRR, 2016, abs/1611.06612.

[30] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [J]. CoRR, 2020, abs/2010.11929.

[31] Zheng S, Lu J, Zhao H, et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers [C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021: 6877-6886.

[32] Wang W, Xie E, Li X, et al. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions [J]. CoRR, 2021, abs/2102.12122.

[33] Xie E, Wang W, Yu Z, et al. SegFormer: Simple and efficient design for semantic segmentation with transformers [J]. Advances in neural information processing systems, 2021, 34: 12077-12090.

[34] Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows [J]. CoRR, 2021, abs/2103.14030.

[35] Han S, Pool J, Tran J, et al. Learning both Weights and Connections for Efficient Neural Networks [J]. CoRR, 2015, abs/1506.02626.

[36] Li H, Kadav A, Durdanovic I, et al. Pruning Filters for Efficient ConvNets [J]. CoRR, 2016, abs/1608.08710.

[37] Courbariaux M, Bengio Y, David J P. BinaryConnect: Training Deep Neural Networks with binary weights during propagations [J]. CoRR, 2015, abs/1511.00363.

[38] Zhang D, Yang J, Ye D, et al. LQ-Nets: Learned Quantization for Highly Accurate and Compact Deep Neural Networks [J]. CoRR, 2018, abs/1807.10029.

[39] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network [Z]. (2015).

[40] Romero A, Ballas N, Kahou S E, et al. FitNets: Hints for Thin Deep Nets [Z]. (2015).

[41] Tai C, Xiao T, Zhang Y, et al. Convolutional neural networks with low-rank regularization [Z]. (2016).

[42] Sainath T N, Kingsbury B, Sindhwani V, et al. Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets [C]//2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013: 6655-6659.

[43] Zhang H, He J, Ko S B. Improved Hybrid Memory Cube for Weight-Sharing Deep Convolutional Neural Networks [C]//2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS). 2019: 122-126.

[44] Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017: 1800-1807.

[45] Liu C, Zoph B, Neumann M, et al. Progressive Neural Architecture Search [C]//FERRARI V, HEBERT M, SMINCHISESCU C, et al. Computer Vision – ECCV 2018. Cham: Springer International Publishing, 2018: 19-35.

[46] Wang S, Li B Z, Khabsa M, et al. Linformer: Self-Attention with Linear Complexity [J]. CoRR, 2020, abs/2006.04768.

[47] Beltagy I, Peters M E, Cohan A. Longformer: The Long-Document Transformer [J]. CoRR, 2020, abs/2004.05150.

[48] Wu H, Xiao B, Codella N, et al. CvT: Introducing Convolutions to Vision Transformers [C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021: 22-31.

[49] 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.

[50] 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). 2018: 405-420.

[51] Li G, Yun I, Kim J, et al. DABNet: Depth-wise Asymmetric Bottleneck for Real-time Semantic Segmentation [J]. CoRR, 2019, abs/1907.11357.

[52] Li H, Xiong P, Fan H, et al. DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation [C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019: 9514-9523.

[53] Mehta S, Rastegari M, Shapiro L G, et al. ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network [J]. CoRR, 2018, abs/1811.11431.

[54] Yu C, Gao C, Wang J, et al. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation [J]. CoRR, 2020, abs/2004.02147.

[55] Poudel R P K, Liwicki S, Cipolla R. Fast-SCNN: Fast Semantic Segmentation Network [J]. CoRR, 2019, abs/1902.04502.

[56] Shi M, Lin S, Yi Q, et al. Lightweight Context-Aware Network Using Partial-Channel Transformation for Real-Time Semantic Segmentation [J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(7): 7401-7416.

[57] Song X, Fang X, Meng X, et al. Real-time semantic segmentation network with an enhanced backbone based on Atrous spatial pyramid pooling module [J]. Engineering Applications of Artificial Intelligence, 2024, 133: 107988.

[58] Peng X, Cheng J, Tang X, et al. HSNet: An Intelligent Hierarchical Semantic-Aware Network System for Real-Time Semantic Segmentation [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2024, 54(7): 4318-4330.

[59] Ye B, Xue R, Wu Q. A hybrid attention multi-scale fusion network for real-time semantic segmentation [J]. Scientific Reports, 2025, 15(1): 872.

[60] Chen J, Lu Y, Yu Q, et al. Transunet: Transformers make strong encoders for medical image segmentation [J]. arXiv preprint arXiv:2102.04306, 2021.

[61] Ge T, Chen S Q, Wei F. EdgeFormer: A Parameter-Efficient Transformer for On-Device Seq2seq Generation [Z]. (2022).

[62] Vasu P K A, Gabriel J, Zhu J, et al. FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization [Z]. (2023).

[63] Li T, Cui Z, Zhang H. Semantic segmentation feature fusion network based on transformer [J]. Scientific Reports, 2025, 15(1): 6110.

[64] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3213-3223.

[65] Brostow G J, Shotton J, Fauqueur J, et al. Segmentation and Recognition Using Structure from Motion Point Clouds [C]//ECCV (1). 2008: 44-57.

[66] Brostow G J, Fauqueur J, Cipolla R. Semantic object classes in video: A high-definition ground truth database [J]. Pattern recognition letters, 2009, 30(2): 88-97.

[67] Everingham M, Van Gool L, Williams C K I, et al. The Pascal Visual Object Classes (VOC) Challenge [J]. International Journal of Computer Vision, 2010, 88(2): 303-338.

[68] Zhou B, Zhao H, Puig X, et al. Scene Parsing through ADE20K Dataset [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017: 5122-5130.

[69] Zhou B, Zhao H, Puig X, et al. Semantic Understanding of Scenes through the ADE20K Dataset [Z]. (2018).

[70] Romera E, Álvarez J M, Bergasa L M, et al. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation [J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(1): 263-272.

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

[72] Fan M, Lai S, Huang J, et al. Rethinking BiSeNet For Real-time Semantic Segmentation [C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021: 9711-9720.

[73] Pan H, Hong Y, Sun W, et al. Deep Dual-Resolution Networks for Real-Time and Accurate Semantic Segmentation of Traffic Scenes [J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(3): 3448-3460.

[74] Xu J, Xiong Z, Bhattacharyya S P. PIDNet: A Real-time Semantic Segmentation Network Inspired by PID Controllers [Z]. (2023).

[75] Mehta S, Rastegari M, Caspi A, et al. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation [C]//Proceedings of the european conference on computer vision (ECCV). 2018: 552-568.

[76] Xu G, Li J, Gao G, et al. Lightweight Real-Time Semantic Segmentation Network With Efficient Transformer and CNN [J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(12): 15897-15906.

[77] Xu Z, Wu D, Yu C, et al. Sctnet: Single-branch cnn with transformer semantic information for real-time segmentation [C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 38. 2024: 6378-6386.

[78] Dong B, Wang P, Wang F. Head-Free Lightweight Semantic Segmentation with Linear Transformer [Z]. (2023).

[79] Wang J, Gou C, Wu Q, et al. RTFormer: Efficient Design for Real-Time Semantic Segmentation with Transformer [Z]. (2022).

[80] Wan Q, Huang Z, Lu J, et al. SeaFormer++: Squeeze-enhanced Axial Transformer for Mobile Visual Recognition [Z]. (2024).