Lightweight Context-Aware Network Using Partial-Channel Transformation for Real-Time Semantic Segmentation

Min Shi, Shaowen Lin, Qingming Yi, Jian Weng, *Senior Member, IEEE*, Aiwen Luo, *Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— Optimizing the computational efficiency of the artificial neural networks is crucial for resource-constrained platforms like autonomous driving systems. To address this challenge, we proposed a Lightweight Context-aware Network (LCNet) that accelerates semantic segmentation while maintaining a favorable trade-off between inference speed and segmentation accuracy in this paper. The proposed LCNet introduces a partial-channel transformation (PCT) strategy to minimize computing latency and hardware requirements of the basic unit. Within the PCT block, a three-branch context aggregation (TCA) module expands the feature receptive fields, capturing multiscale contextual information. Additionally, a dual-attention-guided decoder (DD) recovers spatial details and enhances pixel prediction accuracy. Extensive experiments on three benchmarks demonstrate the effectiveness and efficiency of the proposed LCNet model. Remarkably, a smaller model LCNet$_{3\_7}$ achieves 73.8% mIoU with only 0.51 million parameters, with an impressive inference speed of ∼142.5 fps and ∼9 fps using a single RTX 3090 GPU and Jetson Xavier NX, respectively, on the Cityscapes test set at 1024 × 1024 resolution. A more accurate version of the LCNet$_{3\_11}$ can achieve 75.8% mIoU with 0.74 million parameters at ∼117 fps inference speed on Cityscapes at the same resolution. Much faster inference speed can be achieved at smaller image resolutions. LCNet strikes a great balance between computational efficiency and prediction capability for mobile application scenarios. The code is available at https://github.com/lztjy/LCNet.**

***Index Terms*— Real-time semantic segmentation, partial-channel transformation, context-aware aggregation, reverse attention, spatial attention.**

I. Introduction

Semantic segmentation aims to classify each image pixel into a specified semantic category, which is a fundamental and vital task in computer vision. As a crucial component in street scene understanding, vision-based semantic segmentation is widely applied to identify the surrounding environment for developing intelligent fully autonomous vehicles to make reasonable decisions. Since the fully convolutional network (FCN) [1] transformed the semantic segmentation task into a pixel-by-pixel classification problem, the deep-learning-based semantic segmentation methods can be trained end-to-end to yield accurate accumulated values to approximate the true values that it has achieved rapid development. There are largely two development tendencies for semantic segmentation networks: 1) high accuracy, and 2) fast speed.

The existing high-accuracy frameworks pursue powerful feature expression capabilities by adopting deep backbones, such as VGG [2] and ResNet [3]. However, these deep convolutional neural network (CNN) frameworks with complex architectures normally require tremendous computing resources and lead to an undesired high latency. A Dual-CNN [4] employs a high-resolution system for training while a cost-effective standard-resolution system for inference, achieving a high quality estimation result. Recently, the vision transformer (ViT) schemes [5], [6], [7] have leveraged the self-attention mechanism and large-scale data to model long-range dependencies, capture spatial relationships, and incorporate global context to pursue a higher accuracy. However, ViT-based approaches can be computationally expensive since the self-attention mechanism on large-scale datasets requires significant computational resources for model training.

For real-time mobile applications such as street scene parsing in automatic driving, an ideal semantic segmentation solution requires appropriate precision, fast processing speed, and a small model size so that it meets the needs of resource-constrained mobile application scenarios. Transformer knowledge distillation can be employed to build a lightweight Transformer or CNN-based student model for semantic segmentation [9], [10]. Nevertheless, the student model in ViT-based models such as [11] and [12] should be guided and supervised by a larger, more complex teacher

Manuscript received 3 February 2023; revised 30 June 2023, 5 September 2023, and 21 December 2023; accepted 27 December 2023. Date of publication 22 January 2024; date of current version 2 July 2024. This work was supported in part by NSFC under Grant 62002134, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515010834, in part by the Key Laboratory of New Semiconductors and Devices of Guangdong Higher Education Institutes under Grant 2021KSY001, in part by the Guangdong Key Laboratory of Data Security and Privacy Preserving under Grant 2017B030301004, in part by the Jinan University-Techtotop (JNU) Joint Foundation for Postgraduates Training under Grant 82621176, and in part by the Special Fund for Science and Technology Innovation Strategy of Guangdong Province (Cultivation of Science and Technology Innovation for College Students) under Grant pdjh2023b0061. The Associate Editor for this article was D. F. Wolf. *(Corresponding author: Aiwen Luo.)*

Min Shi, Shaowen Lin, Qingming Yi, and Aiwen Luo are with the Department of Electronic Engineering, Jinan University, Guangzhou 510632, China (e-mail: faith.awluo@gmail.com).

Jian Weng is with the Department of Computer Science, Jinan University, Guangzhou 510632, China, and also with the Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangzhou 510632, China.

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau, China.

Digital Object Identifier 10.1109/TITS.2023.3348631

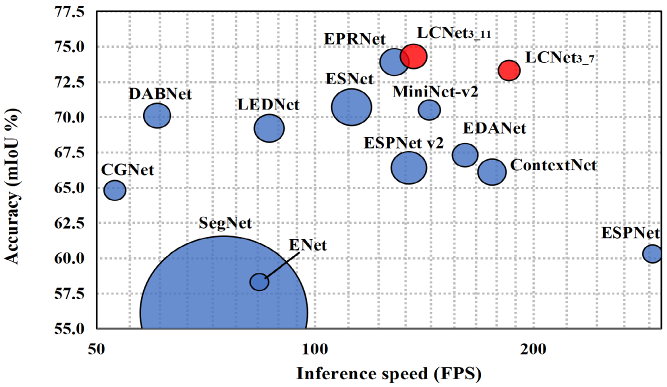


Fig. 1. Comparison with state-of-the-art neural networks in terms of the trade-off between accuracy and computing efficiency on the Cityscapes dataset [8], where the parameter amount is indicated by the relative diameter of bubbles.

model by transferring knowledge trained with large-scale datasets. Apart from knowledge-distillation-based schemes, many existing real-time CNN models [13], [14], [15], [16] focus on miniaturizing network architecture and reducing computational redundancy. For example, ENet [13] achieves 58.3% mIoU and outperforms the SegNet [17] by reducing the model parameters to about 0.4 Million using depthwise convolution (DwConv). ERFNet [18] employs factorized convolution technology to reduce redundant computation. DABNet [19] adopts an effective bottleneck to improve computing efficiency while keeping a comparable accuracy. The above methods inspire numerous subsequent studies but a better trade-off between computing efficiency and prediction accuracy still leaves much to be desired for practical mobile scenarios. In this paper, we proposed a novel attention-guided Lightweight Context-aware Network (LCNet) and achieved a remarkable performance in balancing the inference speed, parameter number, and segmentation accuracy as illustrated in Fig. 1. Our main contributions can be summarized as follows:

- A Three-branch Context Aggregation (TCA) module is elaborately designed to capture the local features and aggregate the surrounding contextual information by skip-connection and dilated convolution with different dilation rates, improving the segmentation accuracy greatly at a very low computational cost.
- A lightweight basic block named Partial Channel Transformation (PCT) is proposed for stacking a compact two-stage encoder. The PCT transmits only partial channels of feature maps across the TCA branch and identifies the rest channels to the output, significantly reducing the computing redundancy.
- A novel Dual-attention-guided Decoder (DD) to emphasize inter-class differences and intra-class boundaries by refining shallow and deep hierarchy features adaptively from the outputs of two stages in the encoder, achieving accurate pixel-level predictions.

Finally, a highly efficient LCNet for fast semantic segmentation is built by leveraging the above components using an asymmetric encoder-decoder framework.

The remainder of this paper is organized as follows. Section II introduces relevant research work on semantic segmentation; Section III describes the architecture of each core component and the overall structure of the proposed semantic segmentation model of LCNet in detail. A series of experiments are carried out to estimate the effectiveness of LCNet on the powerful RTX 3090 GPU and a low-power mobile GPU on Xavier NX in Section IV. Section V concludes this paper.

II. Related Work

A multitude of semantic segmentation techniques have been devised towards achieving high computational efficiency while maintaining remarkable accuracy within affordable resource consumption for practical application scenarios.

A. *Basic Stacking Unit Construction*

The rational combination of various convolution techniques to construct the basic stacking unit (e.g., block or bottleneck) is critical for building new structures with fewer parameters and faster processing speed. Four typical existing basic units and our proposed basic block for semantic segmentation networks are illustrated in Fig. 2. ResNet [3] proposes an impressive bottleneck structure to greatly reduce the number of model parameters for deep learning models. DABNet [19] presents a depthwise asymmetric bottleneck (DAB) module to greatly improve the accuracy with a small number of parameters by integrating with depthwise-dilated convolution (DwDConv) and factorized convolution. LEDNet [20] designs a split-shuffle-non-bottleneck (SS-nbt) to simplify the computing operation for channel-width transformation, facilitating information exchange between channels. However, the essential spatial information necessary is ignored and more convolutional layers are needed in the SS-nbt block. Both DABNet and LEDNet employ factorized convolutions to reduce the computational complexity but result in a potential loss of modeling capability. LMFFNet [21] employs DwDConv to reduce memory requirements. However, the skip connection in the above four basic stacking blocks increases the memory requirements for storing all channels of the intermediate feature maps during both forward and backward passes. To address these challenges, we propose a Partial Channel Transformation (PCT) block as illustrated in Fig. 2 (e) for building a fast, accurate, and lightweight backbone for semantic segmentation.

B. *Depthwise Convolution and Dilated Convolution*

Depthwise convolution (DwConv) separates each input channel for convolution operations and then combines the outputs using pointwise convolution to obtain the final feature maps. DwConv is often employed to significantly reduce the computational complexity for lightweight CNNs [22], [23]. Instead of employing complex architectures, many lightweight models such as ESPNet V2 [24], ESNet [25], and LEDNet [20] utilize distributed dilated convolution (DConv) throughout the encoding process to gain multiscale feature information without additional parameter consumption. The DConv aims to capture broader spatial context without increasing parameter

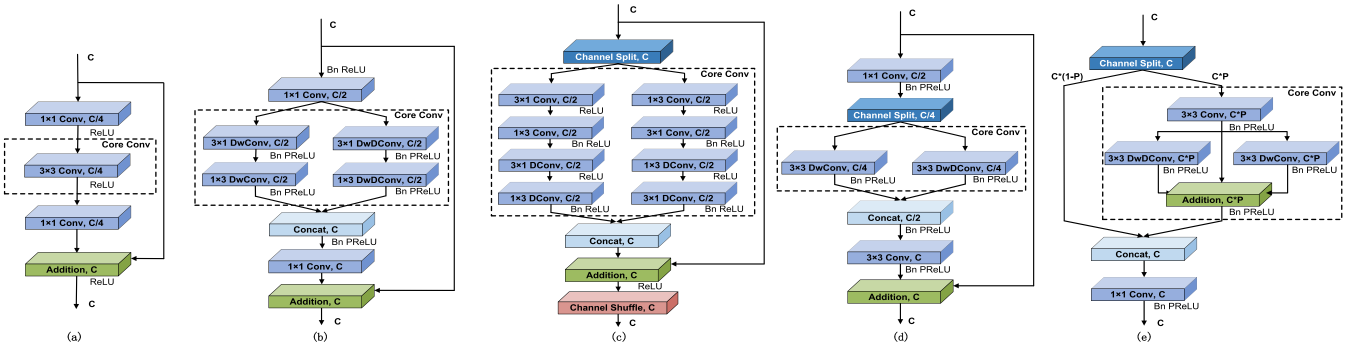


Fig. 2. Architectures of five basic stacking units for the backbone networks: (a) Bottleneck of ResNet; (b) DAB module; (c) Split-shuffle-non-bottleneck (SS-nbt); (d) SEM_B module; (e) Our Partial Channel Transformation (PCT) module. "DConv", "DwConv" and "DwDConv" denote dilated convolution, depthwise convolution, and depthwise-dilated convolution, respectively. "Core Conv" indicates the core convolutional layers for feature extraction in each module. C is the number of input channels while P indicates the partial rate of feature channels transmitted to the TCA module of PCT.

amount or computational complexity with a larger receptive field. For instance, Deeplab V3 [26] suggests using parallel filters with different dilation rates in the Atrous Spatial Pyramid Pooling (ASPP) module to capture multiscale receptive features. Multishuffle-block dilated convolution (MSDC) is reported in [27] to achieve more accurate detection by using adaptive image segmentation. However, DConv can bring extra computational burden to real-time networks, particularly within a multi-branch structure to capture contextual information across various receptive fields. And the grouping operation in DwConv can destroy the channel correlation of the feature maps. To address these challenges, ShuffleNet V2 [14] brings in shuffle operation while DABNet [19], EPRNet [28], and MinNet-v2 [15] choose to use the pointwise convolution to re-establish the information exchange between channels. Careful consideration is needed when incorporating DwConv and DConv in network architecture to balance the computational burden and the desired level of contextual information.

C. Attention Mechanism (AM)

AM refers to the phenomenon that human beings always pay more attention to the most important part when viewing an image. AM selectively captures important information in different dimensions of the input feature maps in semantic segmentation tasks. Normally, spatial attention (SA) [29] and channel attention (CA) [30] are two typical AMs used for achieving accurate CNNs. SA focuses on modeling relationships between spatial correlations of pixels while CA focuses on correlations between feature channels. However, SA involves modeling inter-dependencies between spatial locations and thus leads to increased computational complexity. CA can be more computationally efficient so it is more widely used in various convolutional layers. For example, DFN [31] designs a channel attention block (CAB) to apply high-level semantic features to guide the selection of low-level features. PANet [32] proposes a global attention upsample (GAU) strategy to apply CA to the upsampling operation for the sake of better detail recovery and lower computing latency. A dual attention module (DAM) [33] is presented to integrate a global attention block and local attention block for extracting different scales of image features. Lin et al. [34] report a dual-branch geometric attention network in 3D dental models for accurate segmentation. Nevertheless, achieving accurate semantic segmentation relies heavily on spatial information. The conventional SA only focuses on spatial relationships of relevant parts of the input feature maps. It may yield confusing results in regions that are located at the boundaries between two objects or between the objects and the background. Chen, et al. [35] report a reverse attention (RA) mechanism to discover the missing object parts and residual details for salient object detection. The object regions are expanded progressively by the RA and gradually improve the segmentation accuracy by capturing the salient differences between objects as in RAN [36]. Inspired by these works, the conventional SA and the recently introduced RA are incorporated to enable the fusion of deep and shallow feature maps by the decoder in our proposed model, thereby enhancing the segmentation of object boundaries and the background more precisely.

D. Asymmetric Encoder-Decoder Architecture

The encoder-decoder framework allows for the extraction of rich hierarchical features by progressively downsampling and upsampling the input images, leading to accurate and efficient semantic segmentation results. The encoder mainly aims to generate feature maps containing essential semantic information while the decoder is used to compensate for the loss of spatial details in feature maps, restore the image size, and predict the pixel label. Many existing approaches, such as ERFNet [18] and FDDWNet [37], demonstrate that the asymmetric encoder-decoder architecture is more computation-efficient than the symmetric architecture mainly because the asymmetric architecture advocates spending extremely low resources on the decoder. Similarly, we place an adequate number of elaborated basic stacking units in the encoder to enhance the model capacity for extracting sufficient feature information and employ both SA and RA mechanisms within the decoder to effectively improve the prediction accuracy and minimize the computational cost.

III. Proposed Method

To achieve a better trade-off between accuracy and computational efficiency for real-time semantic segmentation tasks,

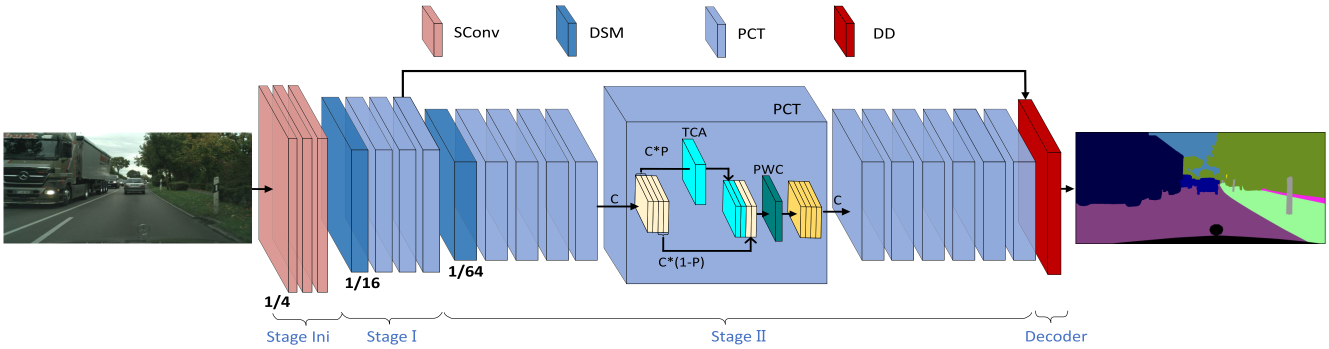


Fig. 3. Overview architecture of the proposed LCNet model for semantic segmentation. Here, Stage 'Ini' denotes the initialization stage and SConv represents a 3 × 3 standard convolution. The backbone includes two encoding stages, each of which starts with a downsampling module (DSM). An elaborately designed block named Partial Channel Transformation (PCT) is applied to stack the backbone. A dual-attention-guided decoder (DD) is designed for final prediction.

we elaborately design a simple yet effective lightweight context-aware network (LCNet) based on the asymmetric encoder-decoder framework as illustrated in Fig. 3.

A. Three-Branch Context Aggregation (TCA)

To capture multiscale contextual information for accurate semantic segmentation with less computational expense, we propose an elaborately-designed three-branch context aggregation (TCA) module as illustrated in Fig. 4. Different from the widely-used yet costly-computed multiscale feature extraction strategies, such as Gaussian or Laplacian image pyramids, feature pyramids in FPN [38], and pyramid pooling modules in PSPNet [39] and LAANet [40], etc., we use a 3 × 3 standard convolution (SConv) to maintain affluent information from the input feature map at the beginning.

The convolutional results are then transmitted to three branches in TCA to capture different scales of contextual information. Specifically, the first branch employs a 3 × 3 depthwise convolution (DwConv) to extract the local contextual information without using the convolution factorization. The second branch apples the dilated convolution to the DwConv with an assigned dilation rate to build the depthwise-dilated convolution (DwDConv) so that a larger receptive field can be generated to extract surrounding context which is potentially correlated to the pixel labels. The DwConv operates only on the corresponding input channel to significantly reduce the computational cost. However, it results in an unrecoverable loss of partial information while stacking multiple DwConv in the backbone since the information of an output channel in DwConv is only derived from one input channel. Therefore, we extend the identity mapping of the convolutional result of the 3 × 3 SConv as the third branch to retain abundant spatial detail, which preserves object boundaries in images and thus avoids significant performance degradation due to the parameter-reduced DwConv.

Finally, the context-aware TCA module jointly gathers three different levels of local features and surrounding context from three branches. The DwDConv layer generates a different scale of feature representation from the DConv layer by assigning a desired dilation rate D for a larger receptive field. The identity mapping implemented by the shortcut connection avoids the sub-optimal problem resulting from DwConv. Note that the output feature maps of our context-aware TCA module remain the same size as input feature maps and thus can be plugged easily into existing dense feature extraction architectures and improve the performance of dense prediction by capturing multiscale contextual information.

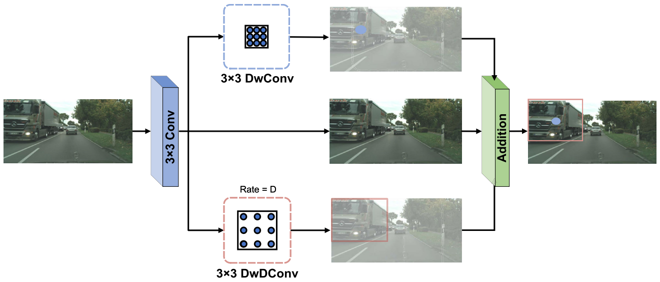


Fig. 4. Structure of the three-branch context aggregation (TCA) module.

B. Partial Channel Transformation (PCT)

Instead of directly utilizing the TCA module as the basic unit to build the backbone network for semantic segmentation, we contemplate a computation-efficient block structure based on a novel convolution strategy of partial channel transformation (PCT) as illustrated in Fig. 2 (e), aiming at achieving an appropriate trade-off between computation cost and feature representation capability. The ResNet bottleneck structure [3] reveals that there are a lot of redundant computations in the full channel convolution. Hence, we split the channels of the input feature map into two parts and apply partial channels for identity mapping while the others are used to extract multiscale context information by the context-aware TCA module. In other words, partial input channels are identified to the output feature maps, preventing the network model from serious gradient vanishing when the model deepens. Besides, the shortcut branch concatenates with the preliminarily-aggregated output results from the TCA branch to restore the channel number as the input. Finally, one single 1 × 1 pointwise convolution is preserved after feature concatenation to enable feature interaction among all channels from two branches.

The PCT block brings two advantages to the proposed model. First, it expands the collected contextual features and lowers the over-fitting risk. Second, it makes the network model more compact and more efficient, resulting in an accurate pixel-level prediction with fewer parameters and less computational cost. The PCT strategy reduces the computing complexity by replacing the dense convolution with a sparse convolution using only a part number of channels. It takes less processing time while the PCT block is placed at the deeper layers of the backbone because the downsampling module (DSM) reduces the resolution of feature maps. However, the DSM is more computationally expensive than the PCT. To make the encoder more efficient with fewer parameters, the depth of the backbone network stacked by the PCT blocks and DSMs should be carefully considered. In this work, affluent context information is able to be captured by our elaborate PCT block that a relatively shallow and lightweight backbone network compared to the existing deep CNNs is built by stacking a finite number of PCT blocks and two DSMs. We investigate an appropriate network depth through extensive ablation experiments in Section IV.

C. Dual-Attention-Guided Decoder (DD)

1) Reverse-Attention Guidance (RG): A confusing region often emerges in the boundary of two different objects, especially in scenarios where there is a complex mixture of foreground and background. Rather than directly predicting the class of the target object like many existing approaches, we develop a compact reverse attention (RA) to guide the learning of differences between objects. As illustrated in Fig. 5, the feature map Y is transmitted from the backbone and compressed by the pointwise convolution $PWC(\cdot)$ according to $Y_c = PWC(Y)$. Then, the compressed feature map Y_c is fed into an attention mask implemented by the Sigmoid function $\mathrm{Sig}(\cdot)$ to convert the compressed map into the range of [0, 1]. The attention-guided map is further converted to its opposite so that the corresponding negative response R_{neg} is generated by:

$$R_{neg}\left(Y_c\right) = -\,\mathrm{Sig}\left(Y_c\right). \qquad (1)$$

Compared to the complex RAN [36] which generates a per-class mask to amplify the reverse-class response in the confused region using three branches and yields the final prediction based on the fused responses from three branches, our RA uses a simpler structure to learn the attention of the target classes and the reverse classes from the input feature Y_c in parallel, highlighting both positive and negative responses in the input feature maps in forward propagation.

The reverse ground truth is learned and highlighted explicitly while the positive response for the ground truth is suppressed by the RA mask. The original positive response is assigned an attention-guided score by the Sigmoid attention mask and is combined with the negative response from the reverse Sigmoid attention implemented by element-wise product with '-1', flipping the sign of the response score of each pixel. Besides, we utilize another pointwise convolution $PWC(\cdot)$ after the negative Sigmoid operation to apply a linear transformation to amplify non-linearity and learn channel-wise correlations and dependencies within the response scores. The reweighted scores are then added to the original Y_c and get the accurate response for semantic segmentation tasks, where the salient difference of objects is learned and highlighted. The complete processing procedure of the reverse-attention guidance (RG) is described as:

$$F_{RG}(Y) = \mathrm{Ups}\left(\mathrm{Sig}\left(Y_c\right) \otimes PWC\big(R_{neg}\left(Y_c\right)\big) + Y_c\right), \qquad (2)$$

where the upsampling operation $\mathrm{Ups}(\cdot)$ is implemented by a fast bilinear interpolation. The upsampling operation is included in the RG branch to restore a 1/16-scale feature map of the input resolution from 1/64-scale feature maps.

Since the RG branch learns the attention of target classes and reverse classes from the input feature Y_c in parallel, it adds more relevant prediction guidance and amplifies the salient boundaries of different objects, and finally makes the decoder more discriminative.

2) Spatial-Attention Guidance (SG): Different from the RG branch, we only use the Sigmoid function $\mathrm{Sig}(\cdot)$ to transform the 1/64 feature maps X from the shallow layers to keep more spatial information by:

$$F_{SG}(X) = \mathrm{Sig}(X). \qquad (3)$$

The shallow features are rich in texture information which retains accurate object boundaries. The Sigmoid function converts these feature responses into probabilities of possible labels for each pixel. Therefore, F_{SG} is essentially a position distribution of different objects in the images.

3) Dual-Attention-Guided Prediction (DP): In the final prediction stage of the DD as illustrated in Fig. 5, the exact object position provided by F_{SG} and the class response provided by F_{RG} are combined by the element-wise product as:

$$F(X, Y) = F_{SG}(X) \otimes F_{RG}(Y), \qquad (4)$$

where each element in the fused feature map $F(X, Y)$ denotes the correlation degree between its location and a certain class. A 3 × 3 depthwise separable convolution is applied to the $F(X, Y)$ for further integrating local information. Eventually, the decoder DD is elaborately designed using the dual-attention mechanism to aggregate different levels of features to make the prediction of pixel class more accurate. Additionally, we also illustrate some examples of visualization of attention maps in each layer of the dual-attention-guided decoder (DD), as shown in Fig. 6.

D. Architecture of LCNet

A simple yet effective lightweight context-aware network (LCNet) for real-time semantic segmentation tasks is finally built on the asymmetric encoder-decoder framework and the structure details of LCNet$_{3\_11}$ are summarized in Table I.

The encoder is stacked by three stages, including the common initialization stage 'Ini'. The 'Ini' stage is composed of three 3 × 3 SConvs, which are essential to extract sufficient low-level features for the following two encoding stages. We expand the input image channels from '3' to '32' for collecting sufficient texture information from the input images and set the stride of the first convolution in the 'Ini' stage to be '2', significantly reducing the calculation cost of subsequent stages and improving the computing speed of the network.

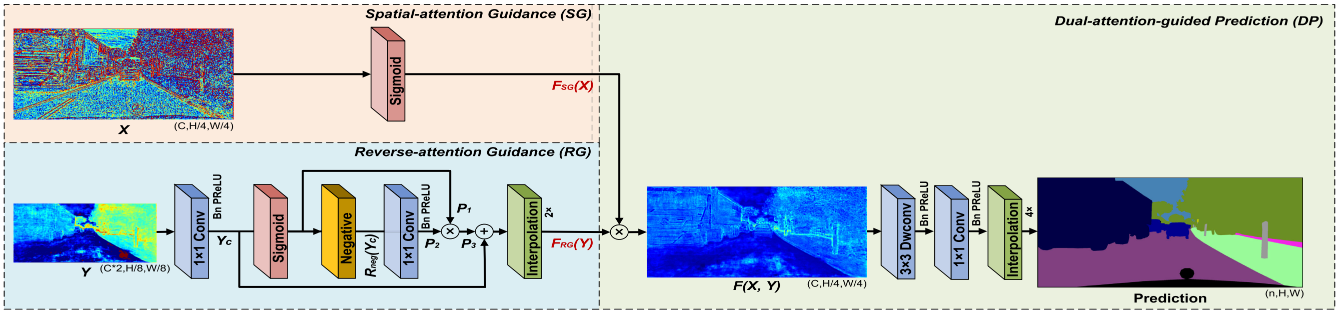


Fig. 5. The overall structure of the dual-attention-guided decoder (DD). Here, 'X' denotes the input feature map from shallow layers while 'Y' represents the feature map with larger receptive field from deep layers. 'Negative' means the multiplication operation with '-1'. 'n' represents the category number.

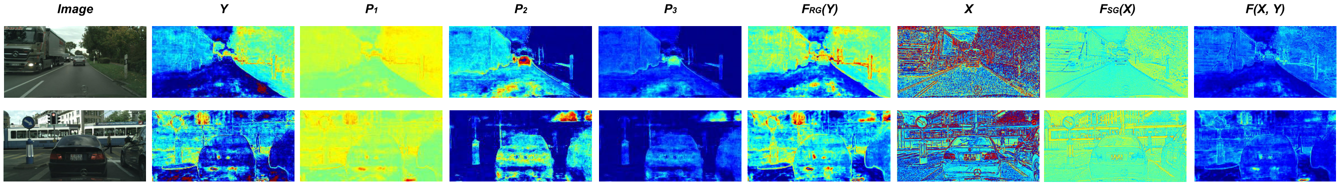


Fig. 6. Visualization of feature maps of various layers of DD. Here, P_1 signifies a positive attention-guided result of the input feature maps Y_c, while P_2 demonstrates the attention of non-strongly-expressed information by using a negative operation to suppress the positive response of Y_c. P_3 refers to the element-wise product of P_1 and P_2. $F_{RG}(Y)$ showcases the merged information of P_3 and Y_c after undergoing bilinear interpolation. $F_{SG}(X)$ represents the resultant feature maps from the spatial-attention branch for processing shallower layers. Finally, $F(X, Y)$ denotes the fused feature maps aimed at enhancing the prediction performance.

TABLE I

OVERALL STRUCTURE OF OUR PROPOSED LCNET$_{3\_11}$

Stage		Input	Layer	Output	D*	S*	Output size
Encoder	Ini	**Image** (3×H×W)	SConv	i_s1	1	2	32×H/2×W/2
		i_s1	SConv	i_s2	1	1	32×H/2×W/2
		i_s2	SConv	i_s3	1	1	32×H/2×W/2
	I	i_s3	DSM	I_ds	1	2	64×H/4×W/4
		I_ds	PCT×3	I_p3	2	1	64×H/4×W/4
	II	I_p3	DSM	II_ds	1	2	128×H/8×W/8
		II_ds	PCT×2	II_p2	4	1	128×H/8×W/8
		II_p2	PCT×2	II_p4	8	1	128×H/8×W/8
		II_p4	PCT×2	II_p6	16	1	128×H/8×W/8
		II_p6	PCT×5	II_p11	32	1	128×H/8×W/8
Decoder		I_p3, II_p11	DD	**Prediction**	-	-	n×H×W*

[*] D: Dilation rate; S: Stride; n: Category number; H: Height; W: Width.

As listed in Table I, Stages I and II start by using a downsampling module (DSM), resulting in a much smaller resolution with less costly computation. We exploit an effective downsampling structure referred to [13], which is integrated with a max pooling and a SConv, leading to more channels with a smaller volume in the output feature map. The encoder extracts multiscale context information by stacking an appropriate number of PCT blocks with gradually-increased dilation rates in each stage. Whereas, the appropriate number of encoding stages with a suitable number of PCT blocks should be determined properly. Thus, we conduct comprehensive experiments in Section IV to find out the optimum model architecture with a proper model size for achieving a decent trade-off between computing efficiency and segmentation accuracy. The experimental results confirm that a two-stage encoder that combines with the initialization stage performs better since it extracts high hierarchical contextual features on a smaller model size. Two optimized versions LCNet$_{3\_7}$ and LCNet$_{3\_11}$ are respectively emphasized for computing-efficiency-oriented and accuracy-oriented semantic segmentation tasks in this paper.

In the decoder, both positive attention and reverse attention are used for adaptive learning of the weights of multiscale features from the encoder. On the one hand, we introduce reverse attention to highlight the boundary between the ground-truth class and the non-ground-truth classes from the semantic information extracted by the deep layers in Stage II. On the other hand, a long-range connection from the output feature maps of the shallow layer in Stage I to the reverse-attention-guided prediction from deep layers in Stage II is applied for recovering spatial information in the decoder. The long-range connection between feature maps in different resolutions has proved to be of great significance for accurate boundary location detection. However, the widely-used long-range connection strategy by fusing the downsampled images in many existing state-of-the-art approaches, such as DABNet [19], MSCFNet [41], and LMFFNet [21], is not employed for feature enhancement in our work due to its expensive computing cost.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we comprehensively evaluate the performance of the proposed LCNet on three benchmark datasets for urban driving scenes: Cityscapes [8], CamVid [42], and BDD100K [43]. A series of ablation experiments on the Cityscapes validation set are conducted to estimate each key component of LCNet. We report the overall performance in terms of parameter size, accuracy (mIoU), inference speed (fps), and computational complexity (FLOPs) based on the above three benchmark datasets. Finally, we compare the performance of the proposed LCNet with multiple existing state-of-the-art real-time semantic segmentation networks.

A. Datasets

1) Cityscapes: As one of the most challenging semantic segmentation benchmarks, the Cityscapes dataset contains diverse high-resolution street scene images across 50 different European cities. Specifically, 5,000 fine-annotated images are separated into three groups: training set (2,975 images), validation set (500 images), and testing set (1,525 images). The Cityscapes dataset contains 30 class labels while only 19 semantic categories are considered for training and validation.

2) CamVid: As another popular dataset of urban street scenes in driving, CamVid contains a total number of 701 densely-annotated images with a resolution of 720×960. Likewise, these images are divided into three sets: 367 images for training, 101 images for validation, and 233 images for testing. The ground-truth images of CamVid are annotated to 11 semantic categories for model training.

3) BDD100K: BDD100K is a large-scale driving video dataset with 100K videos. Specifically, 8,000 images with fine-grained pixel-level annotations are sampled and applied for semantic segmentation evaluation, 7,000 of which are used for training and 1,000 of which are applied for validation. The class labeling of the BDD100K is compatible with Cityscapes while the BDD100K is more challenging due to its scenario diversity in geography, weather, time, scene types, and so on.

B. Implementation Protocol

All training experiments are performed on an RTX 3090 GPU with CUDA 11.4 and cuDNN V8 using the Pytorch platform. Specifically, our LCNet is trained from scratch using the initialization manner [44]. Stochastic gradient descent (SGD) [45] optimizer is employed as the training optimization strategy. The initial learning rate is set as $4.5e-2$ and the 'poly' learning rate decay policy is adopted with a power of 0.9. Besides, the momentum and weight decay are set to 0.9 and $1e-4$, respectively. In particular, for CamVid dataset, the weight decay is set to $5e-4$. For all datasets, we adopt a batch size of 8 and set a maximum of 1000 epochs for training.

For the data augmentation, the random horizontal flip, mean subtraction and random scale are performed on the training images in the preprocessing phase, where the random scale factors are respectively set to {0.75, 1.0, 1.25, 1.5, 1.75, 2.0}. For the Cityscapes dataset, we randomly crop the training data into two resolutions of 512×1024 and 1024×1024. For the CamVid dataset, we evaluate the model performance under two resolutions of 720×960 and 360×480, respectively. For the BDD100K dataset, we directly train and validate in its original image resolution without any clipping. Moreover, the online hard example mining (OHEM) [46] scheme is employed to alleviate the category imbalance problem. The OHEM loss ($OL_{i,j}$) of the pixel located at coordinate (i, j) can be respectively assigned according to:

$$OL_{i,j} = \begin{cases} CL_{i,j}, & CL_{i,j} \geq \min\left(T, \mathrm{Top}(CL, k)\right), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $CL_{i,j}$ represents the cross-entropy loss of the predicted output to the corresponding pixel class at coordinate (i, j) in the ground truth and the threshold T is empirically set to 0.7 to filter easy examples while retaining hard examples. $\mathrm{Top}(CL, k)$ indicates the k largest values within the cross-entropy loss matrix CL. Typically, k is set to 256 to ensure at least $k = 256$ pixels are used for gradient backpropagation. Besides, for training the network models more properly, the class weight W_{class} of each object category in the datasets is set to be $[ln\,(c + P_{class})]^{-1}$, where P_{class} is the label class distribution of each pixel in the image samples and c is an adjustable hyper-parameter which is set to be 1.10 in this work.

C. Comprehensive Evaluation Metric

The main evaluation indices of real-time semantic segmentation tasks include model size in terms of parameter amount, inference speed, and segmentation accuracy. In previous works, researchers generally tended to adopt the method of subjective evaluation to select the optimal model. In this work, we attempt to use another objective performance metric to comprehensively evaluate the overall performance of network models through three critical indicators of accuracy, parameter amount, and inference speed. The critical method [47] comprehensively measures the objective weight of indicators based on the contrast strength of evaluation indicators and the conflict between indicators. By combining the three critical indices, we propose a new fusion evaluation index, namely, the critical index (CI):

$$CI = w_1 \times p_{norm} + w_2 \times s_{norm} + w_3 \times a_{norm}, \tag{6}$$

where the normalized indicators can be respectively calculated by $p_{norm} = \frac{p_{max}-p}{p_{max}-p_{min}}$, $s_{norm} = \frac{s-s_{min}}{s_{max}-s_{min}}$ and $a_{norm=\frac{a-a_{min}}{a_{max}-a_{min}}}$. Specifically, p_{max} and p_{min} respectively denotes the maximal or minimal parameter amount; s_{max} and s_{min} indicate the fastest inference speed and the slowest speed; a_{max} denotes the best accuracy while a_{min} indicates the worst accuracy. The objective weight w_i of each index can be calculated by $w_i = C_i / \sum_{j=1}^{3} C_j$, where $C_i (i \in 1, 2, 3)$ can be calculated by:

$$\begin{cases} C_i = S_i \times R_i, \\ S_i = \sigma\left(I_i\right), \\ R_i = \sum_{j=1}^{3}\left(1 - r_{ij}\right), \\ r_{ij} = corr\left(I_i, I_j\right). \end{cases} \tag{7}$$

The *i-th* index $I_i (i \in 1, 2, 3)$ denotes the normalized indicator of the parameter amount p_{norm}, inference speed s_{norm}, or accuracy a_{norm} within all models for comparison. $\sigma\left(\cdot\right)$ denotes the standard deviation of the corresponding index and $corr\left(\cdot\right)$ indicates the correlation between two normalized indices I_i and I_j $(i, j \in 1, 2, 3)$. The value range of CI is $[0, 1]$. A higher CI value stands for the better trade-off of network performance between the involved indices, i.e., parameter amount, inference speed, and accuracy.

TABLE II

EXPERIMENTAL RESULTS OF DIFFERENT BASIC UNITS ON THE CITYSCAPES VALIDATION SET

Module	mIoU(%)	Speed (fps)	Params. (M)	FLOPs (G)
ResNet [3]	50.60	297.1	0.20	4.68
DAB [19]	71.01	176.4	0.90	9.62
SS-nbt [20]	71.17	128.2	0.75	10.74
SEM_B [21]	71.60	174.3	0.89	12.27
TCA (ours)	71.72	221.9	1.05	14.61
ResNet-TCA	60.67	191.8	0.21	4.77
DAB-TCA	71.04	170.2	0.91	12.36
SS-nbt-TCA	67.68	216.2	0.61	11.81
SEM_B-TCA	72.04	145.7	1.15	15.10
PCT (ours)	71.51	195.7	0.46	7.96

TABLE III

EXPERIMENTAL RESULTS OF ENCODERS WITH DIFFERENT DEPTHS BUILT BY 9 PCT BLOCKS ON THE CITYSCAPES VALIDATION SET

Module	mIoU(%)	Speed (fps)	Params. (M)	FLOPs (G)
LCNet$_9$	63.25	210.8	0.17	7.39
LCNet$_{3\_6}$	71.51	195.7	0.46	7.51
LCNet$_{3\_3\_3}$	71.52	194.5	1.10	7.72
LCNet$_{3\_2\_2\_2}$	72.66	190.9	3.16	7.99

D. Ablation Studies

1) Performance of Different Basic Units: The effectiveness of different basic units for stacking the baseline framework of the LCNet is respectively investigated in this work. We then summarize the experimental results for indicating their contributions in terms of accuracy, speed, parameter amount, and floating-point operation (FLOPs) in Table II.

ResNet bottleneck [3] results in the smallest model size, the fewest FLOPs, and the fastest speed at the expense of significant accuracy loss. In contrast, the TCA module leads to the largest model size and larger FLOPs number since no remedial action for the three-branch architecture has been taken. Nevertheless, the backbone network turns out to be fairly compact when the TCA module is applied as one of the two branches of the PCT block, letting only a proportion (e.g., $P = 50\%$) of the input channels pass through the PCT. Although the introduction of additional pointwise convolutions in PCT makes the computation a bit slower than TCA, the parameter number (0.46 Million) and FLOPs number (7.96 Giga) of PCT are both smaller than those of TCA because only a part of the channels of the input feature maps are convoluted in PCT. PCT achieves a smaller model size and computational cost, faster processing speed, and higher accuracy than the models using basic units of the DAB module [19] and SS-nbt module [20].

Moreover, we have integrated the TCA module into the basic stacking unit to replace the core convolutional layers of ResNet, DABNet, LEDNet, and LMFFNet as illustrated in Fig. 2, respectively, to evaluate the impact of TCA on existing network models. Typically, the TCA module can enhance various performance metrics in these models as demonstrated in Table II. However, substituting the core convolutional layers of the original SS-nbt module with the TCA incurs a decline in both the accuracy and parameter amount of SS-nbt-TCA, due to the possession of a greater number of convolutional layers in the original SS-nbt. The PCT block, incorporating the TCA module as one branch, achieves a reduction in parameters, albeit with a slight accuracy decrease, as only half of the input feature map channels undergo processing for feature extraction. Note that the channel split operation and concatenation in PCT may result in increased computation and slower inference speed when compared to the TCA. In real-time semantic segmentation tasks for mobile applications, the PCT emerges as a preferable choice for building the backbone network to strike an optimal balance between accuracy and computing efficiency.

2) Ablation Studies on Encoder Capacity: The encoder structure in different depths with a different number of convolution stages can determine the overall performance of the whole network. To achieve a lightweight network model, the backbone preferably includes no more than four stages (excluding the initialization stage). Therefore, we construct the backbone network with maximal four convolution stages as LCNet$_{S1\_S2\_S3\_S4}$, where $S1$, $S2$, $S3$, and $S4$ indicate the number of PCTs in stages I, II, III, and IV of the encoder, respectively. Note that each encoding stage starts with a DSM and is followed up by a set of PCT blocks.

To evaluate the impact of the encoder depth on the network performance, we designate 9 PCT blocks which are distributed to different stages in the encoder in four cases, and summarize the experimental results in Table III. Specifically, the LCNet$_9$ indicates that the encoder contains only one encoding stage which includes 1 DSM and 9 subsequent successively-connected PCTs for feature extraction. LCNet$_{3\_6}$ combines two stages for the encoder where the first stage $S1$ includes 1 DSM and 3 subsequent PCTs while the second stage $S2$ includes 1 DSM and 6 PCTs. Likewise, LCNet$_{3\_3\_3}$ and LCNet$_{3\_2\_2\_2}$ include three and four stages, respectively, with the corresponding number of PCTs distributed in each stage. Fewer stages always lead to fewer model parameters and FLOPs as summarized in Table III since fewer convolution kernels are involved in the computation. The four-stage LCNet$_{3\_2\_2\_2}$ results in the most parameters and the FLOPs but the best accuracy performance when compared to other shallow structures. In deeper LCNets, more DSMs with a larger channel width are dragged on the inference speed and model size. In contrast, the accuracy of lighter network LCNet$_9$ is much worse. The two-stage LCNet$_{3\_6}$ and the three-stage LCNet$_{3\_3\_3}$ obtain a better trade-off between accuracy and computing efficiency while LCNet$_{3\_6}$ is more compact with fewer parameters.

Obviously, deeper semantic features are important for improving the segmentation accuracy. However, the deep networks lead to a larger model size as well. So, we further investigate the performance of the mediate-size two-stage encoder structure for LCNet$_{S1\_S2}$. We used the index CI defined in (6) to select the most cost-effective regimen for a lightweight yet accurate semantic segmentation network. As depicted in Fig. 7, the two-stage encoder performs better with a higher CI score when $S1 = 3$ in both cases with a fixed $S2 = 7$ and $S2 = 11$. The peak point of the CI curve indicates that the network model achieves a better comprehensive performance of model size, accuracy and speed. Specifically, the

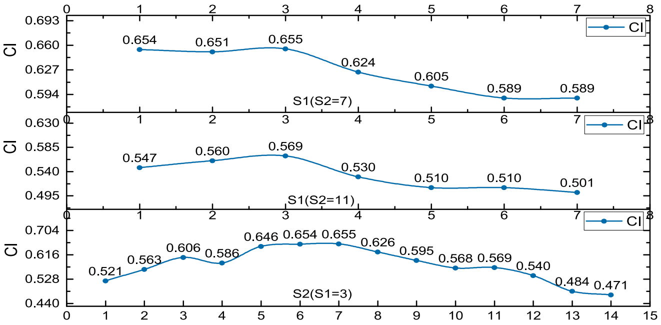


Fig. 7. Variation of CI values with respect to different S1 and S2. Up: Fixed S2 (S2 = 7) and varied S1; Middle: Fixed S2 (S2 = 11) and varied S1; Down: Fixed S1 (S1 = 3) and varied S2.

TABLE IV

EXPERIMENTAL RESULTS WITH DIFFERENT S2 LENGTHS IN TWO-STAGE LCNet_{S1_S2} ON THE CITYSCAPES VALIDATION SET

S1	S2	mIoU(%)	Speed (fps)	Params. (M)	FLOPs (G)	CI
3	6	71.51	195.7	0.46	7.51	0.654
	7	73.00	185.2	0.51	7.96	0.655
	8	73.65	163.4	0.57	8.42	0.626
	9	73.51	155.2	0.62	8.87	0.595
	10	73.70	144.4	0.68	9.33	0.568
	11	75.07	136.8	0.74	9.79	0.569
	12	75.04	129.5	0.79	10.24	0.540

model LCNet_{3_7} (i.e., $S1 = 3$, $S2 = 7$) and LCNet_{3_11} (i.e., $S1 = 3$, $S2 = 11$) are two potential solutions for high-speed-oriented and high-accuracy-oriented applications, respectively.

Besides, we summarize the experimental results for the two-stage LCNet_{S1_S2} with a fixed $S1 = 3$ in stage I and a variant $S2$ of PCTs in stage II in Table IV. The LCNet_{3_7} improves accuracy by 1.5% compared to LCNet_{3_6}. As the value of $S2$ increases, there is an observed improvement in accuracy (mIoU), albeit at the cost of a larger model size, more FLOPs computations and slower inference speed. For instance, LCNet_{3_11} requires 0.74 M relatively high parameters at only 136.8 fps inference speed. The inference speed of LCNet_{3_12} has been reduced by one-third with about 71% more parameters compared to LCNet_{3_6}. However, it is still meaningful to build a deeper encoder for certain practical applications, given the substantial improvement in accuracy performance it yields.

3) Impact of the Channel Width of PCT: In our work, the channel width is changed by the DSM at the initialization stage and then is doubled by the DSM in each encoding stage. Besides, only partial channels of the input feature maps are transferred to the TCA branch in the PCT for convolution operation according to the PCT structure. Therefore, there are two hyper-parameters in the encoder determining the final channel width of each stage in the encoder, i.e., the channel number at the initialization stage C_{Ini} and the proportion P of channel number of input feature maps passing through the TCA branch of PCT.

TABLE V

EXPERIMENTAL RESULTS OF DIFFERENT CHANNEL NUMBER C_{Ini} AT INITIALIZATION STAGE AND CHANNEL PROPORTION P IN PCT BLOCKS ON THE CITYSCAPES VALIDATION SET

C_{Ini}	P	mIoU(%)	Speed (fps)	Params. (M)	FLOPs (G)
16	0.5	64.53	198.0	0.13	2.11
32	0.25	68.98	190.6	0.29	5.60
32	0.5	73.00	185.2	0.51	7.96
32	1.0	75.15	183.3	1.39	17.23
64	0.5	75.91	130.8	2.01	30.89

We investigate the impact of channel width determined by C_{Ini} and P on the LCNet based on a two-stage model LCNet_{3_7}. We adjust the values of C_{Ini} and P to observe the variation of network capability and find out the appropriate C_{Ini} and P for building an LCNet model with a better balance between computing efficiency and accuracy. As the experimental results summarized in Table V, the accuracy (mIoU) increases when the C_{Ini} is enlarged. Approximately 75.91% mIoU can be achieved while C_{Ini} is set to 64 at the expense of a significantly-declined inference speed because a large C_{Ini} in the initialization stage amplifies the volume of feature maps of every following layer in the encoder. A faster inference speed, a smaller model size, and fewer FLOPs number can be achieved while C_{Ini} or P decreases. For a fixed C_{Ini} (e.g., $C_{Ini} = 32$), the accuracy increases gradually while P increases from 0.25 to 1. P controls the channel width of TCA and the computation time spent on convolution. Therefore, both C_{Ini} and P should be increased if a higher segmentation accuracy is strongly desired while a smaller C_{Ini} and P should be assigned if a faster inference speed with an ultra-small model size is preferred. And C_{Ini} has a greater impact on the performance of the whole network than P. In this work, we set $\{C_{Ini} = 32, P = 0.5\}$ for the two-stage encoder as the baseline to achieve a better trade-off between accuracy and computation efficiency for mobile application scenarios.

4) Ablation Study for the DD: To verify the effectiveness of the dual-attention (DA) mechanism in the decoder, we have investigated different strategies, including element-wise addition, point-wise convolution, and global attention upsample (GAU) [32] for yielding a lightweight yet efficient decoder. Element-wise addition indiscriminately blends two inputs for feature fusion while point-wise convolution enhances partial channel representation through inductive bias. In contrast, the attention mechanism utilizes the input itself to improve the feature expression and thus introduces higher adaptability to the network. The GAU requires more computation operations to collect global information. In the DA strategy, it is possible to combine feature maps from different layers of the encoder since there are two inputs (X, Y) in $\text{DD}_{X,Y}$ for decoding the final output result. As the experimental results demonstrated in Table VI, GAU achieves higher accuracy compared to the element-wise addition and point-wise convolution, but the speed declines slightly with more model parameters and higher FLOPs. Compared to GAU, the DA strategy fuses the two levels of output features from the encoder and achieves a faster processing speed and higher segmentation accuracy while consuming 20% fewer model parameters.

We further investigate the performance of DD by blending the feature maps from different levels. Since the encoder LCNet_{S1_S2} includes one initialization stage 'Ini' and two

TABLE VI

Experimental Results of Different Feature Fusion Schemes in the Decoder on the Cityscapes Validation Set

Scheme	mIoU(%)	Speed (fps)	Params. (M)	FLOPs (G)
STDC [48] $^{\textit{CVPR2021}}$	70.64	169.3	0.52	7.82
UFAM [49] $^{\textit{ArXiv2022}}$	70.44	172.8	0.52	8.56
CWF [50] $^{\textit{TNNLS2023}}$	72.87	179.7	0.55	9.41
CABD [51] $^{\textit{TPAMI2021}}$	73.35	135.4	0.53	8.55
AFF [52] $^{\textit{TVC2023}}$	72.33	169.3	0.51	8.01
Addition	71.93	186.6	0.51	7.93
PwConv	72.27	185.7	0.51	8.27
GAU [32]	72.83	181.8	0.67	10.09
DA	73.00	185.2	0.51	7.96

TABLE VII

Experimental Results of Using Different Strategies for the Decoder on the Cityscapes Validation Set

Decoder	mIoU(%)	Speed (fps)	Params. (M)	FLOPs (G)
None (Baseline)	71.92	196.7	0.50	7.81
APN [20]	71.15	160.7	1.01	7.82
MAD [21]	72.87	183.6	0.51	7.94
$\mathrm{DD}_{Ini,II}$	71.96	184.4	0.50	7.98
$\mathrm{DD}_{I,II}$	73.00	185.2	0.51	7.96
$\mathrm{DD}_{I,II}+\mathrm{DD}_{Ini,O}$	73.13	172.8	0.52	8.31

encoding stages of I and II, two of three outputs from these three stages can be combined to the inputs X and Y for the $\mathrm{DD}_{X,Y}$. We evaluate structures of $\mathrm{DD}_{Ini,II}$, $\mathrm{DD}_{I,II}$ and the two-stage decoder "$\mathrm{DD}_{I,II} + \mathrm{DD}_{Ini,O}$", where O denotes the output of the former $\mathrm{DD}_{I,II}$. As the experimental results summarized in Table VII, the accuracy performance of three different DD solutions is better than the baseline model without connecting any decoder. $\mathrm{DD}_{Ini,II}$ has a slight improvement in accuracy but a significant drop in inference speed compared to the baseline. $\mathrm{DD}_{I,II}$ results in 1.08% mIoU improvement while incurring a marginal increase of merely 0.01 Million parameters and additional 0.15 Giga FLOPs (GFLOPs). When multiple DDs are combined for prediction, such as the "$\mathrm{DD}_{I,II} + \mathrm{DD}_{Ini,O}$" strategy, the feature maps from the 'Ini' stage provide negligible assistance for the accuracy improvement but introduce more model parameters and computational operations, making the inference speed decline significantly compared to using a single $\mathrm{DD}_{I,II}$. Hence, the single $\mathrm{DD}_{I,II}$ is preferred to be our final solution for building the decoder.

Additionally, we also analyzed the DA mechanism through examples of visualization of attention maps as shown in Fig. 6. From the intermediate feature maps produced by the RG branch, P_1 exhibits a similar expression to Y_c, while P_2 captures the non-strongly-expressed information and supresses the positive response in Y_c through the Negative operation. Consequently, the output of the RG branch, $F_{RG}(Y)$, contains a greater number of expressive features compared to Y_c. By fusing the details from $F_{SG}(X)$ and the richer semantic features from $F_{RG}(Y)$, the output feature map $F(X, Y)$ achieves a higher prediction confidence for semantic segmentation.

E. Inference Speed With Varying Resolutions and GPUs

To comprehensively investigate the computing effiency in terms of inference speed of our LCNet, we have deployed the two versions of LCNet on the powerful RTX 3090 GPU and the Jetson Xavier NX platform with an embedded GPU with or without TensorRT acceleration engines, respectively. The Jetson Xavier NX is equipped with a 6-core Carmel ARM CPU, a 384-CUDA-core GPU, 8 GB RAM, and two deep-learning accelerator engines. Specifically, Fig. 8 (a) shows the inference speeds of LCNet_{3_7} and LCNet_{3_11} with floating 32-bit operation on the embedded GPU while Fig. 8 (b) illustrates the corresponding inference speeds evaluated on RTX 3090. The experimental findings unequivocally illustrate that TensorRT delivers improved speed when applied to input images of diverse resolutions on both types of GPUs. When considering 128 × 256 resolution images, TensorRT yields a substantial speed gain (Δ) of 216.9 fps on the embedded GPU. Likewise, the utilization of TensorRT proves highly advantageous in accelerating the speed of low-resolution images on the 3090 GPU as well. However, it is observed that the speed gains for high-resolution images are not as substantial as those achieved for low-resolution images. The acceleration from TensorRT is less pronounced for high-resolution inputs mainly because the high-resolution images have larger spatial dimensions, resulting in more computations and increased memory space required for computing. Besides, high-resolution images may push the hardware (RTX 3090 or embedded GPU on Xavier NX) to its limits, limiting the speed gains achievable.

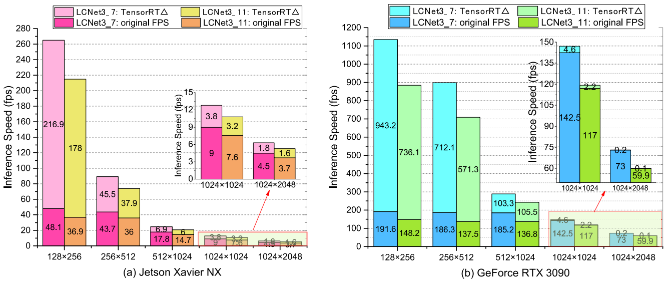


Fig. 8. Inference speed of LCNet on Jetson Xavier NX (a) and RTX 3090 GPU (b) at different input resolutions with or without TensorRT acceleration.

F. Performance Comparison With the State-of-the-Arts

In this section, we evaluate the overall performance of the proposed LCNet model based on three dataset benchmarks: Cityscapes, CamVid, and BDD100K. Two optimal model versions (i.e., the computing-efficiency-oriented LCNet_{3_7} and the accuracy-oriented LCNet_{3_11}) without pretrain and TensorRT acceleration for real-time semantic segmentation are evaluated and further compared with multiple existing state-of-the-art semantic segmentation approaches.

1) Performance on Cityscapes: We estimate the existing state-of-the-art semantic segmentation networks and our LCNet models in our local servers on the Cityscapes dataset. The comprehensive comparison with existing state-of-the-art semantic segmentation networks is illustrated in Fig. 1. Table VIII summarizes the quantitative comparison of the experimental results on the Cityscapes dataset.

Our LCNet_{3_7} achieves 73.8% mIoU with only 0.51 million parameters while LCNet_{3_11} gains a higher mIoU of

TABLE VIII

PERFORMANCE COMPARISON OF LCNETS AGAINST EXISTING SEMANTIC SEGMENTATION NETWORKS ESTIMATED ON THE CITYSCAPES DATASET

Method	**Source**	**Pretrain**	**Input Size**	**mIoU (%) ↑**		**Params.↓**	**Speed ↑**	**Devices**	**FLOPs (G)[a]↓**	**CI ↑**
				val.	**test**	**(M)**	**(fps)**			
SegNet [17]	TPAMI2017	ImageNet	360×640	57.8	56.1	29.5	74.9	3090	652.5	0.060
ENet [13]	ICCV2015	No	360×640	59.0	58.3	**0.36**	83.8	3090	8.7	0.437
SQNet [53]	NeurIPS2016	ImageNet	1024×2048	59.9	59.8	16.3	18.7	3090	288.2	0.212
ESPNet [54]	ECCV2016	No	512×1024	60.0	60.3	**0.36**	**292.0**	3090	**6.9**	0.693
ESPNet V2 [24]	CVPR2019	No	512×1024	66.2	66.4	1.3	134.8	3090	7.4	0.617
CGNet [55]	TIP2016	No	1024×2048	63.5	64.8	0.49	53.0	3090	14.0	0.512
ICNet [56]	ECCV2018	ImageNet	1024×2048	-	69.5	7.8	24.4	3090	14.2	0.477
EDANet [57]	MMAsia2019	No	512×1024	68.1	67.3	0.68	161.0	3090	17.9	0.667
LEDNet [20]	ICIP2019	No	512×1024	70.6	69.2	0.95	86.5	3090	23.0	0.617
DABNet [19]	BMVC2019	No	1024×2048	69.6	70.1	0.76	60.6	3090	20.9	0.606
ESNet [25]	PVCR2019	No	512×1024	70.4	70.7	1.66	112.3	3090	48.7	0.661
SwiftNet [58]	CVPR2019	ImageNet	1024×2048	75.4	75.5	11.8	39.9	1080Ti	52	0.549
DFANet [59]	CVPR2019	ImageNet	1024×1024	-	71.3	7.8	100.0	1080Ti	-	0.588
MiniNet-v2 [15]	TROBOT2020	No	512×1024	-	70.5	0.5	143.8	3090	25.8	0.704
AGLNet [60]	ASC2019	Coa.Cityscapes	512×1024	-	71.3	1.12	52.0	1080Ti	-	0.613
RTHP-SIS [61]	TITS2020	ImageNet	448×896	74.4	73.6	6.2	51.0	TitanX	-	0.592
MSCFNet [41]	TITS2021	No	512×1024	-	71.9	1.15	50.0	TitanXP	-	0.620
EACNet [62]	SPL2021	No	512×1024	-	74.2	1.10	133.5	3090	28.6	0.749
EPRNet [28]	TITS2021	No	512×1024	-	73.9	0.9	128.7	3090	30.2	0.741
STDC2-Seg50 [48]	CVPR2021	ImageNet	512×1024	74.2	73.4	16.1	119.0	3090	59.6	0.549
NDNet-18 [63]	TNNLS2022	ImageNet	512×1024	-	76.5	18.7	160.8	3090	26.8	0.616
HoloSeg [64]	ICRA2022	ImageNet	512×1024	-	76.2	16.6	153.5	3090	34.7	0.627
DDRNet-23-slim [65]	TITS2022	ImageNet	1024×2048	77.8	77.4	5.7	155.8	3090	36.4	0.774
LMFFNet-3-8 [21]	TNNLS2023	No	512×1024	74.9	75.1	1.4	130.9	3090	33.3	0.757
PIDNet-S [66]	CVPR2023	No	1024×2048	**78.8**	**78.6**	7.6	99.4	3090	23.79	0.712
PIDNet-S [66]	CVPR2023	No	512×1024	76.3	75.5	7.6	116.5	3090	23.79	0.696[b]
RegSeg [67]	CVPR2023	No	1024×2048	78.5	78.3	3.34	62.7	3090	19.56	0.716
RegSeg [67]	CVPR2023	No	512×1024	77.9	77.6	3.34	86.7	3090	19.56	0.750[b]
LCNet$_{3\_7}$ (ours)	This work	No	512×1024	73.0	73.3	0.51	185.2	3090	15.9	**0.795**
LCNet$_{3\_7}$ (ours)	This work	No	1024×1024	73.5	73.8	0.51	142.5	3090	15.9	0.758
LCNet$_{3\_11}$ (ours)	This work	No	512×1024	75.1	74.3	0.74	136.8	3090	19.6	0.758
LCNet$_{3\_11}$ (ours)	This work	No	1024×1024	**75.6**	**75.8**	0.74	117.0	3090	19.6	0.762

[a] FLOPs normalized to images at the resolution of 1024×1024.

[b] CI values based on the updated maximum and minimum mIoU values and Speeds (fps).

75.8% with 0.74 million parameters at the image resolution of 1024 × 1024. The smaller model LCNet$_{3\_7}$ outperforms other approaches in terms of the better trade-off between accuracy and inference speed. The existing ultra-lightweight networks such as ENet [13], ESPNet [54], and CGNet [55] generate fewer parameters (<0.5 M) but their comprehensive performance indicated by CI is unsatisfactory. Compared to MiniNet-v2 [15], LCNet$_{3\_7}$ increases by 3.3% in mIoU and 40 fps in speed with a similar number of parameters. Compared with the latest model MSCFNet [41], LCNet$_{3\_7}$ achieves more than 3× faster speed and approximately 1.9% higher testing accuracy by using only half the parameters. The deeper version LCNet$_{3\_11}$ chases a higher accuracy improvement but requires 0.23 M more parameters, which is still maintaining a good trade-off in all respects. When compared to EPRNet [28] and EACNet [62], LCNet$_{3\_11}$ reduces by 15% parameters and nearly 30% GFLOPs while achieving a higher accuracy, which is more favorable to resource-constrained applications.

Without great regularization in pretraining using ImageNet [16], our LCNet$_{3\_11}$ remains >0.7% mIoU improvement over RTHP-SIS [61] based on a smaller mode size with only one-ninth of the parameters. With the same input resolution of 512×1024, LCNet$_{3\_11}$ still maintains an accuracy advantage over many existing lightweight networks such as ESNet [25], LEDNet [20], and EDANet [57], increasing by 3.6%, 5.1%, and 7.0% mIoU, respectively. Compared to the AGLNet [60] with additional training data, LCNet$_{3\_11}$ achieves improved accuracy by 3.0% mIoU. The NDNet-18 [63], HoloSeg [64], DDRNet-23-slim [65], PIDNet-S [66] and RegSeg [67] all achieve significantly high accuracy (>76.0% mIoU) at the expense of increased parameter amount, consequently leading to slightly smaller CI values compared to our model. The LMFFNet-3-8 achieves a smaller CI value compared to LCNet$_{3\_7}$ at the same input image resolution of 512 × 1024. The LCNet$_{3\_7}$, in a smaller size, attains an optimal accuracy-efficiency balance, as indicated by its highest CI value. The larger LCNet$_{3\_11}$ obtains higher computational efficiency with fewer GFLOPs at a resolution of 512×1024 compared to LMFFNet. Although the three-branch network in PIDNet [66] can extract different kinds of features for accuracy improvement, it also leads to lower inference speed due to more model parameters and computing operations. RegSeg [67] stacks more dilated blocks into a deeper backbone for getting a higher accuracy and it can run faster in our RTX 3090 GPU. However, RegSeg [67] runs still much slower compared to our LCNets with the same input image size of 512 × 1024 due to its lower computing efficiency.

Moreover, we report the results for each category in detail in Table IX. LCNet is superior to many existing networks in most categories while maintaining an extremely lightweight structure. These experimental results demonstrate that the proposed LCNet model effectively encodes multiscale features

TABLE IX

PRE-CLASS RESULTS (%) OF DIFFERENT SEGMENTATION MODELS ON THE CITYSCAPES TEST SET

Method	Roa	Sid	Bui	Wal	Fen	Pol	TLi	TSi	Veg	Ter	Sky	Ped	Rid	Car	Tru	Bus	Tra	Mot	Bic	Class	Cat
ENet [13]^{ICCV2015}	96.3	74.2	75.0	32.2	33.2	43.4	34.1	44.0	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3	80.4
SQNet [53]^{NeurIPS2016}	96.9	75.4	87.9	31.6	35.7	50.9	52.0	61.7	90.9	65.8	93.0	73.8	42.6	91.5	18.8	41.2	33.3	34.0	59.9	59.8	84.3
ESPNet [54]^{ECCV2016}	97.0	77.5	76.2	35.0	36.1	45.0	35.6	46.3	90.8	63.2	92.6	67.0	40.9	92.3	38.1	52.5	50.1	41.8	57.2	60.3	82.2
CGNet [55]^{TIP2016}	95.5	78.7	88.1	40.0	43.0	54.1	59.8	63.9	89.6	67.6	92.9	74.9	54.9	90.2	44.1	59.5	25.2	47.3	60.2	64.8	85.7
EDANet [57]^{MMAsia2019}	97.8	80.6	89.5	42.0	46.0	52.3	59.8	65.0	91.4	68.7	93.6	75.7	54.3	92.4	40.9	58.7	56.0	50.2	64.0	67.3	85.8
ERFNet [18]^{TITS2017}	97.2	80.0	89.5	41.6	45.3	56.4	60.5	64.6	91.4	68.7	94.2	76.1	56.4	92.4	45.7	60.6	27.0	48.7	61.8	66.3	85.2
ICNet [56]^{ECCV2018}	97.1	79.2	89.7	43.2	48.9	61.5	60.4	63.4	91.5	68.3	93.5	74.6	56.1	92.6	51.3	72.7	51.3	53.6	70.5	69.5	86.4
LEDNet [20]^{ICIP2019}	98.1	79.5	91.6	47.7	49.9	62.8	61.3	72.8	92.6	61.2	94.9	76.2	53.7	90.9	64.4	64.0	52.7	44.4	71.6	70.6	87.1
RTHP-SIS [61]^{TITS2020}	98.2	84.0	91.6	50.7	49.5	60.9	69.0	69.4	92.6	70.3	94.4	83.0	65.7	**94.9**	62.0	70.9	53.3	**62.5**	**71.8**	73.6	88.8
EPRNet [28]^{TITS2021}	98.2	83.5	91.7	49.9	52.8	**64.5**	69.4	**74.5**	**93.2**	70.0	**95.1**	**83.9**	64.1	**94.9**	56.8	72.4	60.4	57.2	71.5	73.9	**89.6**
LCNet$_{3\_7}$ (ours)	98.2	83.7	91.7	49.6	53.8	62.1	68.4	72.3	92.8	70.6	94.9	82.6	63.9	94.7	60.8	73.5	62.9	56.7	70.0	73.8	88.9
LCNet$_{3\_11}$ (ours)	**98.3**	**84.2**	**92.1**	**55.1**	**54.9**	63.5	**70.1**	73.2	93.0	**71.4**	**95.1**	83.4	**66.7**	**94.9**	**64.5**	**79.3**	**68.4**	59.4	71.6	**75.8**	89.5

TABLE X

PERFORMANCE COMPARISON OF LCNETS AGAINST EXISTING SEMANTIC SEGMENTATION NETWORKS ESTIMATED ON THE CAMVID TEST SET

Method	Source	Pretrain	Input Size	mIoU (%) ↑	Params. (M) ↓	Speed (fps) ↑	Devices	FLOPs (G)^{a}↓	CI ↑
ENet [13]	ICCV2015	No	360×480	51.3	**0.36**	79.8	3090	8.7	0.295
SegNet [17]	TPAMI2017	ImageNet	360×480	55.6	29.5	92.7	3090	652.5	0.165
ESPNet [54]	ECCV2016	No	360×480	55.6	**0.36**	**296.8**	3090	6.9	0.697
SwiftNet [58]	CVPR2019	No	720×960	63.3	11.8	-	3090	52.0	-
CGNet [55]	TIP2020	No	360×480	65.6	0.5	101.5	3090	14.0	0.656
EDANet [57]	MMAsia2019	No	360×480	66.4	0.68	161.6	3090	17.9	0.739
DABNet [19]	BMVC2019	No	360×480	66.4	0.76	183.9	3090	20.9	0.764
ICNet [56]	ECCV2018	ImageNet	720×960	67.1	7.8	52.9	3090	14.2	0.554
MiniNet-v2 [15]	TROBOT2020	No	720×960	69.0	0.5	143.0	3090	25.8	0.768
AGLNet [60]	ASC2019	No	360×480	69.4	1.12	90.0	1080Ti	-	0.708^{b}
MSCFNet [41]	TITS2021	No	360×480	69.3	1.15	-	-	-	-
LCNet$_{3\_7}$ (ours)	This work	No	360×480	70.2	0.51	187.0	3090	15.9	0.841
LCNet$_{3\_7}$ (ours)	This work	No	720×960	70.7	0.51	182.4	3090	15.9	**0.845**
LCNet$_{3\_11}$ (ours)	This work	No	360×480	70.3	0.73	140.0	3090	19.6	0.786
LCNet$_{3\_11}$ (ours)	This work	No	720×960	**71.8**	0.73	138.3	3090	19.6	0.813

[a] FLOPs normalized to images at the resolution of 1024×1024.
[b] Result calculated on 1080Ti GPU.

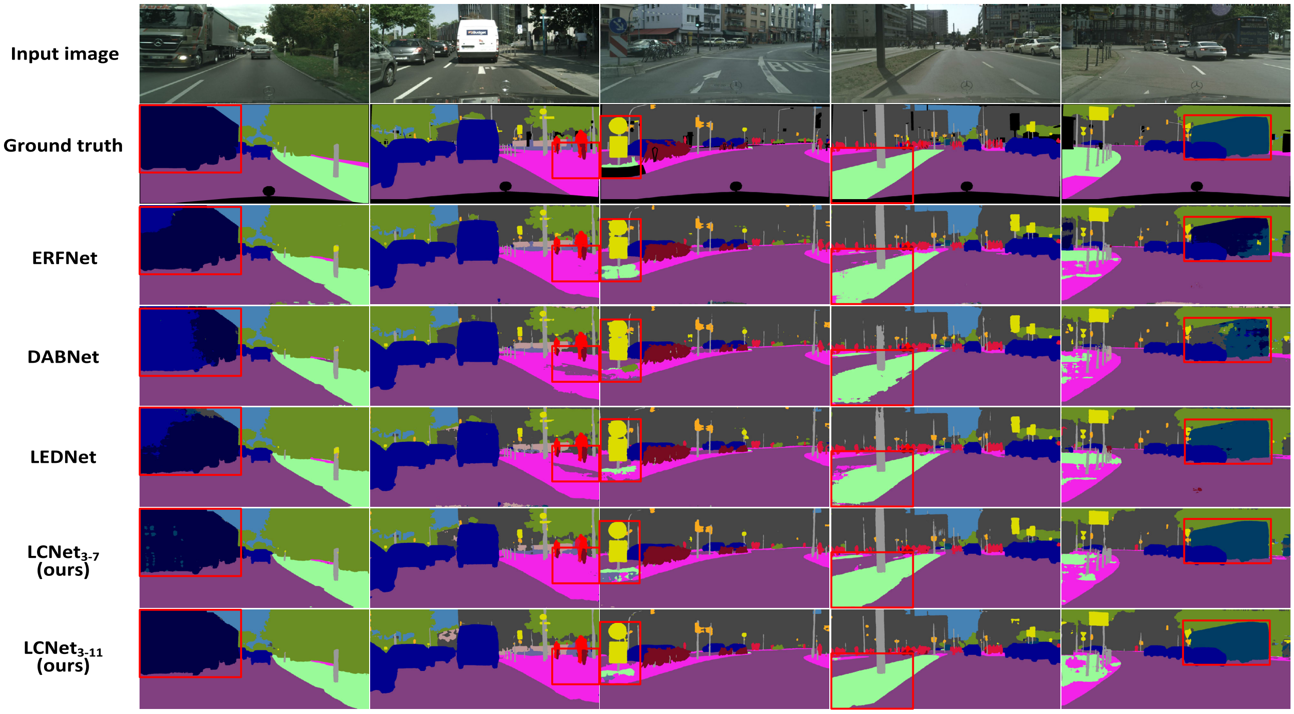


Fig. 9. Qualitative results of LCNet compared to state-of-the-art methods on the Cityscapes validation set.

for variable scales in street scenes and gains a strong learning ability. We also provide some examples of visualization results of semantic segmentation using different network models on the Cityscapes validation set as shown in Fig. 9.

2) Performance on CamVid: To demonstrate the effectiveness and generalization ability of LCNet in various benchmarks, we also evaluate the LCNet in the CamVid dataset [42]. The experimental results of LCNet and other

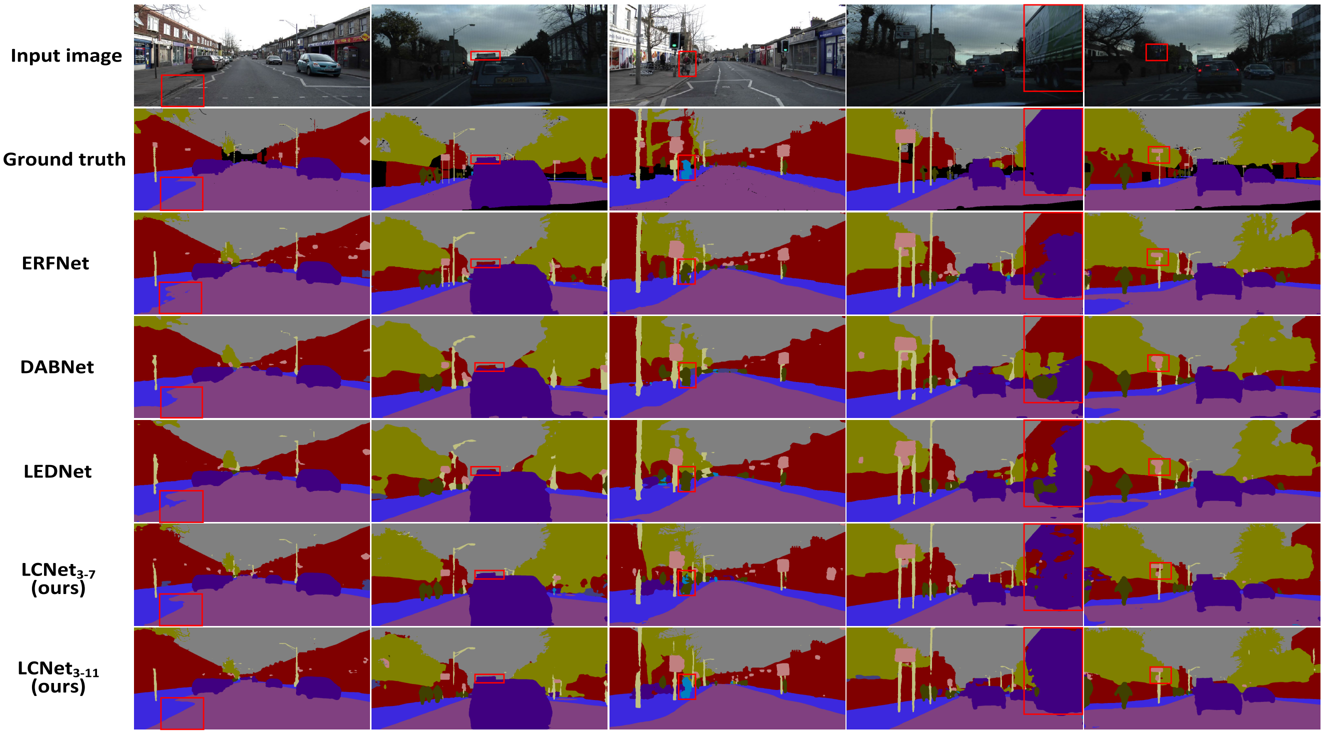


Fig. 10. Qualitative results of LCNet compared to state-of-the-art methods on the CamVid validation dataset.

state-of-the-art real-time semantic segmentation networks on the CamVid test set are summarized in Table X.

Our LCNet$_{3\_7}$ and LCNet$_{3\_11}$ achieve 70.2% and 70.3% mIoU without any pretraining for the input images in the resolution of 360 × 480. For higher-resolution images, LCNet$_{3\_7}$ and LCNet$_{3\_11}$ achieve testing accuracy of 70.7% mIoU and 71.8% mIoU, respectively. The accuracy performance of LCNet$_{3\_7}$ is 1.7% higher than MiniNet-v2 [15]. LCNet$_{3\_7}$ reaches a higher accuracy with <50% of parameters compared to AGLNet [60] and MSCFNet [41]. LCNet$_{3\_11}$ further enhances the segmentation accuracy performance. According to the experimental results in Table X, the accuracy performance of our LCNets on the CamVid test set is better than that of most existing models in the same input image resolution.

About the inference speed, our LCNet$_{3\_7}$ and LCNet$_{3\_11}$ can respectively process at 187 fps and 140 fps for the images in 360 × 480 resolution on a single RTX 3090 GPU. LCNet$_{3\_7}$ achieves sub-optimal inference speed while the ESPNet [54] gains the fastest speed. However, the accuracy of LCNet$_{3\_7}$ increases by 14.6% mIoU compared to ESPNet. Even though the greater model parameters of LCNet$_{3\_11}$ prolong the processing time, it is still faster than many existing networks like CGNet [55]. In brief, the experimental results show that both LCNet$_{3\_7}$ and LCNet$_{3\_11}$ achieve a decent trade-off between accuracy and inference speed with a low parameter amount. Additionally, we exhibit some examples of the visualization semantic segmentation results using different network models in Fig. 10. We find that the LCNet can better discriminate the boundaries between different objects on the CamVid dataset.

3) Performance on BDD100K: To further verify the stable performance of LCNet, we also provide a quantitative estimation of the challenging BDD100K dataset [43]. The experimental results compared with existing real-time approaches are summarized in Table XI. Compared to lightweight network models using a smaller backbone without pre-training, such as the ResNet18 [3] in DSMRSeg [68], DLA-34 [69] backbone in [43], and MobileNetV2 in [70] and [71], our proposed network LCNet$_{3\_11}$ still achieves a higher accuracy and fewer model parameters. The pre-trained backbone ResNet101 [3] with strong learning ability in DeepLab v3+ [72] indicates a progress of higher precision on the BDD100K dataset. Although there are a large number of images with multiple scenes in the challenging BDD100K dataset, the proposed LCNet demonstrates competitive learning capabilities and obtains a significantly high accuracy while utilizing fewer than 1 M parameters for semantic segmentation tasks in complex scenes.

TABLE XI
EXPERIMENTAL RESULTS ON THE BDD100K TEST SET

Method	Source	Backbone	mIoU (%)	Params. (M)
DeepLab v3+ [72]	ECCV2018	ResNet101	64.8	58.0
Yu [43]	CVPR2020	DLA-34 [69]	56.9	15.3
DSMRSeg [68]	LNCS2019	ResNet18	57.3	13.8
MLFNet [70]	T-IV2023	MobileNetV2	53.5	3.99
FSFFNet [71]	ICNIP2021	MobileNetV2	55.2	1.3
LCNet$_{3\_11}$(ours)	This work	From scratch	58.5	**0.74**

V. CONCLUSION

In this paper, we propose a real-time semantic segmentation network named LCNet, achieving a better balance between model size, segmentation accuracy, and inference speed. Our approach involves the meticulous design of key components: a compact convolution unit called three-branch context aggregation (TCA), a novel block incorporating a partial-channel transformation (PCT) strategy with TCA, and an ultra-lightweight dual-attention-guided decoder (DD). The simple and shallow encoder, primarily composed of PCT blocks, efficiently extracts and fuses multiscale contextual features. The DD integrates dual-level features using different attention masks. A comprehensive set of ablation experiments conducted on the Cityscapes validation set confirm the effectiveness of each component. Finally, we compare the overall performance of the LCNet with existing real-time semantic segmentation methods, summarize and briefly analyze the experimental results. LCNet strikes a great balance between computational resource consumption and pixel-level prediction capability, particularly well-suited for mobile application scenarios. In future work, our focus will be on enhancing the interpretability of CNN-based lightweight architectures with semantic information. Additionally, we plan to explore alternative solutions, such as vision transformer (ViT), to achieve an optimal trade-off between accuracy and efficiency on mobile GPUs.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[4] Y. Bai, Z. Zhang, Z. He, S. Xie, and B. Dong, "A dual-convolution-neutral-network enhanced strain estimation method for optical coherence elastography," *Opt. Lett.*, early access, pp. 1–4, Dec. 2023, doi: 10.1364/ol.507931.

[5] W. Zhang et al., "TopFormer: Token pyramid transformer for mobile semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12073–12083.

[6] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12077–12090.

[7] Q. Wan, Z. Huang, J. Lu, Y. Gang, and L. Zhang, "SeaFormer: Squeeze-enhanced axial transformer for mobile semantic segmentation," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–19.

[8] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[9] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2599–2608.

[10] R. Liu et al., "TransKD: Transformer knowledge distillation for efficient semantic segmentation," 2022, *arXiv:2202.13393*.

[11] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19173–19186, Oct. 2022.

[12] C.-B. Wang and J.-J. Ding, "EffSegmentNet: Efficient design for real-time semantic segmentation," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Oct. 2023, pp. 7423–7436.

[13] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.

[14] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[15] I. Alonso, L. Riazuelo, and A. C. Murillo, "MiniNet: An efficient semantic segmentation ConvNet for real-time robotic applications," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1340–1347, Aug. 2020.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[18] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[19] G. Li, I. Yun, J. Kim, and J. Kim, "DABNet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," in *Proc. 30th Brit. Mach. Vis. Conf.* Durham, U.K.: BMVA Press, 2020, pp. 1–12.

[20] Y. Wang et al., "LEDNet: A lightweight encoder–decoder network for real-time semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1860–1864.

[21] M. Shi et al., "LMFFNet: A well-balanced lightweight network for fast and accurate semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3205–3219, Jun. 2023.

[22] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[23] M. Ma, F. Zou, F. Xu, and J. Song, "RTSNet: Real-time semantic segmentation network for outdoor scenes," in *Proc. IEEE 9th Annu. Int. Conf. CYBER Technol. Autom., Control, Intell. Syst. (CYBER)*, Jul. 2019, pp. 659–664.

[24] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9182–9192.

[25] Y. Wang, Q. Zhou, J. Xiong, X. Wu, and X. Jin, "ESNet: An efficient symmetric network for real-time semantic segmentation," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis.* Cham, Switzerland: Springer, 2019, pp. 41–52.

[26] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[27] T. Liu, Z. He, Z. Lin, G.-Z. Cao, W. Su, and S. Xie, "An adaptive image segmentation network for surface defect detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 29, 2022, doi: 10.1109/TNNLS.2022.3230426.

[28] Q. Tang, F. Liu, J. Jiang, and Y. Zhang, "EPRNet: Efficient pyramid representation network for real-time street scene segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7008–7016, Jul. 2022.

[29] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[31] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1857–1866.

[32] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–13.

[33] Z. Lin et al., "Deep dual attention network for precise diagnosis of COVID-19 from chest CT images," *IEEE Trans. Artif. Intell.*, early access, Nov. 29, 2022, doi: 10.1109/TAI.2022.3225372.

[34] Z. Lin et al., "DBGANet: Dual-branch geometric attention network for accurate 3D tooth segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Nov. 8, 2023, doi: 10.1109/TCSVT.2023.3331589.

[35] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 234–250.

[36] Q. Huang, C. Wu, C. Xia, Y. Wang, and C.-C.-J. Kuo, "Semantic segmentation with reverse attention," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.

[37] J. Liu, Q. Zhou, Y. Qiang, B. Kang, X. Wu, and B. Zheng, "FDDWNet: A lightweight convolutional neural network for real-time semantic segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2373–2377.

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[39] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[40] X. Zhang, B. Du, Z. Wu, and T. Wan, "LAANet: Lightweight attention-guided asymmetric network for real-time semantic segmentation," *Neur. Comp. Appl.*, vol. 34, pp. 3573–3587, Jan. 2022.

[41] G. Gao, G. Xu, Y. Yu, J. Xie, J. Yang, and D. Yue, "MSCFNet: A lightweight network with multi-scale context fusion for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25489–25499, Dec. 2022.

[42] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, Jan. 2009.

[43] F. Yu et al., "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[45] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, Paris, France. Cham, Switzerland: Springer, Aug. 2010, pp. 177–186.

[46] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[47] D. Diakoulaki, G. Mavrotas, and L. Papayannakis, "Determining objective weights in multiple criteria problems: The critic method," *Comput. Oper. Res.*, vol. 22, no. 7, pp. 763–770, Aug. 1995.

[48] M. Fan et al., "Rethinking BiSeNet for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9711–9720.

[49] J. Peng et al., "PP-LiteSeg: A superior real-time semantic segmentation model," 2022, *arXiv:2204.02681*.

[50] S. Zhao, Y. Liu, Q. Jiao, Q. Zhang, and J. Han, "Mitigating modality discrepancies for RGB-T semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 6, 2023, doi: 10.1109/TNNLS.2022.3233089.

[51] Z. Huang, Y. Wei, X. Wang, W. Liu, T. S. Huang, and H. Shi, "AlignSeg: Feature-aligned segmentation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 550–557, Jan. 2022.

[52] X. Li et al., "Multi-level feature fusion network for nuclei segmentation in digital histopathological images," *Vis. Comput.*, vol. 39, no. 4, pp. 1307–1322, 2023.

[53] M. Treml et al., "Speeding up semantic segmentation for autonomous driving," in *Proc. Adv. Neur. Inf. Process. Syst.*, 2016, pp. 1–7.

[54] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 552–568.

[55] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 1169–1179, 2021.

[56] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 405–420.

[57] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Proc. ACM Multimedia Asia*, Dec. 2019, pp. 1–6.

[58] M. Oršic, I. Krešo, P. Bevandic, and S. Šegvic, "In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12599–12608.

[59] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9514–9523.

[60] Q. Zhou et al., "AGLNet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106682.

[61] G. Dong, Y. Yan, C. Shen, and H. Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3258–3274, Jun. 2021.

[62] Y. Li, X. Li, C. Xiao, H. Li, and W. Zhang, "EACNet: Enhanced asymmetric convolution for real-time semantic segmentation," *IEEE Signal Process. Lett.*, vol. 28, pp. 234–238, 2021.

[63] S. Li, Q. Yan, X. Zhou, D. Wang, C. Liu, and Q. Chen, "NDNet: Spacewise multiscale representation learning via neighbor decoupling for real-time driving scene parsing," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 21, 2022, doi: 10.1109/TNNLS.2022.3221745.

[64] S. Li, Q. Yan, C. Liu, M. Liu, and Q. Chen, "HoloSeg: An efficient holographic segmentation network for real-time scene parsing," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2395–2402.

[65] H. Pan, Y. Hong, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3448–3460, Mar. 2023.

[66] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "PIDNet: A real-time semantic segmentation network inspired by PID controllers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19529–19539.

[67] R. Gao, "Rethinking dilated convolution for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 4674–4683.

[68] M. Yang and Y. Shi, "DSMRSeg: Dual-stage feature pyramid and multi-range context aggregation for real-time semantic segmentation," in *Neural Information Processing* (Communications in Computer and Information Science). Cham, Switzerland: Springer, 2019, pp. 265–273.

[69] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.

[70] J. Fan, F. Wang, H. Chu, X. Hu, Y. Cheng, and B. Gao, "MLFNet: Multi-level fusion network for real-time semantic segmentation of autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 756–767, Jan. 2023.

[71] T. Singha, D.-S. Pham, A. Krishna, and T. Gedeon, "A lightweight multi-scale feature fusion network for real-time semantic segmentation," in *Proc. Int. Conf. Neural Inform.*, 2021, pp. 193–205.

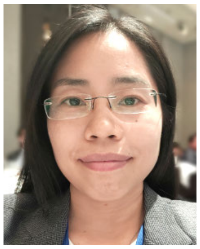
[72] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

Min Shi was born in Hubei, China. She received the M.S. degree in electronic engineering from the Wuhan University of Technology, Wuhan, China, in 2002, and the Ph.D. degree in signal processing and wireless communication from the South China University of Technology, Guangzhou, China, in 2005. She is currently an Associate Professor with Jinan University and the Director of the Technology Research Center for Satellite Navigation Chips and Applications, Guangdong University. Her research interests include machine learning, non-negative signal processing, image processing, and satellite navigation.

Shaowen Lin received the B.E. degree from Jinan University, Guangzhou, China, in 2020, where he is currently pursuing the M.S. degree. His current research interests include computer vision, machine learning, and deep learning.

Qingming Yi received the B.S. degree from Xiangtan University, China, in 1984, the M.S. degree from Jinan University, China, in 1990, and the D.Eng. degree from the South China University of Technology, China, in 2008. She is currently a Full Professor with Jinan University, Guangzhou, China. Her research interests include image processing, computer vision, multimedia security, and digital IC design.

Jian Weng (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and engineering from the South China University of Technology in 2000 and 2004, respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University in 2008. From 2008 to 2010, he held a post-doctoral research position with the School of Information Systems, Singapore Management University. He is currently a Professor, the Dean of the College of Information Science and Technology, and the Vice-Chancellor of Jinan University. His research interests include public key cryptography, cloud security, blockchain, and artificial intelligence. He served as the PC co-chair or a PC member for over 30 international conferences. He serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.



Aiwen Luo (Member, IEEE) received the D.Eng. degree from Hiroshima University, Japan, in March 2018. From April 2018 to August 2019, she was a Researcher with Hiroshima University. She is currently a Faculty Member of Jinan University, Guangzhou, China. Her research interests include computer vision, pattern recognition, robotics, and intelligent IC design.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security. He is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of "Highly Cited Researchers" in 2020 and 2021. He received the Third Price of Macao Natural Science Award as a Sole Winner in 2020 and a co-recipient in 2014. He has been the leading Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society since 2015. He also serves as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.