Parallel complement network for real-time semantic segmentation of road scenes

Qingxuan Lv, Xin Sun, *Member, IEEE* Changrui Chen, Junyu Dong, *Member, IEEE* and Huiyu Zhou

***Abstract*—Real-time semantic segmentation is in intense demand for the application of autonomous driving. Most of the semantic segmentation models tend to use large feature maps and complex structures to enhance the representation power for high accuracy. However, these inefficient designs increase the amount of computational costs, which hinders the model to be applied on autonomous driving. In this paper, we propose a lightweight real-time segmentation model, named Parallel Complement Network (PCNet), to address the challenging task with fewer parameters. A Parallel Complement layer is introduced to generate complementary features with a large receptive field. It provides the ability to overcome the problem of similar feature encoding among different classes, and further produces discriminative representations. With the inverted residual structure, we design a Parallel Complement block to construct the proposed PCNet. Extensive experiments are carried out on challenging road scene datasets, i.e., CityScapes and CamVid, to make comparison against several state-of-the-art real-time segmentation models. The results show that our model has promising performance. Specifically, PCNet* achieves 72.9% Mean IoU on CityScapes using only 1.5M parameters and reaches 79.1 FPS with 1024×2048 resolution images on GTX 2080Ti. Moreover, our proposed system achieves the best accuracy when being trained from scratch.**

***Index Terms*—Road scene understanding, real-time semantic segmentation, deep convolutional neural networks.**

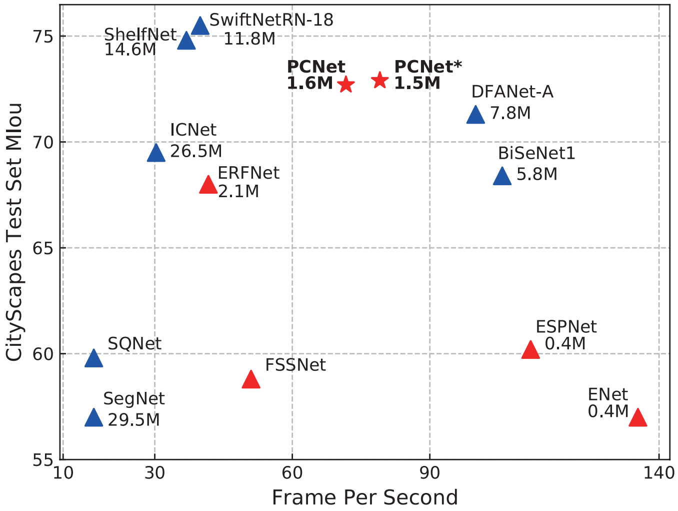


Fig. 1. FPS and Mean IoU comparison on the CityScapes test set. Red points represent the model is trained from scratch. Blue points indicate that the model utilizes the pretrained backbone. The number of the parameters is shown beside the points. PCNet* achieves comparable higher performance compared to the other pretrained methods. The FPS of most methods are evaluated on 1024×2048 resolution except SegNet, SQ, ENet with 640×360, FSSNet, ESPNet and ERFNet with 512×1024, BiseNet with 768×1536, DFANet with 1024×1024.

I. Introduction

SMANTIC segmentation is a basic task in computer vision, which is mainly responsible for labeling each pixel in the image. Thus, the segmentation model is able to understand the specific object details according to the data captured by a single or multiple cameras. It can be widely applied to many vision-based intelligent transportation systems such as autonomous driving, stereo reconstruction and video surveillance [1]–[5]. Therefore, the research of semantic segmentation has a great potential value for supporting above applications.

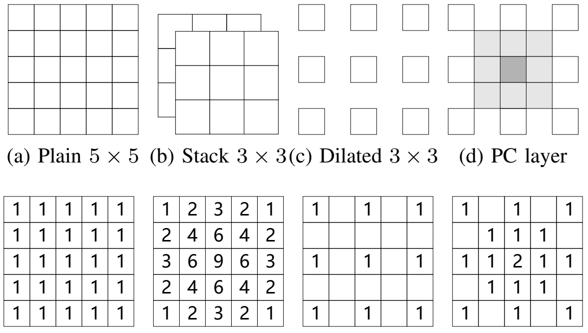
In the past few years, deep convolutional neural networks have made great progress in many areas [6]–[12]. In contrast to traditional methods, they will provide reliable features to accomplish specific task. Since Long *et al.* [13] firstly proposed to use fully convolutional neural network to solve the semantic segmentation task, many high performance networks [14], [15] have been developed in this area. Inspired by the transfer learning, most of the established system transfer general classification neural networks [16]–[18] as the backbone whilst exploiting prior information, which gives the model a strong capability for feature encoding. For example, many state-of-the-art models (DeeplabV2 [19], DenseASPP [20], PSPNet [21]) utilize ResNet [17] to extract high-level features. They also illustrate that multi-scale features are able to gradually improve the segmentation quality. However, limited by huge computational costs, the above methods do not tend to be applied in intelligent systems.

Recently, with rapidly increasing demand of real-time interaction, many researches turn to study how to perform fast semantic segmentation. Although early methods somehow reduce the computational costs, they result in significant performance degradation. For instance, SegNet [22] and SQNet [23], two representative models, are able to achieve a fast inference speed, but lose more than 10% segmentation accuracy compared with the other high-performance networks. Recent methods mainly address the real-time segmentation challenge from two directions. On the one hand, extremely efficient

This work was supported in part by the National Natural Science Foundation of China (No. U1706218, 61971388, L1824025), and Major Program of Natural Science Foundation of Shandong Province (No. ZR2018ZB0852). H. Zhou was supported by Royal Society-Newton Advanced Fellowship under Grant NA160342, and European Union's Horizon 2020 research and innovation program under the Marie-Sklodowska-Curie grant agreement No 720325. (corresponding author: X Sun)

Q Lv, X Sun, J Dong, and C Chen are with the Department of Computer Science and Technology, Ocean University of China, Qingdao, Shandong Province, 266100 China (e-mail: sunxin1984@ieee.org, dongjunyu@ouc.edu.cn).

H. Zhou is with the School of Informatics, University of Leicester, UK. (e-mail: hz143@leicester.ac.uk).



(a) Plain 5 × 5 (b) Stack 3 × 3 (c) Dilated 3 × 3 (d) PC layer

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

(e) Plain

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

(f) Stack

1		1		1
1		1		1
1		1		1

(g) Dilated

1		1		1
	1	1	1	
1	1	2	1	1
	1	1	1	
1		1		1

(h) PC layer

Fig. 2. Different convolutions and their sampling distribution with the same receptive field.

models are expected for the application of edge devices. However, the more lightweight the model is, the harder it is to achieve a breakthrough in performance [24]–[28]. Thus, the segmentation performance of the above models tend to be lower than other competitors. On the other hand, inspired by the success of the high performance models, some methods (i.e. BiSeNet [29], SwiftRN [30]) transfer ImageNet-pretrained backbones to exploit effective prior information. Compared with above lightweight models, they require large amount of computational costs. But the segmentation performance of them are improved considerably. Triangle points in Fig. 1 denote most recent real-time segmentation methods, which shows it is still a challenge to reach a satisfactory balance between computational costs and segmentation performance on large resolution images.

It is a common belief that enlarging receptive field helps the model make accurate prediction [19]–[21]. We first analyze the impact of different convolution operations with the same receptive field. Using a large convolution kernel [31] is a straightforward way to enlarge the receptive field, and this slightly improves the accuracy of segmentation. However, it is not an appropriate way for addressing the above challenge. Stacking [27], [28] and dilated convolution [25] are two alternative methods to acquire the same receptive field as large convolution kernels. Stacking convolution falls into the problem of heavily depending on local features, whereas dilated convolution hardly preserves local features. The problem is mainly due to the unreasonable sampling distribution. Fig. 2f demonstrates the result of a stacking method. Compared with the plain 5×5 convolution, the distribution changes to a hill shape which gradually forces the layer to over-sample the features within the same receptive field. Thus, similar feature representations of different classes confuse the classifier in prediction. As mentioned in [32], dilated convolution tends to ignore local features (see an example shown in Fig. 2g). Therefore, the gradient cannot be propagated in these zero weight areas, leading to the gridding effect on the high-level feature representations.

Based on the above analysis, a Parallel Complement layer (**PC layer**) is proposed to solve the above problems in this paper. The layer adopts two parallel convolutions to preserve local features and further enhance the representations through complementary information in a large scale. As shown in Fig. 2h, the PC layer produces a balanced sampling distribution which does not overuse local features. Therefore, the proposed PC layer has the ability to make consistent and accurate prediction.

Moreover, inspired by the structure of inverted residuals [33], we design an efficient Parallel Complement block (**PC block**). It has been proved in [33] that such architecture effectively captures the 'manifold of interest' embedded in a low-dimensional representation. Based on the PC block, we form the proposed Parallel Complement network (**PCNet**). The overall structure of our proposed model can be regarded as an encoder-decoder framework. The Encoder is comprised of a fast down-sampling module (FDM) and a feature extractor. To reduce computational complexity, the feature extractor is designed with a large down-sampling rate. The Decoder consists of a high-level fusion module (HLF) and a classifier. The HLF generates fused high-level features, and then the classifier combines the features with spatial information to make the final prediction.

The main contributions of this paper are as follows:

1) We design a Parallel Complement layer to enhance local features with proper complement information. The layer enlarges the receptive fields in a cheaper and more effective way than other competitors. More importantly, it is profitable for improving dense classification performance.
2) We design a Parallel Complement block by integrating the PC layer in the inverted residual to maintain the representation power. Moreover, we propose a PC-lite block to reduce parameters and computational costs by near a quarter.
3) We propose a Parallel Complement Network to address the real-time segmentation task. Compared with other state-of-the-art models, our system achieves higher Mean IoU (Intersection over Union) and a faster inference speed on high resolution images. It achieves a better balance among the parameters, computational costs and segmentation performance on two public scene segmentation benchmarks, i.e., CityScapes and CamVid.

II. Related work

A. Semantic Segmentation

FCN [13] is the first model to use fully convolutional network to address the semantic segmentation task. It replaces the fully connected layer which is usually used for classification with convolution operation to acquire the pixel-level prediction. Many subsequent methods are based on the fully convolution fashion.

Inspired by the undecimated wavelet transform [34], Deeplab [35] introduced atrous convolution into segmentation, which inserts zero holes within the convolution kernel. This operation is able to efficiently enlarge the receptive field. Then, in order to enhance the ability of multi-scale feature extraction, the Atrous Spatial Pyramid Pooling (ASPP) [19] is proposed, which uses different dilation rates within several parallel convolution operations. Hence, it is able to enrich scale information during the procedure of feature encoding. Furthermore, Yang *et al.* [20] propose to use dense connections within ASPP in order to make the final output features cover a large-scale range semantic information. In addition, for exploring the effectiveness of global contextual information [36], Zhao *et al.* [21] proposed to use global average pooling instead of atrous convolution to capture global image features.

On the other hand, feature aggregation within the same class helps to improve the segmentation quality. Object context pooling is introduced in OCNet [37] to find unified class feature representation. The features after aggregation are more discriminative. Similarly, DANet [38] performs feature aggregation from the view of attention. The model exploits both spatial and channel attention to adaptively integrate similar features from spatial and channel aspects. And in [39], the attention complementary module is introduced to selectively explore complementary feature from RGB and depth images by using channel attention mechanism. Meanwhile, for improving computational efficiency, CCNet [40] includes a criss-cross attention module to simplify the attention propagation. The above models are able to achieve promising segmentation performance. However, they incur huge computational burden which makes them hard to be applied in practical systems.

B. Real-time Semantic Segmentation

Recent years, real-time segmentation approaches [24]–[26], [29], [30], [41]–[46] have been proposed to reduce computational costs. As the pretrained backbones [17], [18] can provide a promising ability of feature encoding, some methods [29], [30], [41], [42] focus on innovating the decoder structure to address real-time segmentation tasks. ICNet [41] embeds PSPNet [21] in the cascade structure and feeds the network with a 1/4× input image. BiSeNet [29] introduces a two-branch network to preserve spatial information and obtain a sufficient receptive field. To reduce the overall computational costs, BiSeNet scales the input image to the size of 768×1536. DFANet [42] replicates the lightweight backbone and performs cross-level feature aggregation for capturing discriminating features. SwiftNet [30] incorporates the lightweight ResNet18 [17] backbone and fuses features at multiple resolutions to produce accurate segmentation results. Dong *et al.* [45] propose distinctive ASPP, based on MobileNetV2 [33], to exploit multi-scale information as much as possible. Benefiting from the pretrained backbone, the above methods are able to achieve promising performance. However, they still require large amount computational costs during inference.

As for pursuing high computational efficiency, some lightweight models [24]–[28], [43] pay more attention to reduce the number of the used parameters. ENet [24] develops an extremely lightweight architecture which only uses 0.4M parameters to construct the model. ESPNet [25] proposes an efficient spatial pyramid module for exploiting multi-scale features, and the model has 0.4M parameters in total. ERFNet [43] is formed with 2.1M parameters and the model adopts factorized convolution to improve computational efficiency. FSSNet [26] uses only 0.2M parameters to deliver a real-time segmentation task. Such lightweight networks are suitable for resource constrained devices, however they sometimes lead to the degradation of segmentation accuracy.

To bridging the gap between high-performance and lightweight models, knowledge distillation [47] provides a practical way for compact model to learn useful knowledge from a cumbersome model. Wang *et al.* [48] gives a comprehensive review about the knowledge distillation technique in many areas which is helpful for understand the technique details of knowledge distillation. In [49], Liu *et al.* proposed to use pairwise distillation and holistic distillation techniques to distill knowledge from teach segmentation model. In addition, He *et al.* [50] proposed to use a pretrained autoencoder to perform knowledge distillation so that the knowledge is able to be reinterpreted in a simplified form. Meanwhile, another affinity distillation module is used to provide long-range dependencies for student model. Therefore, the knowledge distillation method has great potential to be applied as a postprocessing technique to further optimize the model. In this work, we mainly focus on constructing a novel lightweight real-time segmentation model. Without any knowledge transferred from a cumbersome model, our best model PCNet* only has 1.5M parameters but produces accurate prediction results.

C. Effective Layer Design

Multi-scale information and receptive fields significantly influence the segmentation performance. To generate high quality image descriptors, SDC is proposed in [51] to enlarge the receptive fields so that the model is able to deal with the problem of image description. In terms of the segmentation task, many high performance models [19]–[21], [52] execute multi-scale feature extraction through elaborate modules. For example, ASPP [19] utilizes different dilated convolution schemes to exploit discriminative features. Pyramid pooling module [21] fuses sub-region features generated by four different levels of pooling operations. Dense ASPP [20] module introduces dense connections to make the features cover a larger scale range than ASPP. In addition, to improve the efficiency, QGNet [53] proposes a novel method to decomposes the target mask into a linear quadtree.

For most real-time segmentation models, they have to focus on the operational efficiency of each layer. ESPNet [25] applies spatial pyramids of dilated convolution to acquiring a large effective receptive field. Hierarchical feature fusion operation is used after the pyramid pooling to compensate heavy gridding artifact. ERFNet [43] uses factorized convolutions to improve the computational efficiency while keeping comparable accuracy. In our work, the proposed PC layer utilizes heterogeneous convolutions to generate a balanced sampling weight distribution in a large receptive field, and thus has the ability to produce discriminative representations.

D. Inverted Residuals

Because residual learning [17] alleviates the problem of gradient vanishing, a model with deep structure can be trained to acquire strong representational power. To enhance the representational power of a lightweight model, MobileNetV2 [33] introduces inverted residuals with a linear bottleneck structure to capture critical information embedded in low-dimensional representations. Incorporating with depth-wise separable convolution [18], inverted residuals lead to high computation efficiency while maintaining certain representational power. We form the proposed PC block with a similar structure. To further reduce computational costs, the PC-lite block is designed with a small expand ratio.

III. Methodology

In this section, we first introduce the proposed parallel complementary layer. Then for integrating the PC layer into a building block, we illustrate the structure of PC block and its lightweight version. Finally, we construct the Parallel Complement Network in the form of an encoder-decoder framework.

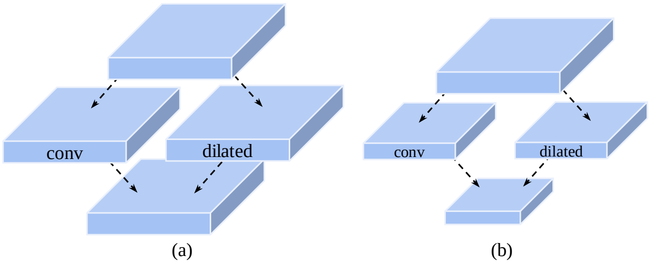


Fig. 3. The structure of parallel complementary layer and its downsampling structure.

A. Parallel Complement Layer

As mentioned in Section I, the stacking method causes the problem of heavily depending on local features and the dilated manner tends to lose essential local features. To overcome the above drawbacks, we propose parallel complementary layer (PC layer) to improve the ability of feature extraction. The PC layer consists of two parallel convolution operations, i.e., plain convolution and is dilated convolution. The two operations have the same settings except the dilation rate. The strides are set to 2 when the PC layer is applied to perform downsampling. Furthermore, we employ the depth-wise separable convolution to extract features instead of the normal convolution. Specifically, the depth-wise separable convolution reduces the parameters of operation by c times, where c denotes the channel number of the operation. Fig. 3 shows the structure of the PC layer. Fig. 3 (b) represents the structure of the PC layer when it is used to downsample features.

The main advantage of PC layer comes from the complementary relationship between features extracted by the two heterogeneous convolutions. In practice, the multi-scale features plays a significant role for improve the segmentation accuracy. For example, Deeplabv2 [19] introduces ASPP to extract multi-scale features, ESNet [28] proposes PFCU to capture multi-scale information and ESPnet [25] designs the ESP module to aggregate features from different scales. However, too many independent and parallel operations within a module also influence the efficiency. To reach a balance of performance and efficiency, we only use one dilated convolution within the PC layer to provide feature from a different scale. However, a problem is how the complementary relationship influences the segmentation performance of model. In [19], the ASPP chooses three relatively large dilation rates (6, 12, 18) to extract features while PFCU [28] employs (2,5,9) as the dilation rates of three parallel factorized convolutions. For the PC layer it will be beneficial for classifying large objects when the dilation rate is large. However, large dilation rate will inevitably confuse the model for small objects, because it will brings redundant features from other classes. On the other hand, the complementary relationship is too weak to provide necessary features from different scales when the dilation rate is small. Therefore, we further investigate the complementary relationship in the ablation study section. The results indicate that a proper dilation rate is profitable for improving the segmentation accuracy.

B. PC Block

As the complementary features from different scales are capable of refining the segmentation performance, we construct the Parallel Complementary block (PC block) with the PC layer as the basic feature extraction module. On the one hand, PC block is good at processing low-dimensional features which are vulnerable in preserving information. This is mainly because our PC block is developed based on the structure of inverted residual [33]. The channel expansion strategy is used in the inverted residual so as to guarantee the strong nonlinear transformation ability. In addition, the linear bottleneck prevents the information embedded in the low dimensional data from being destroyed by the nonlinear activation function. Therefore, PC block is able to exploit low-dimensional features. On the other hand, PC block is lightweight and efficient. A small expansion rate reduces the amount of parameters and computational costs while the performance is still relatively higher than other competitors [29], [41]. Then we describe the detailed structure of proposed PC block.

A PC block consists of three different layers, i.e., projection, transform and linear bottleneck layers. The projection layer employs one 1×1 convolution to project the compressed input data into the high-dimensional space. Although it increases the parameters and computational costs, the expanded features acquire excellent representation power, which alleviates information losing after nonlinear operation to some extent. The transform layer is mainly responsible for feature transformation. We apply the PC layer to form the transform layer. The proposed PC layer performs balanced sampling with a large receptive field. Thus, integrating the PC layer as the transform layer makes the whole block more sensitive for capturing rich scale information. After that, another pixel-wise sum operation is added to generate the transformed features. The detailed structure is shown in Fig. 5a. For improving computational efficiency, we reduce the expansion ratio and use concatenation to replace the pixel-wise sum operation, which can be seen in Fig. 5b. The linear bottleneck layer is

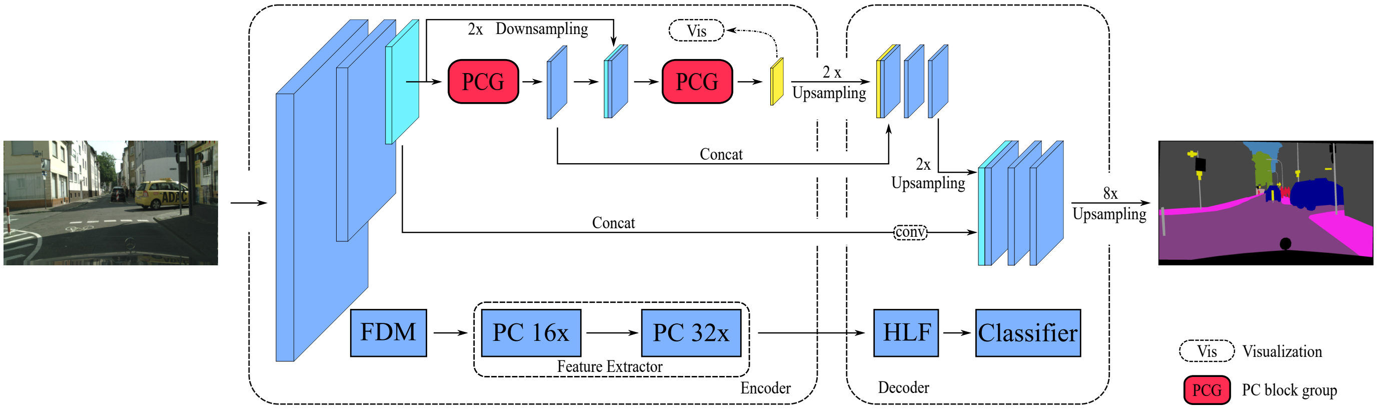
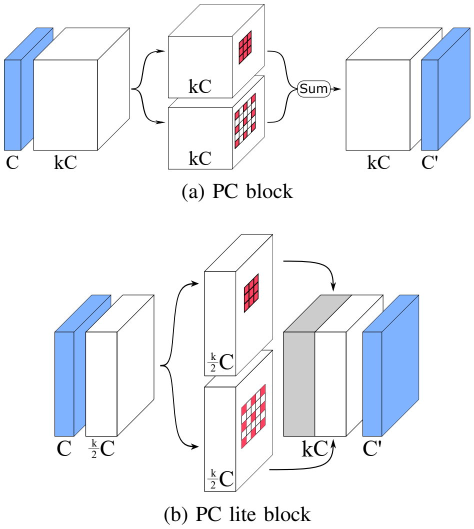


Fig. 4. Network Structure. PC Net adopts a simple encoder-decoder framework to form the whole network. Encoder contains a Fast Down-sampling Module (FDM) and a Feature Extractor. FDM consists of three plain convolution layers with stride 2. Feature Extractor is made up by two groups of PC blocks. Decoder contains a High-Level feature Fusion unit (HLF) and a classifier. HLF mainly focuses on combining different level features and the classifier is used to make the final prediction.



(a) PC block

(b) PC lite block

Fig. 5. Structure of PC and PC-lite block.

constructed without the ReLU activation operation, which is used to prevent non-linearity from destroying too much detail. We also use shortcut to perform residual learning when the channel of the input is equal to the output or the stride of the block is 1. It should be noted that the PC block is only used to illustrate the effectiveness of the PC layer. The PC-lite block is finally unitized to construct the proposed model.

Table I illustrates the implementation of the PC-lite block. For the input features of $C \times H \times W$, the projection layer performs channel expansion to generate $\frac{k}{2}C$ channel high-dimensional features where k represents the expansion ratio. Then the transform layer utilizes heterogeneous convolutions to produce complementary features, and we concatenate them to form kC channel features like the original inverted residual. Finally, the linear bottleneck layer is used to produce the output of $C^{'} \times H \times W$. Compared with the inverted residual, the PC-lite block reduces parameters and computational costs by nearly a quarter. Specifically, suppose that we have an input of $C\times H\times W$, the inverted residual block has total $12C^2+54C$ parameters, while PC-lite only has $9C^2+54C$ parameters.

TABLE I
THE IMPLEMENTATION OF THE PC-LITE BLOCK.

Input	Operators	Output
$C \times H \times W$	1×1 conv	$\frac{k}{2}C \times H \times W$
$\frac{k}{2}C \times H \times W$	3×3 dwise	$\frac{k}{2}C \times H \times W$
	3×3 rate d	$\frac{k}{2}C \times H \times W$
$kC \times H \times W$	1×1 conv	$C^{'} \times H \times W$

C. Network Structure

Our proposed Parallel Complementary network (PCNet) is constructed with an encoder-decoder structure. Different from most modern lightweight segmentation model designs [25], [28], [54], we employ a large downsampling rate to maintain high efficiency. Another advantage is large downsampling rate brings benefits for the network to increase the depth, which improves the representation power of the model. In addition, we expect to use the low-level features to refine the final segmentation prediction. The reason is that low-level feature is able to reserve spatial details which are beneficial to provide accurate location information [29], [41].

The structure of PCNet contains two main parts, named encoder and decoder, which can be seen in Fig. 4. In detail, the encoder contains one fast down-sampling module (FDM) and one feature extractor. The FDM is designed to perform fast downsampling so as to improve the computation efficiency of feature extractor. It's worth noting that we only use normal convolutions with stride in the FDM instead of the lightweight depth-wise separable convolution. Then, the feature extractor is constructed by two groups of PC block (PCG). The two groups contain $n1$ and $n2$ PC blocks respectively. In the best model, the input feature of the second group is formed by concatenating the output feature of the first group and the output feature of FDM. And we simply adopt bilinear interpolation to align two different size feature maps. Furthermore, we perform downsampling in the first PC block of each group. Therefore,

TABLE II
PERFORMANCE COMPARISON FOR THE CITYSCAPES VALIDATION SET. ALL MODELS ARE TRAINED WITH THE 512×1024 INPUT. FPS AND FLOPS ARE EVALUATED ON GTX 1080TI WITH 1024×2048 IMAGES. IR-$k \times k$ MEANS THE FEATURE EXTRACTOR IS MADE UP BY THE INVERTED RESIDUAL BLOCK WITH DEPTH-WISE CONVOLUTION OF A $k \times k$ KERNEL SIZE. dn DENOTES THAT THE DILATION RATE OF DILATED CONVOLUTION IS SET TO n. PC-(k, d) AND PC-LITE-(k, d) REPRESENT THAT THE MODEL IS CONSTRUCTED WITH PC AND PC-LITE BLOCKS WHERE THE PLAIN CONVOLUTION IS $k \times k$ DEPTH-WISE CONVOLUTION AND THE COMPLEMENTARY CONVOLUTION IS DILATED CONVOLUTION WITH RATE d.

Block	FLOPs	FPS	Params	MIoU
IR-3x3 (baseline)	8.49G	68.8	1.51M	67.2%
IR-5x5	8.85G	56.2	1.62M	69.8%
IR-7x7	9.40G	47.2	1.77M	69.4%
IR-3×3,d2	8.49G	68.8	1.51M	65.2%
IR-3×3,d3	8.49G	68.8	1.51M	63.5%
IR-3×3,stack	8.76G	54.1	1.58M	63.6%
IR-3×3,stack-d2	8.76G	54.1	1.58M	68.2%
IR-3×3,stack-d3	8.76G	54.1	1.58M	67.1%
PC-(3,2)	8.76G	52.3	1.58M	70.8%
PC-lite-(3,2)	7.1G	70.5	1.19M	68.3%
PC-lite-(3,2)-deep	9.17G	51.7	1.44M	70.5%

the final downsampling rate of the network is 32. The decoder is comprised of high-level feature fusion unit (HLF) and classifier. HLF is designed to fuse two different size high-level features which come from the two groups of PC block and the output of HLF is fed into classifier to generate the final segmentation prediction. As mentioned before, we introduce the output of FDM to classifier to provide location information and spatial details so as to refine the segmentation result. Finally, we directly apply bilinear interpolation to upsample the prediction. We use the cross entropy loss as the main objective loss function to optimize the whole network and it can be defined as:

$$CELoss = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -log(\frac{e^{p_i}}{\sum_j e^{p_j}}).$$

IV. EXPERIMENTS

To evaluate the performance of our PCNet, we perform experiments on challenging benchmark datasets including CityScapes [55] and CamVid [56]. In this section, we first describe the implementation details of our experiments. Then the ablation study of the proposed method is conducted and discussed. Finally, we carry out comparison experiments with several state-of-the-art real-time segmentation algorithms.

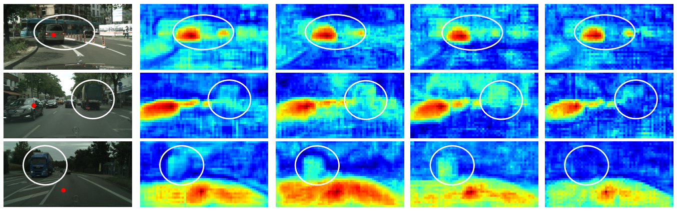

(a) input (b) plain5 (c) stack (d) dilated (e) PClayer

Fig. 6. Visualization results of high-level features. We randomly sample a point and compute its cosine similarity against the whole high-level feature map. Illustrations from left to right represent the original image, plain 5×5 convolution, stacking convolution, dilated convolution and PC layer respectively.

TABLE III
#PARAMETERS, #EFFECTIVE RECEPTIVE FIELDS(ERF) AND MEAN IOU COMPARISON WITH ESP MODULE [25], EDA MODULE [27] AND FCU [28]. FOLLOWING THE SETTING OF [25], HERE $m = n = 100$. THE DILATION RATE OF PC-LITE BLOCK IS SET TO 4.

Block	#Parameters	#FLOPs	#ERF	MIoU
Inverted Residual [33]	125400	529.2M	3×3	-
ShuffleNetV2 unit [58]	5450	-	3×3	-
ESP module [25]	20000	83.15M	33×33	60.3%
EDA module [27]	23200	-	5×5	67.3%
FCU(K=3) [28]	120000	496.44M	5×5	69.1%
PC-lite	95400	402.64M	9×9	**72.7%**

A. Implementation Details

Our proposed model is implemented based on PyTorch and we perform end-to-end training without any other postprocessing like conditional random fields. The Cross-Entropy loss is used as the objective function. We adopt mini-batch stochastic gradient descent (SGD) [57] to optimize the network with momentum 0.9 and weight decay $5e-4$ and the batch size of each iteration is set to 16. Following [19], we also utilize the 'poly' learning rate strategy where the initial rate is multiplied by $(1-\frac{iter}{max_{iter}})^{power}$ per iteration with power 0.9. The initial learning rate of the training is $1e-2$ and experiments are performed on 2× RTX 2080Ti. To avoid over-fitting, common data augmentations are used as preprocessing which includes random scaling of the origin image in the range of [0.5, 2] and image flipping image on horizontal with probability 0.5.

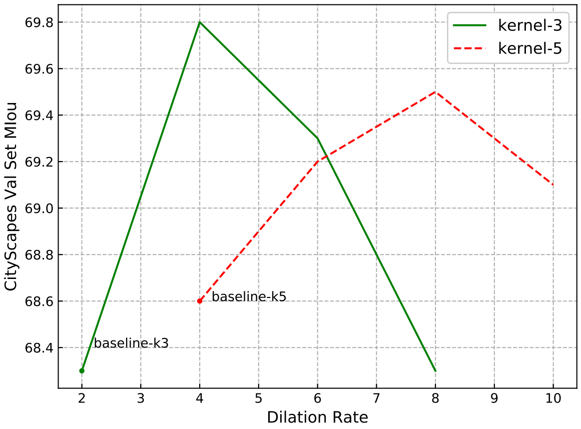


Fig. 7. Various settings of the PC-lite block. Red line denotes that the PC-lite block is constructed by standard 5×5 convolution and dilated convolution of rate d. Green line refers to standard 3×3 convolution and dilated convolution of rate d to form the PC-lite block.

B. Datasets and Evaluation Metrics

*1) **Datasets**:* We evaluate the proposed PCNet on two public road scene datasets: CityScapes [55] and CamVid [56]. Each dataset is introduced in the following content.

a) CityScapes: This dataset is mainly focused on semantic understanding of urban street scenes which contain a large number of paired data. In detail, CityScapes has 5000 finely annotated images and 20000 more coarsely annotated images collected from 50 cities. We only use the finely annotated images to train our model and the overall images are divided into 2975/500/1525 images for training/validation/testing. Each pixel of an image is labeled from 19 pre-defined classes which includes car, person, bicycle, etc. The resolution of each image is 1024×2048. For many ablation experiments, we train the proposed model by randomly cropping a 512×1024 patch from the image. For acquiring the best performance, we use the full resolution for training.

b) CamVid: CamVid is a road scene dataset which is captured from the perspective of a driving automobile. There are 701 labeled images in the dataset. Following SegNet [22], the dataset is split into 367 training images, 101 validation images and 233 images for testing. The dataset contains 11 different classes and the resolution is 960×720.

2) ***Metrics:*** The results is reported by using mean intersection-over-union (MIoU) which is widely used as the evaluation metric in semantic segmentation and scene understanding. It can be formulated as the following equation:

$$Mean \quad IoU = \frac{1}{n_{cls}} \sum_{i}^{n_{cls}} \frac{n_{ii}}{\sum n_{ij} + \sum nji - n_{ii}}$$

where n_{ij} means the number of the pixels of class i predicted to belong to class j. To evaluate the inference speed, we average the total time of 100 forward computations. The FLOPs and parameters are also listed in the result tables.

C. Ablation Study

In this section, we conduct extensive experiments on CityScapes with 512×1024 input resolution to perform ablation study. We first demonstrate the effectiveness of the PC layer and study the influence of the dilation rate of the complementary convolution. Then, different block structures are used to compare the segmentation performance so as to find the best basic block. Finally, to find the best network structure, we explore the impact of the depth of the feature extractor.

1) ***Effectiveness of PC block:*** To investigate the effectiveness of the PC block, we firstly compare it with the original inverted residual within the same network structure. The basic network structure is constructed with ($n_1 = 4$, $n_2 = 8$). The baseline block is an inverted residual with 3×3 depth-wise convolutions. Table II shows the baseline only achieves 67.2% Mean IoU on the CitySccapes validation set. Then, we enlarge the kernel size of depth-wise convolution. When the kernel size is 5×5, the Mean IoU is improved from 67.2% to 69.8%, and Mean IoU of the model reaches 69.4% with 7×7 kernel size. The improvement of segmentation accuracy implies that enlarging kernel size is a working strategy to achieve high performance. **IR-3×3-dn** represents the model uses dilated convolution to replace the normal 3×3 convolution. Results illustrate that zero weight sample points result in the degradation of performance. Meanwhile, directly applying the stacking method into the model also degrades the performance of model. However, the segmentation accuracy of stacking method is slightly improved when it coupled with dilated convolution. **PC-(k,d)** and **PC-lite-(k,d)** reveal that the block is formed with plain convolution with $k \times k$ kernel size and dilated convolution with the d dilation rate. PC-(3,2) reaches 70.8% Mean IoU on the CityScapes validation set, which is a significant improvement compared with IR-5×5, IR-3×3-d2 and the stack method. This promising result is achieved because the PC layer is capable of producing discriminative feature representations. Furthermore, the PC-lite block is designed to reduce computational complexity. With the same network structure, PC-lite-(3,2) saves many parameters while still acquiring high performance than the dilated or stack manner. After increasing the network depth to ($n_1 = 10$, $n_2 = 8$), PC-lite-(3,2)-deep achieves comparable Mean IoU with PC-(3,2). We also perform qualitative comparison on high-level features of different methods to illustrate the effectiveness of PC layer. Fig. 6 shows the visualization results of high-level feature similarity. We randomly sample a point and compute the cosine similarity between the point and the whole feature map. The results show that the PC layer can generate cleaner features than other competitors, which means the high-level features of a certain class in PC layer have few similarities with others.

2) ***Ablation for complementary relationship:*** We conduct ablation experiments on dilation rate to study how the complementary relationship influences the segmentation quality. We compare two PCNet with different basic blocks as the baseline models to investigate the correlation between the features. Fig. 7 summaries the experimental results. The green point represents the block of a 3×3 and dilated convolution to form the PC layer. The red point replaces the 3×3 convolution by 5×5 convolution. When we enlarge the dilation rate of dilated convolution, the green line shows that PC-lite block with a large rate's dilated convolution gradually worsens the segmentation performance. Thus, rate 4 dilated convolution is the most effective choice to exploit the complementary features for 3×3 convolution. Furthermore, the red line also reveals almost the same trend. As a result, the best dilation rate is able to provide the best complementary feature for plain convolution in the PC layer, which enhances the block representational power.

3) ***Quantitative comparison with other methods:*** In this part, we further conduct quantitative comparison with other lightweight blocks [25], [27], [28], [33]. This comparison takes parameters, effective receptive fields and Mean IoU into account and Table III summaries the results. Following [25], we use the same setting to compute the parameters and effective receptive fields. ESP module [25] has few parameters and large receptive fields by using tens of parallel dilated convolution. However, the low-dimensional representation makes it difficult for the module to improve the ability of generating discriminative features. In PC block, the features are expanded when performing transformation, which helps the block to exploit information compressed in the low-dimensional features. Factorized convolution unit (FCU) [28] stacks factorized convolutions to construct efficient structure. Recall the results of **Stack-d2** and **Stack-d3** in Table II, it illustrates that the performance still gets worse than plain 5x5 convolution when the stacking method is coupled with dilated convolution. It is worth noting that here we do not apply the factorized convolution in **Stack-d2** and **Stack-d3** because the factorized convolution and the normal convolution almost have the same affect except the factorized convolution is able to reduce

TABLE IV
COMPARISON OF PER-CLASS ACCURACY RESULTS ON CITYSCAPES TEST SET. THE BEST RESULT AMONG THESE METHODS IS MARKED BY BOLD STYLE.

Model	Pretrain	Roa	Sid	Bui	Wal	Fen	Pol	TLi	TSi	Veg	Ter	Sky	Per	Rid	Car	Tru	Bus	Tra	Mot	Bic	Params	MIoU
SegNet [22]	ImageNet	96.4	73.2	84.0	28.4	29.0	35.7	39.8	45.1	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	29.5M	57.0%
ICNet [41]	ImageNet	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**	26.5M	69.5%
ESPNet [25]	Scratch	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	0.4M	60.3%
PCNet*	Scratch	**98.3**	**84.4**	**91.4**	**48.4**	**52.6**	57.1	**63.8**	**69.7**	**92.3**	**70.0**	**94.6**	**80.6**	**61.5**	**94.5**	**61.2**	**73.9**	**63.2**	**57.3**	69.3	1.485M	**72.9%**

TABLE V
MODEL PERFORMANCE COMPARISON OF DIFFERENT PC-LITE BLOCK STRUCTURES. HERE, THE MEAN IOU IS EVALUATED ON THE CITYSCAPES VALIDATION SET. APS REFERS TO AVERAGE POOLING SIZE AND KS STANDS FOR KERNEL SIZE USED IN THE PLAIN CONVOLUTION. DR REPRESENTS DILATION RATE.

Block	KS	DR	APS	Params	MIoU
PC-lite-(3,2)	3	2	-	1.19M	68.3%
PC-lite-(3,2)-Add	3	2	-	0.85M	68.5%
PC-lite-(3,2)-deep	3	2	-	1.44M	70.5%
PC-lite-(3,2)-deep-Add	3	2	-	1.03M	69.4%
PC-lite	3	-	4	1.16M	64.5%
PC-lite	3	-	8	1.16M	64.0%

TABLE VI
ABLATION STUDY ON HLF.

Model	Input shape	MIoU
HLF PC-lite(3,2)-deep	512×1024	70.5%
HLF-LB PC-lite(3,2)-deep	512×1024	70.6%

TABLE VII
PERFORMANCE COMPARISON ON THE CITYSCAPES VALIDATION SET WITH DIFFERENT NETWORK STRUCTURES OF PCNET. THE INPUT RESOLUTION IS 512×1024. HERE, WE USE PC-LITE-(3,4) TO FORM THE PROPOSED NETWORK.

Methods	n_1	n_2	Params	MIoU
PCNet	4	8	1.382M	70.2%
PCNet	6	8	1.465M	70.4%
PCNet	8	8	1.548M	70.3%
PCNet	10	8	1.631M	70.6%
PCNet	10	6	1.384M	70.1%
PCNet	10	8	1.631M	70.6%
PCNet	16	12	2.241M	72.0%

parameters. We also report the FLOPs of each modules wchich is evaluated with a $100\times64\times64$ input in Table III. The results show that our proposed PC-lite block is able to achieve higher performance with relatively lower computational costs than others.

Meanwhile, the structure of PC block is similar to some other modules to some extent like EDA-ASPP [27] and PFCU in ESNet [28]. However, the meaning of parallel is different. For ASPP and PFCU, the parallel convolutions are all the dilated convolutions. That is complementary relationship existed in these parallel dilated convolutions enriches the scale information, which makes the whole module approximate a large and dense dilated convolution especially for extracting multi-scale features after encoder. As for PC block, the parallel operations are designed to provide complementary features with each other, which makes the block approximate a relatively large and sparse convolution. Additionally, because of the high efficiency, PC block is able to be applied into feature extractor to optimize the process of feature encoding. As a result, the proposed PC block is more profitable for real-time segmentation task.

4) ***Block Structure:*** In this subsection, our baseline network consists of the PC-lite-(3,2) block. We utilize the Global Average Pooling (GAP) operation within the PC-lite block to provide global contexts instead of complementary dilated convolution. As shown in Table V, the PC-lite block with Average Pooling Size (APS) 8 only achieves 64.0% Mean IoU which is lower than its competitor PC-lite-(3,2). There is a slight performance improvement from 64.0% to 64.5% in the case of APS $= 4$, which suggests the global context has little contribution to the features generated by plain convolution. In addition, we replace the concatenation with the point-wise sum operation within the PC-lite to form the 'PC-lite-(3,2)-Add' block. Results show that it not only reduces extra parameters but also improves the segmentation accuracy. However, the 'PC-lite-(3,2)-Add' block cannot achieve higher performance than the original PC-lite block with a deeper structure. In summary, the PC-lite block is more efficient than the others.

5) ***Network Structure:*** We further perform experiments to optimize the PCNet from the viewpoint of the feature extractor. Table VII shows the experimental results with different structures of PCNet. PCNet with $n_1 = 10$ acquires the best segmentation accuracy when we keep $n_2 = 8$ fixed. Meanwhile, PCNet achieves the best performance with $n_2 = 8$ when we fixed $n_1 = 10$. In addition, to prove the performance can be further improved with increasing the number of PC block in each PC block group, we add 6 and 4 more PC blocks in the two groups respectively. The result is presented in the last row of Table VII. To balance inference speed and performance, we choose $n1 = 10$ and $n2 = 8$ as the upper bound. Furthermore, inspired by the structure of inverted residual [33], we redesign HLF with channel expansion and linear bottleneck. Initially, the HLF is comprised of two depth-wise separable convolutions. We first concatenate the two different size features and then feed them to the following two convolutions. As for the redesigned HLF-LB, we first project the two different size features into a high-dimensional space and perform transformation on them respectively. Finally, we align the size of two features and project the feature back to a low-dimensional representation. Table VI shows the ablation results about HLF which can be seen that HLF-LB slightly

TABLE VIII
PERFORMANCE COMPARISON ON THE CITYSCAPES TEST SET. ”-” INDICATES THE CORRESPONDING RESULT IS NOT PROVIDED HERE. ”-VAL” MEANS THE PERFORMANCE IS EVALUATED ON THE VALIDATION SET. ”†” REPRESENTS THE FPS IS EVALUATED ON GTX 2080TI WITH 1024×2048 INPUT.

Model	InputSize	Pretrain	FLOPs	Params	FPS	GPU Type	MIoU
DeepLab [35]	512 × 1024	ImageNet	457.8G	-	0.25	-	63.1%
PSPNet [21]	713 × 713	ImageNet	412.2G	-	0.78	-	81.2%
SegNet [22]	640 × 360	ImageNet	286.03G	29.5M	16.7	-	57%
SQ [23]	640 × 360	ImageNet	270G	-	16.7	-	59.8%
ESPNetV2 [59]	512 × 1024	ImageNet	322M	725K	83	GTX Titan X	66.2%
BiSeNet1 [29]	768 × 1536	ImageNet	14.80G	5.8M	105.8	GTX Titan XP	68.4%
ICNet [41]	1024 × 2048	ImageNet	28.30G	26.5M	30.3	GTX Titan X	69.5%
GUNet [60]	512 × 1024	ImageNet	-	-	33.3	GTX Titan XP	70.4%
DFANet-A [42]	1024 × 1024	ImageNet	3.40G	7.8M	100	GTX Titan X	71.3%
KD-Resnet18 [49]	-	ImageNet	128.2B	15.24M	-	-	71.4%
KD-MobileNetV2 [50]	1025 × 2049	ImageNet	-	-	26.3	-	72.7%
TwoColumn [50]	512 × 1024	ImageNet	-	-	14.7	GTX 980	72.9%
Dong *et al.* [45]	448 × 896	ImageNet	49.5G	6.2M	51	GTX Titan X	73.6%
BiSeNet2 [29]	768 × 1536	ImageNet	55.3G	49M	65.5	GTX Titan XP	74.7%
ShelfNet-18 [44]	1024 × 2048	ImageNet	90.0G	14.6M	36.9	GTX 1080Ti	74.8%
SwiftNetRN-18 [30]	1024 × 2048	ImageNet	104G	11.8M	39.9	GTX 1080Ti	**75.5%**
ENet [24]	640 × 360	Scratch	3.83G	0.4M	135.4	GTX Titan X	57 %
FSSNet [26]	512 × 1024	Scratch	-	0.2M	51	GTX Titan XP	58.8%
ESPNet [25]	512 × 1024	Scratch	-	0.4M	112	GTX Titan X	60.3%
CGNet [54]	640 × 360	Scratch	6G	0.5M	35.2†	GTX 2080Ti	64.8%
ERFNet [43]	512 × 1024	Scratch	-	2.1M	41.7	GTX Titan X	68.0%
FRRN [61]	512 × 1024	Scratch	235G	-	2.1	GTX Titan X	71.8%
PCNet	1024 × 2048	Scratch	11.8G	1.631M	30.1	GTX Titan X	72.7%
PCNet	1024 × 2048	Scratch	11.8G	1.631M	71.8	GTX 2080Ti	72.7%
PCNet*	1024 × 2048	Scratch	11.5G	1.485M	79.1	GTX 2080Ti	**72.9%**
SwiftNetRN-18-val [30]	1024 × 2048	Scratch	104G	11.8M	39.9	GTX 1080Ti	70.4%
PCNet-val	1024 × 2048	Scratch	11.8G	1.631M	71.8	GTX 2080Ti	**72.7%**

improves the segmentation accuracy by 0.1%.

*6) **PCNet*:*** In this part, we introduce the structure about our best model PCNet*. Actually, PCNet* has the same structure with PCNet except that the PC block of the second group is replaced by large kernel PC blocks. In the second group of PCNet, we use 8 PC blocks to extract deep semantic information. While in the PCNet*, we replace the 8 blocks by 5 large kernel PC block. The kernel size of plain convolution of each large kernel convolution is set to be 5×5. As for the dilated convolution, the kernel size is still 3×3 while the dilation rate is set to be 3. The main purpose of this change is to improve the efficiency of PCNet. It is not only profitable for reducing the overall parameters and computational costs but also helpful to refine the final segmentation prediction.

D. Results

*a) **CityScapes**:* Due to the low computational costs of our proposed model, the full resolution inputs are used to train the best model. We carry out extensive comparison experiments against several state-of-the-art methods. Table IV summaries the per-class accuracy results on CityScapes test set of some real-time segmentation models. It can be concluded that our proposed PCNet* achieves better performances than others in most classes except ”Pole” and ”Bicycle”. On the other hand, we perform comprehensive comparison in inference speed, computational costs, parameters and segmentation performance with other methods and the results are presented in Table VIII. We **do not** apply TensorRT for acceleration. The speed evaluation of PCNet is conducted on one Nvidia GTX 2080Ti GPU and one Titan X GPU respectively. For a fair comparison, Table VIII lists the image resolution and the type of GPU. We do not add any testing augmentation strategies (i.e. Multi Scale test) in the evaluation process.

The first two rows of Table VIII show the results of two state-of-the-art high performance segmentation algorithms. Compared with these algorithms, PCNet reduces much more computational complexity and runs at a faster inference speed. For example, the proposed PC Net reduces the computational costs by almost 34× compared with PSPNet [21] and achieves a even high inference speed. Next, we also list twelve real-time semantic segmentation algorithms with the pretrained backbones. We can see that our method is still competitive. It is obvious that PCNet outperforms BiSeNet1 [29] by 4.3% Mean IoU and reduces much computational complexity. Compared with DFANet-A [42], PCNet can also achieve higher performance. SwiftNetRN-18 [30] finally achieves 75.5% Mean IoU with the pretrained backbone. It should be noticed that the PCNet achieves higher Mean IoU than SwiftNetRN-18 on the Cityscapes validation set when we train the model from scratch. The third part reports the performance of the lightweight segmentation models which are trained from scratch. Our proposed PCNet only occupies 1.32 GFLOPs when it is fed with a 640×360 input image, which is more efficient than ENet [24] and CGNet [54]. To make a fair comparison, we reimplement the CGNet [54] and evaluate the FPS on GTX 2080Ti with 1024×2048 resolution input. The results illustrate the competitiveness of PCNet. As for the segmentation performance, the proposed PCNet is able to deliver a descent performance compared with several state-of-the-art methods. In addition, we further refine the structure of PCNet to form our best model. PCNet* utilizes less parameters than PCNet while achieving better performance. Overall, the PCNet improves the segmentation performance in terms of

TABLE IX
PERFORMANCE COMPARISON ON THE CAMVID TEST SET. THE FPS OF OUR PROPOSED PCNET IS EVALUATED ON GTX TITAN X.

Model	Pretrain	Bui	Tre	Sky	Car	Sig	Roa	Ped	Fen	Pol	Sid	Bic	FPS	Params	MIoU
SegNet [22]	ImageNet	88.8	87.3	92.4	82.1	20.5	97.2	57.1	49.3	27.5	84.4	30.7	46	29.5M	55.6%
DFANet A [42]	ImageNet	-	-	-	-	-	-	-	-	-	-	-	120	7.8M	64.7%
BiSeNet1 [29]	ImageNet	82.2	74.4	91.9	80.8	42.8	93.3	53.8	49.7	25.4	77.3	50.0	-	5.8M	65.6%
ICNet [41]	ImageNet	-	-	-	-	-	-	-	-	-	-	-	-	26.5M	67.1%
ENet [24]	Scratch	74.7	77.8	95.1	82.4	51.0	95.1	67.2	51.7	35.4	86.7	34.1	-	0.4M	51.3%
FSSNet [26]	Scratch	84.6	86.0	94.3	84.6	57.9	95.2	80.9	43.4	53.6	92.9	67.4	179	0.2M	58.6%
SwiftNetRN-18-pyr [30]	Scratch	-	-	-	-	-	-	-	-	-	-	-	-	11.8M	65.7%
PCNet	Scratch	82.3	74.5	91.4	80.5	44.8	95.1	56.8	40.2	34.0	81.7	55.3	62.1	1.6M	67.0%

accuracy and speed with few parameters.

*b) **Discussion on Knowledge Distillation:*** As a novel technique to transfer knowledge from a cumbersome model to a compact model, knowledge distillation enables the lightweight model to improve the segmentation accuracy without any other computational costs increased. Wang *et. al.* [48] provide a comprehensive illustration about the technique. Such teacher-student training mechanism helps the compact model improve performance to some extent. For example, KD-MobileNetV2 [50] adopts the ResNet50 as the teacher network to distill knowledge to MobilenetV2 network. KD-Resnet18 [49] introduces a novel structured distillation method to transfer the knowledge to ResNet-18 model. Yang *et. al.* [62] propose a specialized ensemble method to bridge the gap existing in multiple heterogeneous data sources. The above methods give many insights to optimize the performance from the view of training progress. Comparing with above two methods [49], [50], our proposed PCNet also achieves a considerable performance without any other knowledge involved, which demonstrates the effectiveness of PCNet. As a result, knowledge distillation is still having great potential when applied to the training process of lightweight network.

*c) **CamVid:*** We also evaluate the proposed PCNet on the CamVid datatset. Note that in [22], [24], [26], they all downsample the images to 480×360 for classification. While following [29], we use the full resolution to train our model. The results are shown in Table IX and we use the per class accuracy, parameters and Mean IoU to compare our model against the other state-of-the-art models. Without being pretrained on ImageNet, PCNet is able to achieve comparable Mean IoU. Meanwhile, the proposed network is of less parameters than the other high performance models. Therefore, PCNet is capable for a real-time segmentation task.

E. Qualitative Results

Fig. 8 displays the results of ESPNet [25], BiSeNet1 [29] and PCNet. There is gridding artifact existing in the prediction of ESPNet. Even though the hierarchical feature fusion is designed to alleviate the gridding problem, it can not compensate for the side effect which is caused by a large dilation rate. The results of BiseNet1 illustrate that the segmentation results of big objects can be further improved. More importantly, our proposed PCNet achieves better results than the other real-time segmentation methods. On the one hand, the PC layer utilizes highly correlated heterogeneous convolutions to address the gridding artifact without an over large dilation rate. On the other hand, the PC layer is capable of generating complementary features which provide a suitable receptive field to increase the segmentation accuracy for big objects. Overall, our proposed PCNet achieves better segmentation performance while acquiring better efficiency than these state-of-the-art real-time segmentation models.

V. CONCLUSION

In this paper, we proposed a lightweight real-time segmentation network, named Parallel Complement network, which acquires a better balance between segmentation performance and inference speeds. The network employs a PC layer to generate complementary features with large receptive fields, which allows the model to learn discriminative representation among different classes. Moreover, we designed the Parallel Complement block integrating the PC layer in inverted residual. Extensive experiments on the CityScapes and CamVid datasets show that our model achieves best accuracy in the case of being trained from scratch. The results also demonstrate that the proposed PCNet has made a satisfactory balance between segmentation accuracy, inference speed and the model parameters. Real-time semantic segmentation is in intense demand for the application of autonomous driving, which has the ability to provide reliable scene understanding and predictions in a short time. This work contributes to understand the scene and interact with the environment in real-time, which is valuable to intelligent transportation applications.

ACKNOWLEDGMENT

We would like to express our sincere appreciation to the anonymous reviewers for their insightful comments, which have greatly aided us in improving the quality of the paper.

REFERENCES

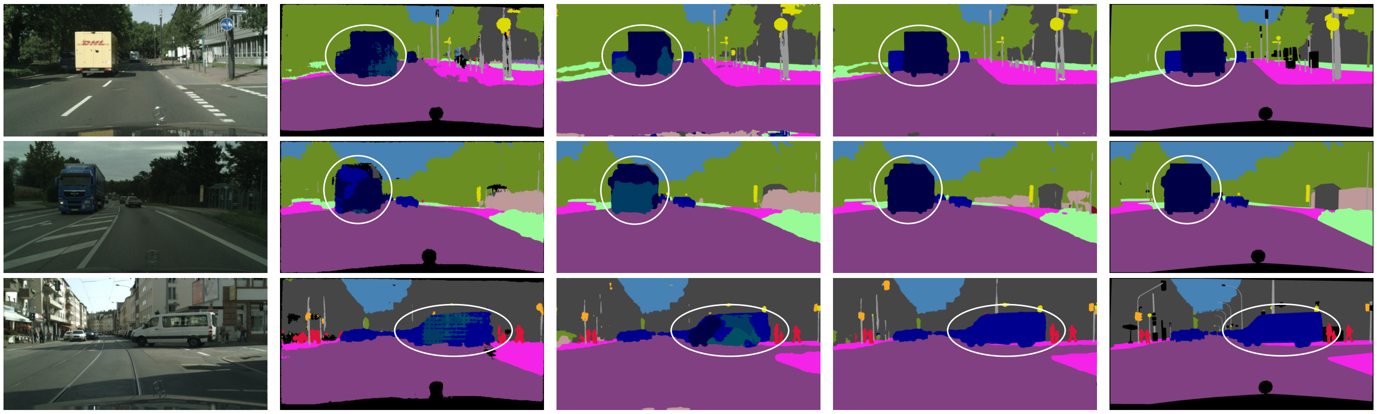
[1] B. Chen, C. Gong, and J. Yang, “Importance-aware semantic segmentation for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 137–148, 2019.

[2] L. Li, B. Qian, J. Lian, W. Zheng, and Y. Zhou, “Traffic scene segmentation based on rgb-d image and deep learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1664–1669, 2018.

[3] Y. Kang, K. Yamaguchi, T. Naito, and Y. Ninomiya, “Multiband image segmentation and object recognition for understanding road scenes,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1423–1433, 2011.

[4] V. Miclea and S. Nedevschi, “Real-time semantic segmentation-based stereo reconstruction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1514–1524, 2020.

[5] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang, “Pass: Panoramic annular semantic segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2019.



(a) Image (b) ESPV1 (c) BiSeNet1 (d) PCNet (e) GT

Fig. 8. Visualization results comparison with ESPNetV1 [25], BiSeNet1 [29] and PCNet. The first column shows the input images. The second and third columns present the results of ESPNet and BiSeNet1 respectively. The fourth column is the result of PCNet and the last column is the ground truth.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012.

[7] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[9] X. Sun, H. Xv, J. Dong, H. Zhou, C. Chen, and Q. Li, "Few-shot learning for domain-specific fine-grained image classification," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2020.

[10] C. Chen, X. Sun, Y. Hua, J. Dong, and H. Xv, "Learning deep relations to promote saliency detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[11] H. Zhao, X. Sun, J. Dong, C. Chen, and Z. Dong, "Highlight every step: Knowledge distillation via collaborative teaching," 2020.

[12] X. Sun, C. Chen, J. Dong, D. Liu, and G. Hu, "Exploring ubiquitous relations for boosting classification and localization," *Knowledge-Based Systems*, vol. 196, p. 105824, 2020.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[14] Z. Wu, C. Shen, and A. V. D. Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, 2016.

[15] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[19] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 40, no. 4, p. 834, 2018.

[20] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[23] M. Treml, J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schuberth, A. Mayr, M. Heusel, M. Hofmarcher, M. Widrich *et al.*, "Speeding up semantic segmentation for autonomous driving," in *ML-ITS, NIPS Workshop*, vol. 2, 2016, p. 7.

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

[25] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 552–568.

[26] X. Zhang, Z. Chen, Q. M. J. Wu, L. Cai, D. Lu, and X. Li, "Fast semantic segmentation for scene perception," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1183–1192, Feb 2019.

[27] 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 *Proceedings of the ACM Multimedia Asia*, ser. MMAsia '19. New York, NY, USA: Association for Computing Machinery, 2019.

[28] Y. Wang, Q. Zhou, and X. Wu, "Esnet: An efficient symmetric network for real-time semantic segmentation," 2019.

[29] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 325–341.

[30] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 599–12 608.

[31] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters — improve semantic segmentation by global convolutional network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1743–1751.

[32] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," *arXiv preprint arXiv:1702.08502*, 2017.

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[34] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," *Wavelets, Time-Frequency Methods and Phase Space*, vol. -1, p. 286, 01 1989.

[35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[36] R. Mottaghi, X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[37] Y. Yuan and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[38] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention

network for scene segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] X. Hu, K. Yang, L. Fei, and K. Wang, "Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1440–1444.

[40] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," 2018.

[41] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420.

[42] H. Li, P. Xiong, H. Fan, and J. Sun, "Dfanet: Deep feature aggregation for real-time semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9522–9531.

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

[44] J. Zhuang, J. Yang, L. Gu, and N. Dvornek, "Shelfnet for fast semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 847–856.

[45] G. Dong, Y. Yan, C. Shen, and H. Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2020.

[46] Z. Wu, C. Shen, and A. van den Hengel, "Real-time semantic image segmentation via spatial sparsity," 2017.

[47] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[48] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *arXiv preprint arXiv:2004.05937*, 2020.

[49] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2604–2613.

[50] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 578–587.

[51] R. Schuster, O. Wasenmüller, C. Unger, and D. Stricker, "Sdc – stacked dilated convolution: A unified descriptor network for dense matching tasks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2551–2560.

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

[53] K. Chitta, J. M. Álvarez, and M. Hebert, "Quadtree generating networks: Efficient hierarchical scene parsing with sparse convolutions," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 2009–2018.

[54] T. Wu, S. Tang, R. Zhang, and Y. Zhang, "Cgnet: A light-weight context guided network for semantic segmentation," 2018.

[55] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[56] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV (1)*, 2008, pp. 44–57.

[57] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proc. of COMPSTAT*, 01 2010.

[58] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[59] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9190–9200.

[60] D. Mazzini, "Guided upsampling network for real-time semantic segmentation," *arXiv preprint arXiv:1807.07466*, 2018.

[61] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3309–3318.

[62] K. Yang, X. Hu, Y. Fang, K. Wang, and R. Stiefelhagen, "Omnisupervised omnidirectional semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2020.

Qingxuan Lv was born in Shanxi, China, in 1996. He received his bachelor's degree in Computer Science and Technology from the Shanxi University of Finance and Economics in 2018. He is currently a candidate of a master's degree at the ocean group of VisionLab OUC. His research interests include computer vision and machine learning. Specifically, he is interested in real-time semantic segmentation and detection.

Xin Sun was born in Shandong, China, in 1984. He received his B.Sc, M.Sc. and Ph.D. from the College of Computer Science and Technology at Jilin University in 2007, 2010 and 2013 respectively. He did the Post-Doc research (2016-2017) in the department of computer science at the Ludwig Maximilians University of Munich. He is currently an associate professor at Ocean University of China. His current research interests include machine learning, computer vision and underwater image processing.

Changrui Chen was born in Shandong, China, in 1995. He received his bachelor's degree in Computer Science and Technology from the Ocean University of China (OUC) in 2017. He is currently a candidate of a master's degree at the ocean group of VisionLab OUC. His research interests are in computer vision and machine learning. In particular, he is interested in relation learning for detection and segmentation.

Junyu Dong received the B.Sc. and M.Sc. degrees in applied mathematics from the Department of Applied Mathematics, Ocean University of China, Qingdao, China, in 1993 and 1999, respectively, and the Ph.D. degree in image processing from the Department of Computer Science, Heriot-Watt University, Edinburgh, U.K., in November 2003. He is currently a Professor and the Head of the Department of Computer Science and Technology. His research interests include machine learning, big data, computer vision, and underwater image processing.

Huiyu Zhou received a Bachelor of Engineering degree in Radio Technology from Huazhong University of Science and Technology of China, and a Master of Science degree in Biomedical Engineering from University of Dundee of United Kingdom, respectively. He was awarded a Doctor of Philosophy degree in Computer Vision from Heriot-Watt University, Edinburgh, United Kingdom. Dr. Zhou currently is a Reader at Department of Informatics, University of Leicester, United Kingdom. His research work has been or is being supported by UK EPSRC, MRC, EU, Royal Society, Leverhulme Trust, Puffin Trust, Alzheimer's Research UK, Invest NI and industry.