SPFNet:Subspace Pyramid Fusion Network for Semantic Segmentation

Mohammed A. M. Elhassan[1], Chenhui Yang[1,*], Chenxi Huang[1] and Tewodros Legesse Munea[1]

[1]School of Informatics, Xiamen University,Xiamen, China,361005

***Abstract*—The encoder-decoder structure has significantly improved performance in many vision tasks by fusing low-level and high-level feature maps. However, this approach can hardly extract sufficient context information for pixel-wise segmentation. In addition, extracting similar low-level features at multiple scales could lead to redundant information. To tackle these issues, we propose Subspace Pyramid Fusion Network (SPFNet). Specifically, we combine pyramidal module and context aggregation module to exploit the impact of multi-scale/global context information. At first, we construct a Subspace Pyramid Fusion Module (SPFM) based on Reduced Pyramid Pooling (RPP). Then, we propose the Efficient Global Context Aggregation (EGCA) module to capture discriminative features by fusing multi-level global context features. Finally, we add decoder-based subpixel convolution to retrieve the high-resolution feature maps, which can help select category localization details. SPFM learns separate RPP for each feature subspace to capture multi-scale feature representations, which is more useful for semantic segmentation. EGCA adopts shuffle attention mechanism to enhance communication across different sub-features. Experimental results on two well-known semantic segmentation datasets, including Camvid and Cityscapes, show that our proposed method is competitive with other state-of-the-art methods. The source code is available at [1]**

***Index Terms*—Encoder-decoder architecture,Multi-scale feature fusion, Shuffle attention,Semantic segmentation,Convolution neural network..**

I. Introduction

SEMANTIC segmentation is one of the most challenging problems in computer vision. It aims to categorize each pixel of an image into a particular semantic class. Accurate semantic segmentation can be applied to wide-ranging applications in real-world scenarios such as autonomous driving [1], video surveillance, and robot sensing [2]. Driven by the recent advancement of deep convolutional neural networks (DCNNs), the pixel-wise semantic segmentation frameworks have witnessed unprecedented progress [3], [4], [5]. Existing CNNs based methods are usually constructed of stacked convolutional and downsampling layers. The shallower layers consist of spatial resolution feature maps, while the deeper stages capture the global context features.

Inspired by these observations, many new methods have followed fully convolutional network (FCN) [3] based structures to perform pixel-wise semantic segmentation tasks [6], [5], [7].

Email address: mohammedac29@stu.xmu.edu.cn(Mohammed A. M. Elhassan).

Corresponding author: Chenhui Yang*

[1]https://github.com/mohamedac29/SPFNet

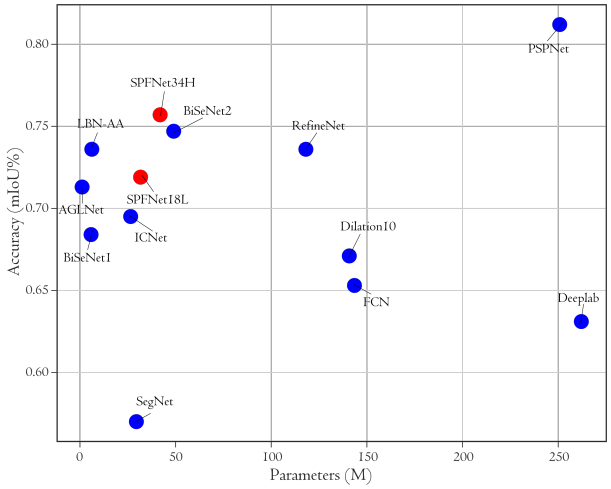
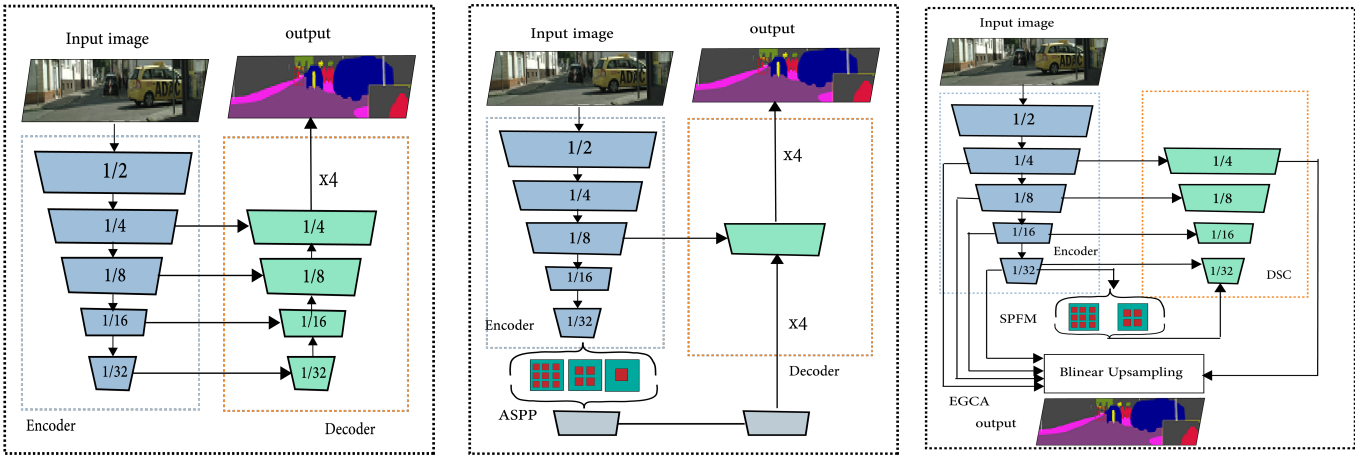


Fig. 1: Accuracy-Speed performance comparison on the Cityscapes test set.Our methods are presented in red dots while other methods are presented in blue dots. Our approaches achieve state-of-the-art speed-accuracy trade-off

The encoder-decoder models [7], [4] first capture the high-level semantic information by employing the original FCN and gradual upsampling to retrieve the original spatial information. Meanwhile, skip connections compensate for the fine information loss in the downsampling process. However, U-shape frameworks have achieved notable progress and received much attention in many computer vision applications [6], [7], but still less effective in extracting sufficient contextual information due to the following issues.

The first issue is dealing with objects that exist at multiple scales and have complex structures. A typical CNNs [3] have a weak ability to extract sufficient multi-scale information for accurate objects prediction. Another issue is that in a typical encoder-decoder framework (2.a), the encoder encodes the global context information in the deeper stages. The encoded information may progressively be diluted and cannot be fully recovered. Also, architectures based on UNet use skip-connections to aggregate features of the same scale by connecting encoder-decoder layers in parallel manners, but this simple implementation of skip-connections neglect global information, which results in pixel labels misclassification.

In recent years, methods with larger receptive fields have been proposed to solve these problems [8], [9] (2.b). Semantic segmentation performance can be improved significantly by



(a) Encoder-Decoder Structure (b) Encoder-Decoder with Atrous Conv (c) Encoder-Decoder with SPFM

Fig. 2: A comparison of an important semantic segmentation architectures. (a) encoder-decoder structure.(b) encoder-decoder structure with atrous convolutions.(c) our network, encoder-decoder structure with atrous convolutions and context aggregation module.

utilizing multi-scale context information. However, how to integrate such information effectively is still an open research question.

Multi-scale information is the most crucial problem that should be considered to improve the performance of semantic segmentation networks. Typically, a class of targets may exist at different scales in the image, and a well-designed model should be able to extract this property, which leads to better semantic segmentation accuracy. In this paper, we propose a Subspace Pyramid Fusion Module (SPFM) to improve the multi-scale feature learning in semantic segmentation. In particular, the SPFM generates global contextual information in each feature subspace based on parallel Reduced Pyramid Pooling submodule (RPP). RPP submodule constructed from three components: dilated convolution or atrous convolution, subpixel convolution, and average pooling. Two dilated convolutions were used for each reduced pyramid pooling level to capture object features with various receptive fields, followed by subpixel convolution instead of the original upsampling method to unify the feature resolution. To address the problems of resolution loss via downsampling in the encoding part, we develop the efficient global context aggregation module (EGCA) with two branches. Through the lower branch, i.e., EGCA integrates the shuffle attention block to facilitate cross-channel information communication for feature maps from different encoder levels and reduces channel redundancy. Finally, a Decoder-based Subpixel Convolution (DSC) is proposed to improve category pixel localization and recover object details. The SPFM module is embedded in the center between the encoder and the decoder to help select the suitable receptive field for objects with different scales by fusing the multi-scale context information. Based on these descriptions, we design the Subspace Pyramid Fusion Network (SPFNet), see Figure3. The proposed SPFNet is validated with two challenging semantic segmentation benchmarks, Camvid [10] and Cityscapes [11]. Figure. 1 shows the accuracy and the speed comparison of different methods on Cityscapes dataset.

Our main contributions are summarized as follows:

1) A novel Subspace Pyramid Fusion (SPFM) module is proposed to learn fused multi-scale and global context information for each feature subspace.
2) We introduce the Efficient Global Context Aggregation (EGCA) module, which utilizes channel shuffle operation to integrate the complementary global context
3) We propose decoder based on subpixel convolution to retrieve the high-resolution feature maps.
4) We proposed an architecture based on SPFM and EGCA modules that can be easily applied for road scene understanding. Furthermore, these modules can be plugged in any network
5) Extensive experimental results on Camvid and Cityscapes semantic segmentation benchmarks show that the proposed method has good generalization ability.

II. Related Work

A. Semantic Segmentation

Semantic segmentation has been widely studied in the past few years, and convolutional neural networks are at the center of this progress. Most available semantic segmentation architectures are motivated by the seminal fully convolutional network [3] or U-Net [5]. FCN was the first architecture to adopt standard classification CNNs for dense pixel prediction by replacing the fully connected layers with fully convolutional layers. In FCN, to retrieve the low-level representations of the input image, the output spatial map is upsampled in a single step using deconvolution [12], [13]. Further studies have utilized skip-connections which combine the semantic information from coarse layers with context information from shallow layers. On the other hand, U-Net comprises two parts. A contracting path is created using convolution layers with pooling to extract context information and expanding path

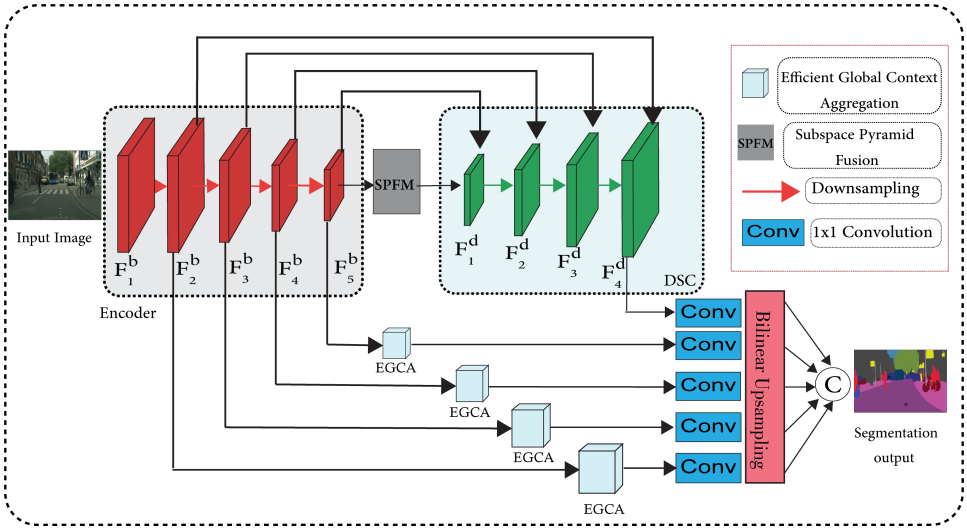


Fig. 3: An overview of the proposed Subspace Pyramid Fusion Network (SPFNet).

to recover the original resolution. Skip-connections are used to propagate the features from contracting to expanding path layers. Various variants of these networks have been developed to solve different pixel-wise segmentation tasks in a wide range of applications [14], [7]

B. Multi-scale and Context Aggregation

Multi-scale feature fusion is given great attention by the deep learning research community [9], [15], [9], [16] and has proven to be powerful in enhancing semantic segmentation performance. The straightforward way is to integrate low-level and high-level features to extract patterns of different granularity [7], [5]. DeepLab [9] applied atrous Spatial Pyramid Pooling (ASPP) in which a dilated convolution is used to increase the receptive field while maintaining the feature map resolution, leading to global context aggregation. PSPNet accomplished a similar goal by the spatial pyramid pooling [8]. In PAN [17] a feature pyramid attention is introduced to extract context information; in this approach, convolutions with large kernel sizes are used instead of using atrous convolution to construct the pyramid. Feature Pyramid Network [18] aggregates the multi-scale feature maps in a top-down fashion with progressive upsampling. Other network such as BiSeNet [19], ContextNet [20], GUN [21], and DSANet [22] utilized detail branch to capture low-level details in shallow layers.

C. Attention Mechanism

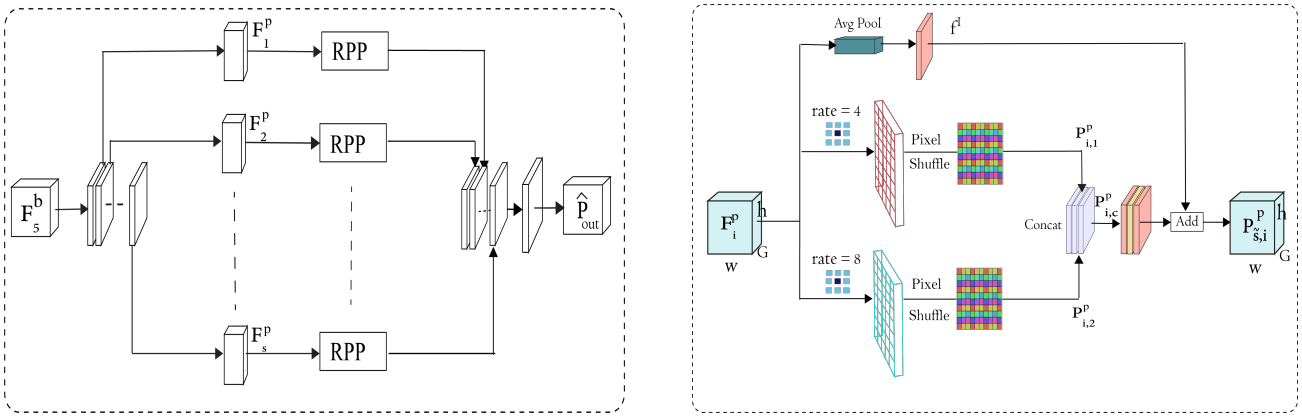
Self-attention mechanism has been extensively studied in the past few years, first in the domain of machine translation [23]. Recently, computer vision related tasks such as classification, detection, and segmentation have integrated different forms of attention mechanism [24], [25]. Some works utilize attention to update features with learned weights. For instance, SENet [26] highlights the channel maps through global pooling features. In [27] an attention mechanism is adopted to highlight the important region of interest at multiple spatial scales. CBAM [28] emphasizes the important regions, it expands the [26] to spatial dimension. SKNet [29] further proposed a dynamic kernel selection mechanism with small parameters overhead to improve the classification performance DANet [30] proposed adaptive method to integrate local and global features for semantic inter-dependencies modeling in spatial and channel dimensions. [31] proposed an attention mechanism that guides the feature update and helps to enrich relevant features adaptively. [32] combines channel attention and spatial attention consecutively. [33] adopted a 1-D convolution filter to avoid dimensionality reduction and increase cross-channel interaction, which reduced SENet model complexity. In this work, we constructed an attention mechanism based on channel shuffle operation to capture discriminative multi-level features. [34] introduced a lightweight attention module to improve the efficiency of compact CNNs architectures. The proposed module can be plugged into a classification network such as MobileNet to help learn multi-frequency and multi-scale feature.

III. Methodology

In this section, we first present an overview of the SPFNet, and then elaborate the mechanism of different modules that used to construct this network.

A. Overview

As illustrated in Figure 3, the SPFNet is constructed under a general encoder-decoder framework, where the encoder downsamples the input data to learn the rich multi-scale features and the decoder performs the reconstruction of the high-level features from the low-level features for pixel-level semantic prediction. The SPFNet first takes an input image and encodes its feature maps with a general multi-scale backbone, i.e., the ResNet34 [35]. The SPFNet utilizes $[F_1^b, F_2^b, F_3^b, F_4^b, F_5^b]$ to denote feature hierarchy from the backbone (encoder), b refers to backbone. Note that the feature maps have a stride of $[2, 4, 8, 16, 32]$. Unlike other methods such as [4], which replace stride with atrous convolution in



(a) Subspace pyramid fusion module (SPFM). (b) Reduced pyramid pooling module (RPP).

Fig. 4: Illustration of the (a) Reduced pyramid pooling module (RPP). (b) Subspace Pyramid Fusion (SPFM) module. SPFM extracts the multi-scale information in each sub-features using RPP module, then transmit the information to decoder.

the feature extraction part, we have set a stride of 1 for F_2^b to train SPFNet34H. Inspired by the discussion in the second problem, we further propose Subspace Pyramid Fusion Module (SPFM) (subsection. III-B), which generates multi-scale global context information. $[F_2^{egca}, F_3^{egca}, F_4^{egca}, F_5^{egca}]$ are the rich context information that extracted from the feature hierarchy in the Efficient Global Context Aggregation (EGCA) module (section. III-C). $[F_1^d, F_2^d, F_3^d, F_4^d]$ are the feature from Decoder-based Subpixel Convolution (DSC) (section. III-D), in which a gradual features upsampling and concatenation are performed to integrate the low-level and high-level feature representations.

B. Subspace Pyramid Fusion Module (SPFM)

It has been demonstrated in semantic segmentation literature [9], [36], [17] that increasing the receptive fields is beneficial to the semantic segmentation task. In a typical encoder-decoder architecture, the encoder extracts global context information from the input image, including the adjacent and class characteristics of the object. However, transmitting information to shallower layers will weaken the extraction of the context information due to the downsampling processes. In this subsection, motivated by this observation and inspired by [9] and [34], we proposed a novel subspace pyramid fusion module. For each subspace feature in SPFM, we use only one reduced pyramid pooling (RPP). Unlike [9] and [37], SPFM (as shown in Figure. 4) collects multi-scale contextual information differently. Instead of fusing the dilated convolution directly, SPFM first generates multi-scale features for each split using reduced pyramid pooling module. RPP constructed with only two parallel dilated convolutions, with dilation rates of (4,8). Next, a pixel shuffle is employed to increase the feature resolution. In addition, adaptive average pooling is also fused with the output from two parallel convolutions to capture complex context information. We then concatenate all the RPP to construct the SPFM. Compared with SPP and ASPP, our module can acquire more information using the subpixel convolution and the stacked RPP blocks. Unlike pyramid-based multi-scale learning such as SPP and ASPP, our module SPFM acquires more multi-scale information by learning multiple Reduced Pyramid Pooling blocks (RPP) for each feature map. Then, it combines these RPP blocks in one Subspace Pyramid Fusion Module. The main difference between the proposed SPFM and [34] is the way we aggregate information. [34] integrate attention across different stages in a pre-existing compact backbones to help the model learn the global information. In contrast, SPFM increases the receptive fields with dilated convolutions and adaptive average pooling to capture global context.

Given the feature map $F_5^b \in \mathbb{R}^{H\times W\times C}$ from the backbone, where H and W are the spatial dimensions of the feature maps and C is the number of channels. Our objective is to learn to capture multi-scale features efficiently. In Figure 4.a, we present the proposed SPFM module and its submodule RPP in Figure. 4.b. SPFM divides the input feature map F_5^b into s splits along the channel dimension, i.e., $F_5^b = \{F_i^p\}_{i=1}^s$, where F_i^p is input feature maps for a single Reduced Pyramid Pooling. Each subspace has $F_i^p \in \mathbb{R}^{H\times W\times G}$ feature maps, where $G = \frac{C}{s}$. Here are the details of SPFM and RPP modules.

Reduced Pyramid Pooling (RPP): Each RPP block takes one split from SPFM. First, the input features F_i^p enter into two identical pathways, each path consists of dilated convolution with kernel size of 3 followed by batch normalization(BN) and Parametric Rectified Linear Unit (PReLU) to generate multi-scale context information from these feature maps. Subsequently, we apply pixel shuffle to obtain $P_{i,1}^p$ as in Eq. 1 and $P_{i,2}^p$ Eq. 2. Then, we concatenate the output features $P_{i,c}^p$ Eq. 3 and refined it with three consecutive convolutions with kernel sizes (1×1), (3×3), and (1×1) respectively, to reduce aliasing effect. Finally, the output is added with feature map of adaptive average pooling and convolution Eq. 4 to produce the final reduced pyramid pooling $P_{\tilde{s},i}^p$ as shown in Eq 5.

$$P_{i,1}^p = \mathbb{PS}(\mathbb{D}_{conv}@4(F_i^p)) \tag{1}$$

$$P_{i,2}^p = \mathbb{PS}(\mathbb{D}_{conv}@8(F_i^p)) \tag{2}$$

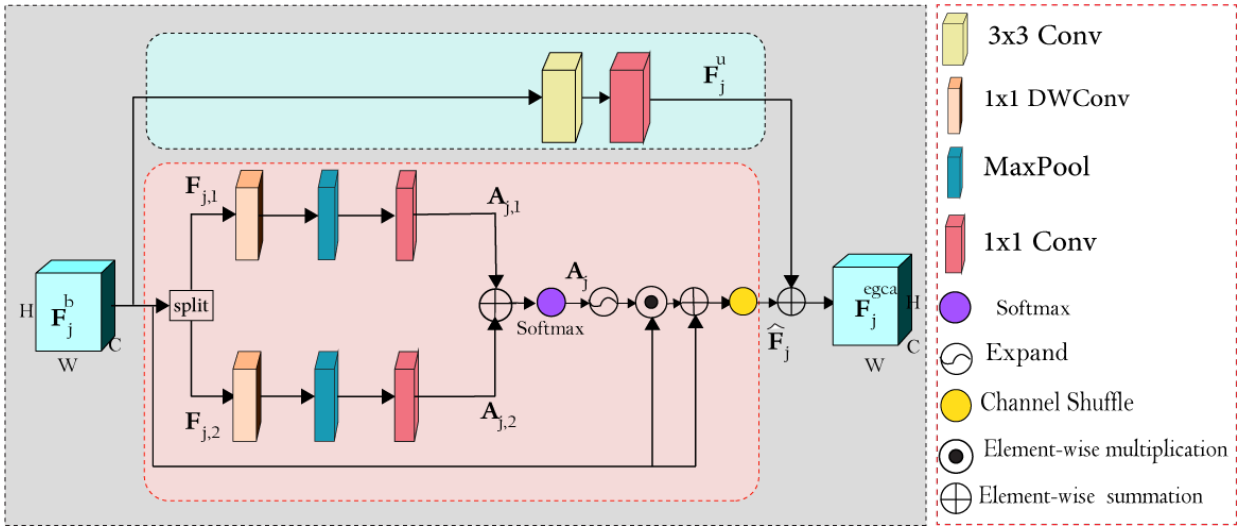


Fig. 5: An overview of the proposed Global Context Aggregation module. It utilize channel split/shuffle to process features in each group.

$$P^p_{i,c} = Concat([P^p_{i,}, P^p_{i,2}]) \tag{3}$$

$$A^p_i = f^1(APool(F^p_i)) \tag{4}$$

where Concat([.]) is the element-wise concatenation. The reduced pyramid fusion for each split $P^p_{\tilde{s},i}$ is computed by the following equation.

$$P^p_{\tilde{s},i} = A^p_i \oplus P^p_{i,c} \tag{5}$$

In Eq. 1, and Eq. 2, $\mathbb{D}_{conv}@$ refers to dilated convolution layer with kernel size of 3×3. In Eq4 APool(.), is the adaptive average pooling, f^1 represents convolution layer with kernel size of 1×1. In Eq. 1 and Eq. 2, $\mathbb{PS}(.)$ is the PixelShuffle with upsample scale factor of 2.

Subspace Pyramid Fusion Module (SPFM): The final output for SPFM module $\hat{P}_{out}$ is obtained by concatenating RPP of all splits (Eq. 6).

$$\hat{P}_{out} = f_1(Cancat([\hat{P}_{\tilde{s},1}, \hat{P}_{\tilde{s},2}, ..., \hat{P}_{\tilde{s},s}])) \tag{6}$$

C. Efficient Global Context Aggregation

Local and global context information are known to be useful for semantic segmentation and other computer vision tasks in even traditional machine learning era [38]. Deep learning methods have demonstrated astounding performance when integrating multi-scale features [16]. In this work, we propose an efficient global context aggregation module (Figure. 5) to make use of both local and global context. EGCA integrates channel attention through the shuffle unit operator to increase cross-feature interaction between all sub-features. In this setting, the input features from backbone are denoted as F^b_j, where j indicates the level in encoder (Figure. 3). Since the features in different levels have different semantic, we employ the EGCA before upsampling them to a common resolution using bilinear interpolation. Then features from all encoder scales and the last decoder-based subpixel convolution outputs are concatenated to from multi-scale feature map. Thus, fuse information from shallow layers and deeper layers.

Given the j^{th} feature $F^b_j \in \mathbb{R}^{H\times W\times C}$ from the backbone, where H and W are spatial dimensions and C is the channel dimension. The output F^{egca}_j of efficient global context aggregation module is computed using the following equation:

$$F^{egca}_j = \widehat{F}_j \oplus F^u_j \tag{7}$$

Where $\oplus$ represents the element-wise summation, $\widehat{F}_j$ is the output of shuffle attention part, and F^u_j represents the output feature of the upper part. At first, the input feature F^b_j is used to process information into upper branch and the shuffle attention branch (see Figure. 5). More details are given as follow:

Part I: Upper branch

in this branch F^b_j is passed through depthwise separable convolutions, leading to feature map of $F^u_j \in \mathbb{R}^{H\times W\times C}$ as shown in Eq. 8.

$$F^u_j = f^3_{DW}(f^1(F^b_j)) \tag{8}$$

where $f^3_{DW}(.)$ represents depthwise convolution with kernel 3×3.

Part II: Lower branch- shuffle attention

The second branch uses shuffle channel attention to generate sub-features that gradually capture multi-scale semantic responses. In this branch, the input F^b_j is divided into two groups $F_{j,1} \in \mathbb{R}^{H\times W\times C/2}$, and $F_{j,2} \in \mathbb{R}^{H\times W\times C/2}$ along the channel dimensions.

$$A_{j,1} = f^1_{PW}(\mathbb{MP}(f^1_{DW}(F_{j,1}))) \tag{9}$$

$$A_{j,2} = f^1_{PW}(\mathbb{MP}(f^1_{DW}(F_{j,2}))) \tag{10}$$

$A_{j,1} \in \mathbb{R}^{H\times W\times C}$,and $A_{j,1} \in \mathbb{R}^{H\times W\times C}$, have the same operations. Each of them starts with depth-wise convolution followed by Max Pooling and Point-wise. The two part are combined to form an attention map A_j for each sub-group to captures the long-range dependencies.

$$A_j = Softmax(A_{j,1} \oplus A_{j,2}) \tag{11}$$

The output $\widehat{F}_j$ is computed using Eq. 12.

$$\widehat{F}_j = Shuffle((A_j \odot F_j^b) \oplus F_j^b) \tag{12}$$

Where $\odot$ is the element-wise multiplication. $f_{DW}^1(.)$ represents a depth-with convolution with kernel size of 1×1, $\mathbb{MP}(.)$ is a Max Pooling and Point-wise $f_{PW}^1(.)$.

The output is normalized with softmax function in Eq. 11. After that, an element-wise multiplication and summation are applied in residual-like connections with input feature A_j. Furthermore, we adopt a channel shuffle operator similar to ShuffleNet [39] to enable cross-group information communication. Finally, the channel attention output Eq. 12 is added to the upper branch to form the Efficient Global Context Aggregation Module Eq. 7. The final output of the EGCA module is the same size as F_i^b, making EGCA quite simple and can be plugged in all the stages of the encoder, helping the module to learn multi-scale feature with different semantic. Thus, the EGCA module can be integrated easily into modern semantic segmentation architecture and other similar tasks. Table II shows the effect of the proposed efficient context aggregation module, in terms of floating-point operations per second (FLOPS), parameters, speed, and the mean intersection over union.

D. Decoder-based subpixel Convolution (DSC)

Many works have suggest a decoder structure in semantic segmentation methods. In some networks a naive decoder with direct bilinear upsampling is proposed [40], [8]. PAN proposed a module in which the low level feature is guided with global context. Different from mentioned works and in order to retrieve the high resolution feature maps efficiently, we exploit subpixel convolution [41] concept and technique. Multiple decoder-based subpixel convolution blocks are utilized in the DSC path to retrieve the spatial information with high level semantic feature generated by the SPFM module, and fuse the global context information gradually. The main component of the proposed DSC is shown in Figure 6 In details, we perform bilinear upsample to the high-level feature maps followed by 3×3 convolution, to reduce aliasing artifacts. 1×1 convolution is applied to the low level feature. Then we concatenate both low high feature and refine the output with 3×3 convolution. Finally, subpixel convolution is applied to generate a DSC with higher resolution. The following equation explains the relation between the different components of DSC.

$$F_{DSC} = \mathbb{PS}(f^3(Concat([up(f^3, f^1])))) \tag{13}$$

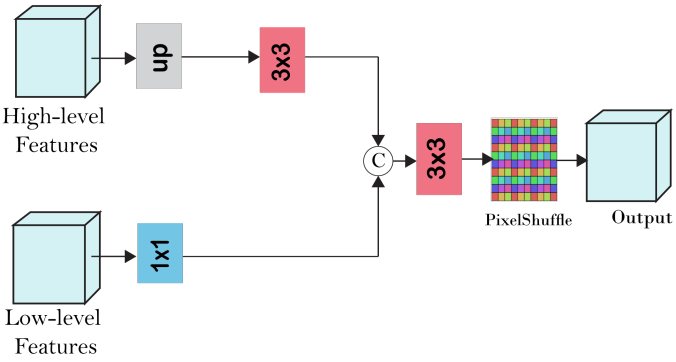


Fig. 6: Illustration of a decoder-based on subpixel convolution.

IV. Experiments

In this section, a comprehensive experiments have been carried out to evaluate the effectiveness of our proposed method, these experiments are conducted on Camvid [10] and Cityscapes [11] datasets. Experimental results demonstrate that SPFNet achieves state-of-the-art performance on Camvid dataset and comparable results to other methods on Cityscapes dataset. In the following we present the dataset and implementation details, then we demonstrate a series of ablation study on Camvid and Cityscapes datasets to explore the effect of each component in SPFNet. Finally, list the comparison of our method with the other state-of-the-arts networks on the two datasets.

A. Experiments Settings

1) Camvid: The Camvid dataset [10] consists of 376 training examples, 101 evaluating images, and 233 testing images. For a fair comparison with the other state-of-the-art models, we evaluated our model in 11 classes such as building, sky, tree, car, and road and the class 12th was marked as ignore class to hold the unlabelled data. The small size of the data and unbalance distribution of its labels makes the dataset more challenging.

2) Cityscapes: The Cityscapes dataset [11] is an urban scene understanding benchmark. It has 5000 high-resolution of 2048,1024pixels images with fine annotations captured from different cities. The dataset is divided into 2,975 images for training,500 images for validation, and the remaining 1525 images are for testing. We evaluate for 19 semantic segmentation classes.

3) Implementation Details: To implement our model, we use Adam optimizer [46] with weight decay 5×10^{-6}, power 0.9, and the starting learning rate lr_{init} for Camvid and Cityscapes datasets is set to 3×10^{-4}. We optimize the network by adopting the poly learning rate policy Eq. 14 similar to the previous works [9], [8]. All experiments are trained on PyTorch [47] on NVIDIA 3090 RTX GPU for 150 epochs and 500 epochs for Camvid and Cityscapes, respectively. The BatchNorm layers in SPFNet (except for the encoder) are replaced with InplaceBN-Sync [48] to reduce memory usage, while the speed, and FLOPS analysis are measured using Nvidia GTX1080 GPU.

TABLE I: THE PER-CLASS, CLASS, AND CATEGORY IOU EVALUATION ON THE CITYSCAPES TEST SET. LIST OF CLASSES FROM LEFT TO RIGHT: ROAD, SIDE WALK, BUILDING, WALL, FENCE, POLE, TRAFFIC LIGHT, TRAFFIC SIGN, VEGETATION, TERRAIN, SKY, PEDESTRIAN, RIDER, CAR,TRUCK,BUS,TRAIN, MOTORBIKE, AND BICYCLE."CLA->MIOU". "-" INDICATES THE CORRESPONDING RESULT IS NOT REPORTED BY THE METHODS

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motor	Bicyclist	mIoU
CRF-RNN[42]	96.3	73.9	88.2	47.6	41.3	35.2	49.5	59.7	90.6	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6	62.5
FCN[3]	97.4	78.4	89.2	34.9	44.2	47.4	60.1.5	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
DeepLabv2[9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	70.4
Dilation10[43]	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55.0	93.3	45.5	53.4	47.7	52.2	66.0	67.1
PSPNet[8]	**98.6**	**86.2**	**92.9**	50.8	**58.8**	64.0	**75.6**	**79.0**	**93.4**	**72.3**	**95.4**	**86.5**	**71.3**	**95.9**	**68.2**	**79.5**	**73.8**	**69.5**	**77.2**	**78.4**
AGLNet[44]	97.8	80.1	91.0	**51.3**	50.6	58.3	63.0	68.5	92.3	71.3	94.2	80.1	59.6	93.8	48.4	68.1	42.1	52.4	67.8	70.1
BiSeNetV2/BiSeNetV2_L[19]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	73.2
LBN-AA[45]	98.2	84.0	91.6	50.7	49.5	60.9	69.0	73.6	92.6	70.3	94.4	83.0	65.7	94.9	62.0	70.9	53.3	62.5	71.8-	73.6
Our(SPFNet)	98.5	85.2	92.6	48.4	55.8	**67.0**	74.5	77.8	**93.4**	70.7	95.2	85.7	68.8	95.2	53.3	73.0	59.7	67.2	75.4	75.7
Our(SPFNet)	98.4	84.9	92.2	47.6	54.1	65.9	73.6	76.2	93.0	66.4	94.8	85.3	68.5	94.6	47.6	53.9	28.8	65.7	75.0	71.9

All the experiments are implemented with the same data augmentation. Specifically, the input images are randomly scaled between 0.75 and 2.0, random horizontal flip, and random cropping of 512×1024 and 360×480 image patches for training Cityscapes and Camvid, respectively. We trained both datasets with weighted cross-entropy loss. The compare the experimental results of our model on the Cityscapes validation and testing sets with the current state-of-the-art segmentation models. Further analysis elaborated in subsequent sections.

$$lr = lr_{init} \times (1 - \frac{iter}{max\_iter})^{power} \tag{14}$$

Where max_iter is the maximum number of iterations.

B. Ablation Study

In this subsection, we run series of experiments to evaluate the impact of each component in the proposed SPFNet. First, an encoder-decoder based network that consists ResNet-based encoder and decoder-based sub_pixel convolution was employed as the baseline network for the ablation study.

1) Ablation for efficient global aggregation(EGCA)module: To test the influence of the efficient global context aggregation module in each stage, we set the integrate our proposed EGCA into a baseline of U-shape network ResNet34H (encoder) and DSC (decoder). Table. II illustrates the experimental results in terms of FLOPS, parameters, speed, and mean intersection over union. This process helps the model to learn multi-level cross-feature communication and reduce the semantic gap of feature from different level. The SPFM is not integrated into the network at this point. Using EGCA increases the computation by a small margin, but as we can see from the results it improves the segmentation accuracy. Table. II shows that integrating EGCA the second stage increases the FLOPS a little higher than other stages. Based on the configuration that uses EGCA in $stages2$ to $stage_5$, the mIoU improves over 2.84% than the baseline. EGCA dropped the speed by 5.7 FPS than the baseline.

2) Ablation on Subspace Pyramid Fusion Module (SPFM): Hyperparameter analysis on SPFM Module on the Camvid dataset: We investigate the impact of the hyperparameter s in the SPFM in terms of efficiency and accuracy on Camvid and Cityscapes datasets using Nvidia GTX1080 GPU. The parameter s represents the number of splits that used to learn

TABLE II: EVALUATION OF EGCA EFFECT AT DIFFERENT STAGES ON CITYSCAPES VALIDATION SET.FLOPS, SPEED ARE ESTIMATED FOR AN INPUT SIZE OF 512,1024.

Baseline	$Stage_2$	$Stage_3$	$Stage_4$	$Stage_5$	FLOPS (G)	Params(M)	Speed (FPS)	mIoU(%)
✓					268.6	37.7	19.3	72.16
✓	✓				280.3	37.8	16.2	72.8
✓	✓	✓			287.0	37.9	15.2	73.19
✓	✓	✓	✓		293.6	38.0	14.2	74.01
✓	✓	✓	✓	✓	300.2	38.7	13.6	75.0

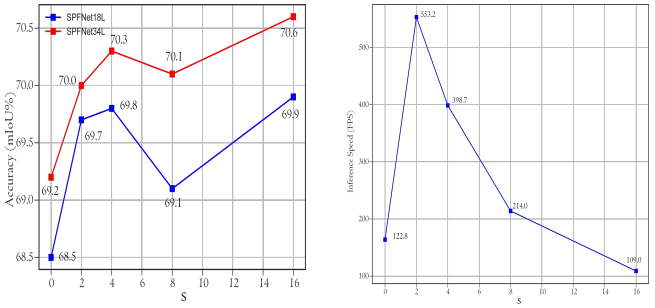
each RPP module in the SPFM. We compared four cases for using ResNet18 (SPFNet18L) as backbone and another four cases with ResNet34 (SPFNet23L). s = {2,4,8,16}. At first, to test the baseline we replace the SPFM with 1x1 Conv and report the results in Table. III for Camvid dataset. When s = 2, the SPFM fuses two reduced pyramid fusion modules to generate the multi-scale features. As shown in Table. III, using SPFM with different number of split show higher accuracy (mIoU) than the baseline, and the model with higher s shows segmentation accuracy improvement. This implies that using higher s increase the network ability to extract complex multi-scale features for both SPFNet34L and SPFNet18, but at the expense of the speed (FPS). AS shown in Table. III and Figure. 7.b. The performance of SPFM drops when using s=8 for both backbones as compared to the baseline. Using 2 split adds more parameters and FLOPS than using bigger s with faster inference. We also compared the SPFM with four splits in terms of FLOPS, number of parameters and speed. As illustrated in Table. III, a larger s adds less FLOPS and parameters, but it decreases the speed. Based on this analysis, we finally chose s=4 for SPFM considering both efficiency and the segmentation accuracy.

Hyperparameter analysis on SPFM Module on the Cityscapes dataset: We conducted similar analysis to examine the hyperparameter s of the SPFM on Cityscapes dataset in terms of the efficiency and accuracy. For this analysis we run the experiments for 150 epochs. From Table. IV and Figure. 8 Experimental results and evaluation shows that the hyperparameter s follows the same patterns as on Camvid dataset.

3) Ablation of SPFM and Other Multi-scale Modules: To further investigate the effect of different modules in our proposed network, we first took a modified ResNet34 by

TABLE III: THE EVALUATION AND ANALYSIS OF THE HYPERPARAMETER S ON THE SPFM MODULE. THE INPUT SIZE FOR THE BASELINE ON THE CAMVID TEST SET IS 360×840.THE INPUT SIZE FOR SPFM IS $512 \times 12 \times 15$.

Network	FLOPS (G)		Params (M)		Speed (FPS)		IoU (%)
	Overall	SPFM	Overall	SPFM	Overall	SPFM	
Baseline SPFNet-34L(ResNet34 as backbone)							
Baseline	22.8		37.7		122.8		69.2
SPFM (s=2)	22.8	3.0	37.7	6.7	122.8	553.2	70.0
SPFM (s=4)	22.8	1.6	37.7	3.4	122.8	398.7	70.3
SPFM (s=8)	22.8	0.896	37.7	1.9	122.8	214.0	70.1
SPFM (s=16)	22.8	0.544	37.7	1.0	122.8	109.0	70.6
Baseline SPFNet-18L(ResNet18+DSC Module)							
Baseline	16.4		27.6		163.9		68.5
SPFM (s=2)	16.4	3.0	27.6	6.7	163.9	553.2	69.7
SPFM (s=4)	16.4	1.6	27.6	3.4	163.9	398.7	69.8
SPFM (s=8)	16.4	0.896	27.6	1.9	163.9	214.0	69.1
SPFM (s=16)	16.4	0.544	27.6	1.0	163.9	109.0	69.9



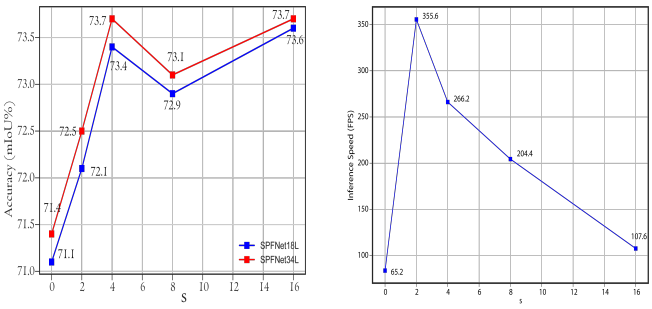
(a) Hyperparameter vs accuracy. (b) Hyperparameter vs FPS.

Fig. 7: (a) The accuracy obtained by the proposed SPFM with different values of the parameter s on the Camvid test set. (b) The inference speed with different s values.

not downsampling the second stage of the encoder. Then we integrated DSC module, and EGCA module as a baseline to test different multi-scale feature extractor modules. For this ablation setting the experiments run for 500 epochs. In detail, we conducted a number of experiments to compare the proposed SPFM with the multi-scale pyramid modules, i.e., ASPP, Vortex pooling, and DenseASPP, proposed in Deeplab [9], and Vortex Pooling [37], and DenseASPP [49], respectively. In these experiments, we used somewhat lighter backbone, the aforementioned ResNet18, and modified ResNet34. As

TABLE IV: THE EVALUATION AND ANALYSIS OF THE HYPERPARAMETER S ON THE SPFM MODULE.THE INPUT SIZE FOR THE BASELINE ON THE CITYSCAPES TEST SET IS 512×1024.INPUT FOR THE SPFM IS $512 \times 16 \times 32$.

Network	FLOPS (G)		Params (M)		peed (FPS)		mIoU (%)
	Overall	SPFM	Overall	SPFM	Overall	SPFM	
Baseline SPFNet-34L(ResNet34 as backbone)							
Baseline	68.1		37.7		65.2		71.4
SPFM (s=2)	68.1	8.6	37.7	6.7	65.2	355.6	72.5
SPFM (s=4)	68.1	4.6	37.7	3.4	65.2	266.2	73.7
SPFM (s=8)	68.1	2.5	37.7	1.9	65.2	204.4	73.1
SPFM (s=16)	68.1	1.5	37.7	1.0	65.2	107.6	73.7
Baseline SPFNet-18L(ResNet18+DSC Module)							
Baseline	48.7		27.6		83.7		71.1
SPFM (s=2)	48.7	8.6	27.6	6.7	83.7	355.6	72.1
SPFM (s=4)	48.7	4.6	27.6	3.4	83.7	266.2	73.4
SPFM (s=8)	48.7	2.56	27.6	1.9	83.7	204.4	72.9
SPFM (s=16)	48.7	1.5	27.6	1.0	83.7	109.0	73.6



(a) Hyperparameter vs accuracy. (b) Hyperparameter vs FPS.

Fig. 8: (a) The accuracy obtained by the proposed SPFM with different values of the parameter s on the Cityscapes validation set. (b) The inference speed with different s values.

TABLE V: PERFORMANCE COMPARISON OF SPFM WITH STATE-OF-THE-ART MULTI-SCALE EXTRACTOR METHODS SUCH AS ASPP, DENSEASPP, AND VORTEX POOLING ON THE CITYSCAPES VALIDATION. FLOPS, SPEED ARE ESTIMATED FOR AN INPUT SIZE OF 512,1024. UNDER THE SPFNET34H SETTING.

Method	FLOPS	Params	FPS	mIoU(%)
ResNet34+DSC baseline	268.6	37.7	19.3	72.16
ResNet34+DSC+EGCA	300.0	38.7	13.6	75.0
ResNet34+DSC+SPFM	286.6	40.8	16.3	75.8
ResNet34+DSC+EGCA+DenseASPP	380.7	77.9	8.7	75.21
ResNet34+DSC+EGCA+ASPP	304.8	41.0	12.3	75.64
ResNet34+DSC+EGCA+Vortex Pooling	322.0	49.4	11.1	75.72
ResNet34+DSC+EGCA+SPFM	317.0	41.8	12.4	78.04

shown in Table. V, all the multi-scale modules have improved the baseline. The baseline model with the proposed SPFM outperforms those with ASPP (75.64% mIoU), Vortex Pooling (75.72% mIoU), and DenseASPP (75.21% mIoU), showing the highest accuracy 78.04 mIoU with parameters and speed closer to ASPP. Besides, SPFM requires less parameters and FLOPS than DenseASPP and Vortex Pooling. This could be attributed to the capability of SPFM module to extract complex multi-scale semantic information more efficiently. SPFM learns multiple RPP at the same time, also it uses small dilation rate to obtain different-resolution feature maps, which introduces less noise.

C. Results on Cityscapes Dataset

As shown in Table VI, we compared the performance of the proposed model SPFNet against the other state-of-the-art models in terms of FLOPS, parameters, speed. We run the experiments with the training set and validation set. The input resolution is reduced from 2048 × 1024 into half to train our models. We evaluate the model segmentation accuracy on the test set, then submit the results to Cityscapes dataset online server [2] to get the results on the Cityscapes benchmark. Here, the network is compared against small networks such ICNet [50], BiSeNet [51], and larger networks PSPNet [8], RefineNet [52], and Deeplab [53]. The proposed SPFNet34H achieves 75.7% mean IoU, with 41.9M parameters and inference speed

[2]https://www.cityscapes-dataset.com/submit/

TABLE VI: COMPARISON BETWEEN THE PROPOSED METHOD SPFNET AND OTHER STATE-OF-THE-ARTS METHODS ON THE CITYSCAPES TEST DATASET. FLOPS, SPEED ARE ESTIMATED FOR AN INPUT SIZE OF 512,1024. "-" INDICATES THE CORRESPONDING RESULT IS NOT REPORTED BY THE METHODS.

Method	Backbone	Resolution	Parameters(M)	FLOPS (G)	Speed(FPS)	test set	mIoU
CRF-RNN[42]	VGG16	512×1024	-	-	1.4	✓	62.5
Deeplab[53]	VGG16	512×1024	262.1	457.8	0.25	✓	63.1
FCN[3]	VGG16	512×1024	134.5	136	2	✓	65.3
Dilation10[43]	VGG16	512×1024	140.8	-	-	✓	67.1
RefineNet[52]	ResNet101	512×1024	118.1	526	9.1	✓	73.6
PSPNet[8]	ResNet101	713×713	250	412.2	0.78	✓	78.4
SegNet[7]	BiseNet18	640×360	29.5	286	16.7	✓	57
TD4-Bise18[54]	BiseNet18	1024×2048	-			✓	74.9
BiSeNet1[55]	Xception39	768×1536	5.8	14.8	72.3	✓	68.4
BiSeNet2[55]	ResNet18	768×1536	49.0	54.0	45.7	✓	74.7
ICNet[50]	PSPNet50	1024×2048	26.5	28.3	30.3	✓	69.5
LBN-AA[45]	LBN-AA+MobileNetV2	488×896	6.2	49.5	51.0	✓	74.4
BiSeNetV2L[19]	No	512×1024			47.3	✓	75.3
SPFNet34H	ResNet34M	512×1024	41.9	317	12.7	✓	75.7
SPFNet18L	ResNet18	512×1024	31.7	61.0	46.5	✓	71.9

of 12.7 FPS, while SPFNet18L obtains 71.9% mIoU, with 31.7M parameters and inference speed of 46.5. Most of the larger models in TableVI incorporate very deep feature extraction architectures with a larger number of parameters, whereas the proposed model used ResNet34 and ResNet18, which have fewer parameters but still obtain comparable results. We do not use multi-scale testing or multi-crop evaluation, two techniques that been used by many practitioners to help improve the accuracy. We illustrate some visual examples of SPFNet on Cityscapes validation set in Figure. 9.

D. Results on Camvid Dataset

Table VIII presents the segmentation accuracy of our proposed method on the Camvid test set and its comparison with other state-of-the-arts methods. The model trained with an input resolution of 360x480 pixels with no use of external data we use the training samples and validation samples to train and validate our models while the testing samples are used to get the result to compare with other state-of-the-art models. Overall, our model has better accuracy. Our SPFNet34H with ResNet34 as a backbone achieves a significantly 75.1% mIoU on the test set and inference speed 33.0 FPS, which is trade-off between speed and accuracy. Our SPFNet43L uses downsampling at stage 2 achieves 71.4% mIoU with 79.3 FPS. Furthermore, we test our methods with two configurations of ResNet18. The SPFNet18H obtains 72.2%mIoU with inference speed of 39.7. SPFNet18L obtains 70.3%mIoU with inference speed of 109 FPS. These experiments validate the proposed architecture design. As demonstrated in TableVIII, training the model with extra data increases the performance, for instance. BiSeNetV2* [19] performance improves from 72.4% mIoU, when trained without extra data, to 76.7% with model first train on Cityscapes data, while BiSeNetV2-L* performance improves from 73.2% mIoU, when trained without extra data, to 78.5% with th same settings. We report the individual category results on Camvid test set in Table. VIII. Finally, we show some of the visual results of our method SPFNet on Camvid test set.

TABLE VII: THE ACCURACY, SPEED AND PARAMETERS COMPARISON OF THE PROPOSED METHOD AGAINST OTHER SEMANTIC SEGMENTATION METHODS ON THE CAMVID TEST SET. THE SPEED OF OUR MODELS ARE ESTIMATED FOR AN INPUT SIZE OF 360,480. * INDICATES THE MODEL PRE-TRAINED ON CITYSCAPES. "-" INDICATES THE CORRESPONDING RESULT IS NOT REPORTED BY THE METHODS.

Method	Year	Resolution	Params (M)	Speed (FPS)	mIoU
Deeplabv2[9]	2017	720 × 960	262.1	4.9	61.6
PSPNet[8]	2017	720 × 960	250	5.4	69.1
DenseDecoder[51]	2018	720 × 960	-	-	70.9
Dilation8[56]	2016	720 × 960	-	4.4	65.3
SegNet[7]	2015	360 × 480	29.5	-	55.6
ENet[57]	2016	360 × 480	0.37	-	51.3
DFANet-A[58]	2019	720 × 960	7.8	120	64.7
DFANet-B[58]	2019	720 × 960	4.8	160	59.3
BiSeNet1[55]	2018	720 × 960	5.8	175	65.7
BiSeNet2[55]	2018	720 × 960	49.0	116.3	68.7
ICNet[50]	2018	720 × 960	26.5	27.8	67.1
DABNet[59]	2019	360 × 480	0.76		66.4
CAS[60]	2020	720 × 960	-	169	71.2
GAS[61]	2020	720 × 960	-	153.1	72.8
AGLNet[44]	2020	360 × 480	1.12	90.1	69.4
CGNet[62]	2020	360 × 480	0.5	-	65.6
NDNet45-FCN8-LF[63]	2020	360 × 480	1.1	-	57.5
LBN-AA[45]	2020	720 × 960	6.2	39.3	68.0
BiSeNetV2/BiSeNetV2L[19]	2021	720 × 960	-	124.5/32.7	72.4/73.2
BiSeNetV2*/BiSeNetV2L*[19]	2021	720 × 960	-	124.5/32.7	76.7/78.5
SPFNet34H(ours)		360 × 480	41.8	33	75.1
SPFNet34L(ours)		360 × 480	41.8	79.3	71.4
SPFNet18H(ours)		360 × 480	31.7	39.7	72.2
SPFNet18L(ours)		360 × 480	31.7	109	70.3

V. CONCLUSION

This paper proposed subspace pyramid fusion module SPFM, which learns multi-scale context information by dividing the input feature maps into multiple subspaces. Furthermore, we have introduced the efficient global context aggregation module EGCA, which uses the channel-shuffle operator to enhance the information communication across different subfeatures. We further propose a semantic segmentation structure based on our SPFM and EGCA modules, called SPFNet, to address the multi-scale feature fusion. The ablation studies in the Cityscapes dataset show the effectiveness of the proposed SPFM and EGCA. SPFNet34H, SPFNet34L, SPFNet18H, and SPFNet18L achieve 75.1% mIoU, 71.4% mIoU, 72.2% mIoU, and 70.3% mIoU , respectively on Camvid dataset. Our

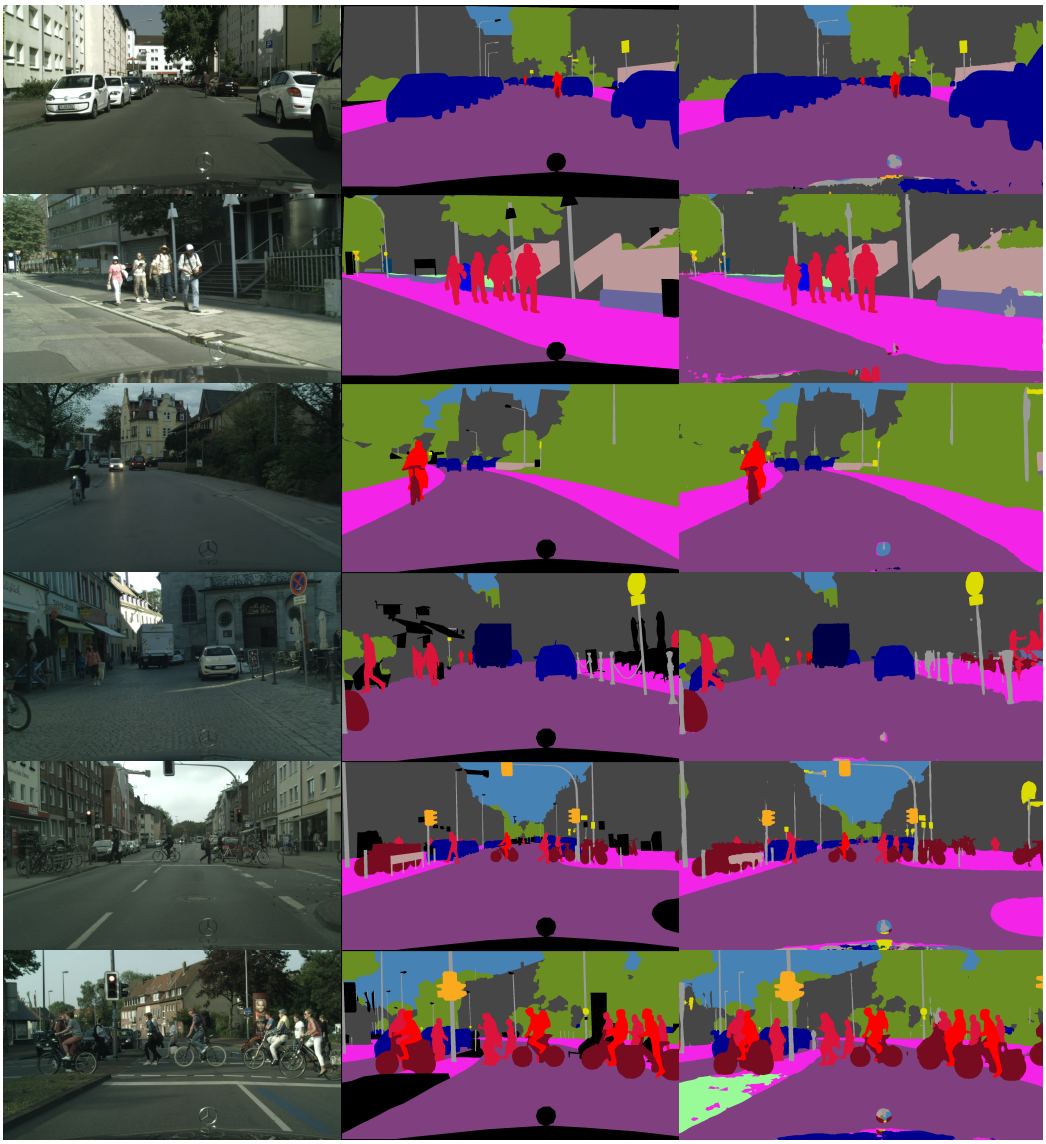


Fig. 9: Visual results of our method SPFNet on Cityscapes dataset. The first row is the original image,the second row is the ground truth while the last row is represents the model performance.

TABLE VIII: INDIVIDUAL CATEGORY RESULTS ON CAMVID TEST SET IN TERMS OF MIOU FOR 11 CLASSES. "-" INDICATES THE CORRESPONDING RESULT IS NOT REPORTED BY THE METHODS.

Method	Building	Tree	Sky	Car	Sign	Road	Ped	Fence	Pole	Sidewalk	Bicyclist	mIoU
SegNet[7]	**88.8**	**87.3**	92.4	82.1	20.5	97.2	57.1	49.3	27.5	84.4	30.7	65.2
BiSeNet1[55]	82.2	74.4	91.9	80.8	42.8	93.3	53.8	49.7	25.4	77.3	50.0	65.6
BiSeNet2[55]	83.0	75.8	92.0	83.7	46.5	94.6	58.8	53.6	31.9	81.4	54.0	68.7
AGLNet[44]	82.6	76.1	91.8	87.0	45.3	95.4	61.5	39.5	39.0	83.1	62.7	69.4
LBN-AA[45]	83.2	70.5	92.5	81.7	51.6	93.0	55.6	53.2	36.3	82.1	47.9	68.0
BiSeNetV2/BiSeNetV2L[19]	-	-	-	-	-	-	-	-	-	-	-	72.4/73.2
SPFNet34H(ours)	86.4	79.2	**92.6**	**90.3**	**56.3**	**95.9**	**68.1**	**54.5**	**42.3**	85.4	**74.2**	**75.1**
SPFNet34L(ours)	84.6	77.2	91.6	90.2	49.9	95.6	61.5	47.9	35.4	85.0	66.1	71.4
SPFNet18H(ours)	84.7	78.0	92.5	89.9	52.0	95.4	66.0	43.1	40.7	84.1	67.7	72.2
SPFNet18L(ours)	83.7	76.7	91.8	89.6	49	95.4	59.7	45.2	34.4	83.5	63.7	70.3

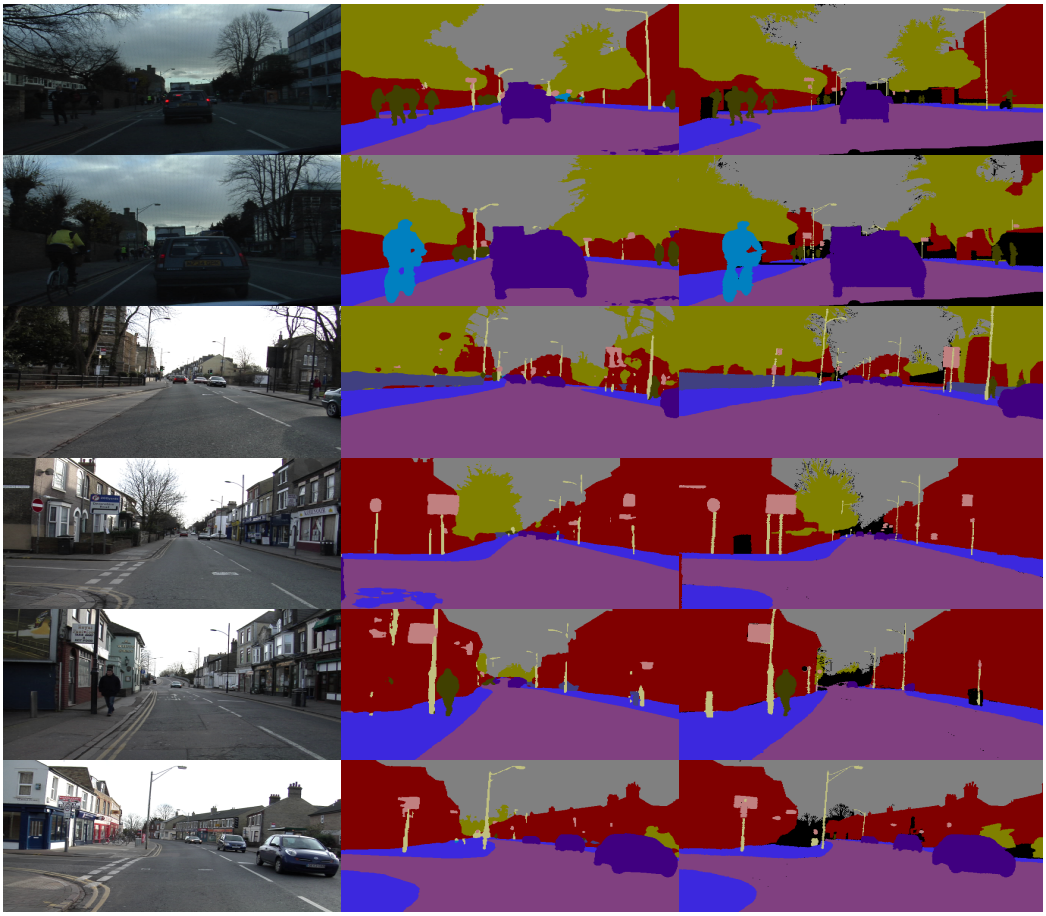


Fig. 10: Visual results of our method SPFNet on Camvid test set. The first row is the image,the second row is the prediction while the last row is ground truth.

SPFNet34H and SPFNet18L obtain 75.7% mIoU and 71.9% mIoU , respectively on Cityscapes test sets.

REFERENCES

[1] J. Janai, F. Güney, A. Behl, A. Geiger *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[2] I. Alonso, L. Riazuelo, and A. C. Murillo, "Mininet: An efficient semantic segmentation convnet for real-time robotic applications," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1340–1347, 2020.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[6] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4151–4160.

[7] 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.

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[9] 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 and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

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

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

[12] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2018–2025.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[14] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[15] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," *arXiv preprint arXiv:1707.07958*, 2017.

[16] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[17] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[19] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, pp. 1–18, 2021.

[20] R. P. Poudel, U. Bonde, S. Liwicki, and C. Zach, "Contextnet: Exploring context and detail for semantic segmentation in real-time," *arXiv preprint arXiv:1805.04554*, 2018.

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

[22] M. A. Elhassan, C. Huang, C. Yang, and T. L. Munea, "Dsanet: Dilated spatial attention for real-time semantic segmentation in urban street scenes," *Expert Systems with Applications*, vol. 183, p. 115090, 2021.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[24] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu, "Disentangled non-local neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 191–207.

[25] Q.-L. Zhang and Y.-B. Yang, "Sa-net: Shuffle attention for deep convolutional neural networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2235–2239.

[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[27] P. Zhang, W. Liu, H. Wang, Y. Lei, and H. Lu, "Deep gated attention networks for large-scale street-level scene segmentation," *Pattern Recognition*, vol. 88, pp. 702–714, 2019.

[28] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[29] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510–519.

[30] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[31] J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao, and H. Lu, "Scene segmentation with dual relation-aware attention network," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[32] X. Li, X. Hu, and J. Yang, "Spatial group-wise enhance: Improving semantic feature learning in convolutional networks," *arXiv preprint arXiv:1905.09646*, 2019.

[33] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: efficient channel attention for deep convolutional neural networks, 2020 ieee," in *CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2020.

[34] R. Saini, N. K. Jha, B. Das, S. Mittal, and C. K. Mohan, "Ulsam: Ultra-lightweight subspace attention module for compact convolutional neural networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1627–1636.

[35] 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.

[36] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, S. Gao, and J. Liu, "Ce-net: Context encoder network for 2d medical image segmentation," *IEEE transactions on medical imaging*, vol. 38, no. 10, pp. 2281–2292, 2019.

[37] C.-W. Xie, H.-Y. Zhou, and J. Wu, "Vortex pooling: Improving context representation in semantic segmentation," *arXiv preprint arXiv:1804.06242*, 2018.

[38] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[39] 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)*, 2018, pp. 116–131.

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

[41] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[42] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[43] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions (2015)," *arXiv preprint arXiv:1511.07122*, 2016.

[44] Q. Zhou, Y. Wang, Y. Fan, X. Wu, S. Zhang, B. Kang, and L. J. Latecki, "Aglnet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network," *Applied Soft Computing*, vol. 96, p. 106682, 2020.

[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*, vol. 22, no. 6, pp. 3258–3274, 2020.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS 2017 Workshop on Autodiff*, 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ

[48] S. R. Bulo, L. Porzi, and P. Kontschieder, "In-place activated batchnorm for memory-optimized training of dnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5639–5647.

[49] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3684–3692.

[50] 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.

[51] P. Bilinski and V. Prisacariu, "Dense decoder shortcut connections for single-pass semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6596–6605.

[52] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[53] 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.

[54] P. Hu, F. Caba, O. Wang, Z. Lin, S. Sclaroff, and F. Perazzi, "Temporally distributed networks for fast video semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8818–8827.

[55] 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.

[56] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[57] 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.

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

[59] G. Li, I. Yun, J. Kim, and J. Kim, "Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," *arXiv preprint arXiv:1907.11357*, 2019.

[60] Y. Zhang, Z. Qiu, J. Liu, T. Yao, D. Liu, and T. Mei, "Customizable architecture search for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 641–11 650.

[61] P. Lin, P. Sun, G. Cheng, S. Xie, X. Li, and J. Shi, "Graph-guided architecture search for real-time semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4203–4212.

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

[63] Z. Yang, H. Yu, M. Feng, W. Sun, X. Lin, M. Sun, Z.-H. Mao, and A. Mian, "Small object augmentation of urban scenes for real-time semantic segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 5175–5190, 2020.