Technical Report on Subspace Pyramid Fusion Network for Semantic Segmentation

Mohammed A. M. Elhassan¹, Chenhui Yang², Chenxi Huang² and Tewodros Legesse Munea²

¹School of Computer Science, Zhejiang Normal University,688 Yingbin Road,Jinhua,Zhejiang,China
²School of Informatics, Xiamen University,Xiamen, China,361005

Abstract—The following is a technical report to test the validity of the proposed Subspace Pyramid Fusion Module (SPFM) to capture multi-scale feature representations, which is more useful for semantic segmentation. In this investigation, we have proposed the Efficient Shuffle Attention Module(ESAM) to reconstruct the skip-connections paths by fusing multi-level global context features. Experimental results on two well-known semantic segmentation datasets, including Camvid and Cityscapes, show the effectiveness of our proposed method. The source code is available at

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.

I. INTRODUCTION

Semantic Semantic segmentation is an essential high-level topic in computer vision and has been widely used in various challenging problems such as, medical diagnosis, autonomous vehicles, and scene analysis. Unlike image classification, which aims to classify the whole image, semantic segmentation predicts the per-pixel class for each image content. The current frontiers in semantic segmentation methods are driven by the success of deep convolution neural networks, specifically the fully convolution network (FCN) Framework. In the original FCN approach to obtain larger receptive fields, they increase the depth of the network with more convolutional and downsampling layers. However, simply increasing the number of downsampling operations leads to reduced feature resolution, which causes spatial information loss. On the other hand, the increase in the number of convolutional layers adds more challenges to network optimization.

Over the years, various approaches have been proposed to tackle the drawbacks of the typical FCN-based architecture. Here we listed three categories of methodologies as follows: (i) increasing the resolution of feature maps or maintaining a high-resolution feature across stages, e.g., through decoder network [1, 7, 8], dilated convolutions [9, 10] or high resolution [11]. (ii) improve the segmentation performance with the subsequent approaches, e.g., PSPNet [5], DeepLab [10, 12], PAN [13], DenseASPP [14], RefineNet [15], PPANet [16], and ParseNet [17]. These previous networks enlarge the receptive field to exploit the rich contextual information. Nevertheless, the latter solution is infeasible for real-time applications because it requires heavy networks and expensive computation. (iii) some networks such as [18, 19] have added spatial encoding path to preserve the spatial details to perform fast segmentation with light computation cost, small size and satisfactory segmentation accuracy, there are certain design philosophy that could be followed: (i) incorporate a pre-trained lightweight classification networks [20, 21, 22, 23] as a backbone to construct segmentation architecture such as BiSeNet [19], STDC-Seg [24]. (ii) design a building block that is suitable for low computation cost [25, 26]. (iii) put into consideration the downsampling strategy and the depth of the network [27]. (iv) combine the low-level and high-level features using the multi-path strategy in addition to that, restricting the input image size plays an important part in increasing the inference speed.

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
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.

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 shuffle attention module (ESAM) with two branches. Through the lower branch, i.e., ESAM 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 Figure 3.

The proposed SPFNet is validated with two challenging semantic segmentation benchmarks, Camvid [29] and Cityscapes [30]. Figure 1 shows the accuracy and the speed comparison of different methods on Cityscapes dataset.

Furthermore, many existing frameworks [31], [32], [33], [34], [35] have been developed based on FPN [35] to exploit the inherent multi-scale feature representation of deep convolutional networks. More specifically, FPN-based architectures combine large-receptive field features, low-resolution- with small-receptive-field features, high-resolution to capture objects at different scales. For instance, [34], [35] utilize the lateral path to fuse adjacent features in a top-down manner. This way FPN strengthen the learning of multi-scale representation. Until now, feature pyramid network based approaches have achieved state-of-arts performance in object detection. Our main contributions are summarized as follows:

1) A novel Subspace Pyramid Fusion (SPFM) module is proposed to learn multi-scale representations.
2) We introduce the Efficient Shuffle Attention Module (ESAM) module, which utilizes channel shuffle operation to integrate the complementary global context.
3) We proposed an architecture based on SPFM and ESAM modules that can be easily applied for road scene understanding. Furthermore, these modules can be plugged in any network.

II. RELATED WORK

A. Efficient Network Designs

In recent literature, real-time model design has become an essential part of computer vision research development. Several lightweight architectures were introduced as a trade-off between accuracy and latency SqueezeNet [20], MobileNet [21], [36], ShuffleNet [37], [22], and Xception [23]. SqueezeNet reduces the number of parameters through the squeeze and expansion module. MobileNetV1 implemented depth-wise separable convolutions to reduce the number of parameters. MobileNetV2 proposed an inverted residual block to alleviate the effect of depth-wise separable convolutions. ShuffleNet and Xception utilize group convolution to reduce FLOPs. ResNet [38] adopted residual blocks to obtain outstanding performance. These models were designed primarily for image classification tasks. Later on, incorporated to work as backbones for semantic segmentation networks to achieve real-time semantic segmentation.

B. Real-time Segmentation

Real-time semantic segmentation methods aim to predict dense pixels accurately while maintaining high inference speed. Over the years there are different approaches have been proposed in this regard. ENet [39], ERFNet [26], and ESPNet [40], [41] proposed compact semantic segmentation networks that utilize lightweight backbones. LEDNet [42], DABNet [25]
introduced a lightweight model to address the computation burden that limits the usage of deep convolutional neural networks in mobile devices. The former used asymmetric encoder-decoder-based architecture and implemented channel split and shuffle operations to reduce the computation cost. The latter structured a depth-wise symmetric building block to jointly extract local and global context information. FDDWNet [43] utilized factorized depth-wise separable convolutions to reduce the number of parameters and the dilation to learn multi-scale feature representations. ICNet [44] proposes a compressed version of PSPNet based on an image cascade network to increase the semantic segmentation speed. DFANet [27] obtains the multi-scale feature propagation by utilizing a sub-stage and sub-network aggregation. Tow-pathway networks such as BiSeNetV2 [28] and DSANet [18] Fast-SCNN [45] introduce a shallow detailed branch to extract the spatial details and a deep semantic branch to extract the semantic context. This paper proposes a lightweight model that utilizes (ResNet18 or ResNet34) as backbones.

### 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

Figure 3 illustrates the architectural framework of SPFNet, designed within a conventional encoder-decoder paradigm. The encoder is responsible for downsampling the input data to acquire diverse multi-scale features, while the decoder focuses on reconstructing high-level features from low-level ones, facilitating pixel-level semantic predictions. SPFNet initiates the process by taking an input image and encoding its features through a widely-used multi-scale backbone, specifically ResNet34 [38], with hierarchical representations \([F^b_1, F^b_2, F^b_3, F^b_4, F^b_5]\). The notation \([F^{egca}_2, F^{egca}_3, F^{egca}_4, F^{egca}_5]\) denotes the rich context information extracted from the feature hierarchy within the Efficient Shuffle Attention Module (ESAM) (see section III-C).

#### B. Subspace Pyramid Fusion Module (SPFM)

In the realm of semantic segmentation research [10], [46], [13], it has been substantiated that augmenting receptive fields is advantageous for the semantic segmentation task. A conventional encoder-decoder architecture involves the extraction of global context information by the encoder from the input image, encompassing the adjacent and class characteristics of the object. However, the transmission of information to shallower layers is known to attenuate the extraction of context information owing to downsampling processes. Motivated by this observation and drawing inspiration from prior works such as [10] and [47], we introduce a pioneering Subspace Pyramid Fusion Module (SPFM) in this subsection.

Within the framework of SPFM, each subspace feature undergoes a singular reduced pyramid pooling (RPP) treatment. In contrast to methodologies presented in [10] and [48], SPFM, illustrated in Figure 4, diverges in its approach to collecting multi-scale contextual information. Rather than directly fusing dilated convolutions, SPFM initiates the generation of multi-scale features for each split using a reduced pyramid pooling module. The RPP is constructed with only two parallel dilated convolutions, employing dilation rates of (4,8). Subsequently, pixel shuffle is employed to enhance feature resolution. Additionally, adaptive average pooling is integrated with the output from two parallel convolutions to capture complex contextual information. The resulting RPP blocks are concatenated within the Subspace
Pyramid Fusion Module. A distinctive characteristic of SPFM, as opposed to \cite{47}, lies in the approach to information aggregation. While \cite{47} integrates attention across different stages in pre-existing compact backbones to facilitate global information learning, SPFM achieves an increase in receptive fields through dilated convolutions and adaptive average pooling to capture global context.

Given the feature map $F^b_i \in \mathbb{R}^{H \times W \times C}$ The objective of our study is to efficiently capture multi-scale features from the feature map $F^b_i \in \mathbb{R}^{H \times W \times C}$ obtained from the backbone, where $H$ and $W$ represent the spatial dimensions of the feature maps, and $C$ denotes the number of channels. In Figure 4, we introduce the proposed Subspace Pyramid Fusion Module (SPFM) and its submodule Reduced Pyramid Pooling (RPP) in Figure 4. SPFM segments the input feature map $F^b_i$ into $s$ splits along the channel dimension. 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 $F^p_i$ 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 $P^p_{i,1}$ as in Eq. 1 and $P^p_{i,2}$ Eq. 2 Then, we concatenate the output features $P^p_{i,c}$ Eq. 3 and refined it with three consecutive convolutions with kernel sizes $(1 \times 1), (3 \times 3)$, and $(1 \times 1)$ 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 $P^p_{s,i}$ as shown in Eq. 5.

\[
P^p_{i,1} = \text{PS}(\text{D}_{\text{conv}}@4(F^p_i)) \quad (1)
\]
\[
P^p_{i,2} = \text{PS}(\text{D}_{\text{conv}}@8(F^p_i)) \quad (2)
\]
\[
P^p_{i,c} = \text{Concat}([P^p_{i,1}, P^p_{i,2}]) \quad (3)
\]
\[
P^p_i = f^1(\text{APool}(F^p_i)) \quad (4)
\]

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

\[
P^p_{s,i} = A^p_i \oplus P^p_{i,c} \quad (5)
\]

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

**Subspace Pyramid Fusion Module (SPFM):** The final output for SPFM module $F^{out}_s$ is obtained by concatenating RPP of all splits (Eq. 6).

\[
F^{out}_s = f_1(\text{Concat}([\hat{P}_{s,1}, \hat{P}_{s,2}, ..., \hat{P}_{s,s}])) \quad (6)
\]

**C. Efficient Shuffle Attention Module(ESAM)**

The significance of local and global context information in semantic segmentation and various computer vision tasks has been acknowledged, even in the traditional machine learning era \cite{49}. The Efficient Shuffle Attention Module (ESAM) is introduced to amalgamate channel attention via the shuffle unit operator, thereby enhancing cross-feature interaction among sub-features. In this context, the input features from the backbone are represented as $F^b_j$, where $j$ denotes the level in the encoder (refer to Figure 3). Considering the distinct semantic content across different levels, ESAM is applied before upsampling to a common resolution using bilinear interpolation. Subsequently, features from all encoder scales and the final decoder-based subpixel convolution outputs are concatenated to formulate a multi-scale feature map. This process facilitates the fusion of information from both shallow 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 $F^{\text{ESAM}}_j$ of efficient shuffle attention
module is computed using the following equation:

\[ F_j^{esa} = \hat{F}_j \oplus F_j^a \]  

(7)

Where \( \oplus \) represents the element-wise summation, \( \hat{F}_j \) is the output of shuffle attention part, and \( F_j^a \) represents the output feature of the upper part. At first, the input feature \( F_j^b \) 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 \( F_j^b \) is passed through depthwise separable convolutions, leading to feature map of \( F_j^u \in \mathbb{R}^{H \times W \times C} \) as shown in Eq. 8

\[ F_j^u = f_{DW}^3(f_1(F_j^b)) \]  

(8)

where \( f_{DW}^3(.) \) represents depthwise convolution with kernel \( 3 \times 3 \).

**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 \( F_j^b \) 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.

\[ A_{j,1} = f_{PW}^1(MP(f_{DW}^1(F_{j,1}))) \]  

(9)

\[ A_{j,2} = f_{PW}^1(MP(f_{DW}^1(F_{j,2}))) \]  

(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.

\[ A_j = Softmax(A_{j,1} \oplus A_{j,2}) \]  

(11)

The output \( \hat{F}_j \) is computed using Eq. 12

\[ \hat{F}_j = Shuffle((A_j \odot F_j^b) \odot F_j^b) \]  

(12)

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

The obtained output undergoes normalization through the softmax function, as expressed in Equation 11. Subsequently, element-wise multiplication and summation operations are implemented within residual-like connections involving the input feature \( A_j \). Additionally, we incorporate a channel shuffle operator, akin to ShuffleNet [22], to facilitate cross-group information communication. The resulting channel attention output, as delineated in Equation 12, is then summed with the upper branch to compose the Efficient Shuffle Attention Module (ESAM), as outlined in Equation 7. Importantly, the final output dimensions of the ESAM module align with those of \( F_j^b \), rendering ESAM a straightforward and adaptable component that can be seamlessly integrated into diverse stages of the encoder. This flexibility empowers the module to acquire multi-scale features with distinct semantic attributes. Consequently, the ESAM module proves conducive to effortless integration into contemporary semantic segmentation architectures and analogous tasks. The impact of the proposed efficient context aggregation module is further illustrated in Table ??, where evaluations encompass floating-point operations per second (FLOPS), parameters, processing speed, and mean intersection over union.

### IV. EXPERIMENTS

In this section, we conduct a comprehensive set of experiments to assess the efficacy of the proposed SPFNet. The evaluations are performed on the Camvid [29] and Cityscapes [30] datasets. The experimental results showcase SPFNet’s state-of-the-art performance on the Camvid dataset and comparable results to other methods on the Cityscapes dataset. The subsequent sections detail the datasets and implementation...
specifics, followed by a series of ablation studies conducted on the Camvid and Cityscapes datasets to elucidate the impact of each component in SPFNet. Finally, we present a comparative analysis of our method with other state-of-the-art networks on both datasets.

### A. Experiments Settings

1) **Camvid:** The Camvid dataset [29] comprises 376 training examples, 101 evaluation images, and 233 testing images. To ensure a fair comparison with other state-of-the-art models, we assessed our model across 11 classes, including building, sky, tree, car, and road. The 12th class was designated as an ignore class to accommodate unlabelled data. The dataset’s small size and imbalanced label distribution pose significant challenges.

2) **Cityscapes:** The Cityscapes dataset [30] serves as an urban scene understanding benchmark, featuring 5000 high-resolution images of dimensions 2048 x 1024 pixels with fine annotations from various cities. The dataset is divided into 2975 images for training, 500 images for validation, and the remaining 1525 images for testing. The evaluation spans 19 semantic segmentation classes.

The subsequent sections delve into the ablation studies, exploring the impact of each constituent in SPFNet on the Camvid and Cityscapes datasets. The experimental design aims to comprehensively assess the model’s performance across diverse scenarios. Finally, a comparative analysis is presented, juxtaposing SPFNet against other state-of-the-art networks on the two datasets.

3) **Implementation Details:** To implement our model, we use Adam optimizer [54] with weight decay $5 \times 10^{-6}$, power 0.9, and the starting learning rate $l_{init}$ for Camvid and Cityscapes datasets is set to $3 \times 10^{-4}$. The network optimization utilized the Poly learning rate policy, as described in Equation [13] a strategy consistent with prior works such as [10], [55]. All experiments were conducted using the PyTorch framework [55] on an NVIDIA 3090 RTX GPU, with training extending over 150 epochs for Camvid and 500 epochs for Cityscapes. Notably, BatchNorm layers in SPFNet, excluding those in the encoder, were substituted with InplaceBN-Sync [56] to curtail memory usage, enhance speed, and facilitate FLOPS analysis, which was conducted using an Nvidia GTX1080 GPU. Consistent data augmentation was applied across all experiments, involving random scaling between 0.75 and 2.0, random horizontal flips, and random cropping of $512 \times 1024$ and $360 \times 480$ image patches for training Cityscapes and Camvid, respectively. Weighted cross-entropy loss was employed for training on both datasets. To assess the experimental results of our model, comprehensive comparisons were conducted on the Cityscapes validation and testing sets against current state-of-the-art segmentation models. Further detailed analyses are presented in subsequent sections.

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

Where $max\_iter$ is the maximum number of iterations.

### B. Ablation Study

In this subsection, a series of experiments is conducted to assess the individual contributions of components within the proposed SPFNet. Initially, a baseline network is established, employing an encoder-decoder architecture consisting of a ResNet-based encoder and a decoder incorporating subpixel convolution. This baseline serves as the foundation for the subsequent ablation study.

1) **Ablation for Efficient Shuffle Attention Module (ESAM):** To assess the impact of the Efficient Shuffle Attention Module (ESAM) at each stage, we seamlessly integrate our proposed ESAM into a baseline U-shaped network architecture, denoted as ResNet34H (encoder) and DSC (decoder). The experimental results, presented in Table ??, encompass evaluations in terms of FLOPS, parameters, processing speed, and mean Intersection over Union (mIoU). This integration enables the model to acquire multi-level cross-feature communication and mitigate the semantic gap among features from different levels. Importantly, the Subspace Pyramid Fusion Module (SPFM) is not incorporated into the network at this juncture. The introduction of ESAM leads to a marginal increase in computational requirements, as evident from the FLOPS results. However, the corresponding improvement in segmentation accuracy, as demonstrated in Table ??, substantiates the efficacy of ESAM. Notably, integrating ESAM in the second stage results in a slightly higher FLOPS compared to other stages. In the configuration utilizing ESAM across stages 2 to 5, the mIoU experiences an improvement of over 2.84% compared to the baseline. It is noteworthy that ESAM incurs a reduction in processing speed by 5.7 FPS compared to the baseline.

2) **Ablation on Subspace Pyramid Fusion Module (SPFM):** In our hyperparameter analysis on the Subspace Pyramid Fusion Module (SPFM) applied to the Camvid dataset, we systematically examine the impact of the hyperparameter $s$, representing the number of splits used in each Reduced Pyramid Pooling (RPP) module within the SPFM. The evaluations

| Method                  | Road | Sidewalk | Building | Wall | Sky | Person | AgNet                   | Ground | Car | Truck | Bus | Train | Motor | RPP | Indications |
|-------------------------|------|----------|----------|------|-----|--------|-------------------------|--------|-----|-------|-----|-------|-------|-----|-------------|
| **Cityscapes**          |      |          |          |      |     |        |                         |        |     |       |     |       |       |     |             |
| FCN                     | 90.5 | 77.9     | 98.2     | 47.6 | 47.7 | 47.6   | 89.2                    | 97.4   | 89.2 | 89.1  | 73.9 | 89.2  | 73.9  | 89.2 |             |
| DilationNet             |      |          |          |      |     |        |                         |        |     |       |     |       |       |     |             |
| BiSeNet                 | 97.6 | 79.2     | 98.9     | 47.3 | 58.3 | 63.0   | 85.5                    | 97.2   | 92.4 | 91.6  | 77.1 | 97.2  | 70.4  | 97.2 |             |
| BiSeNetV2              | 97.8 | 80.1     | 98.9     | 47.6 | 58.3 | 63.0   | 85.5                    | 97.5   | 92.5 | 91.6  | 77.1 | 97.2  | 70.4  | 97.2 |             |
| LBN, SAA               | 98.2 | 84.0     | 91.6     | 56.7 | 69.5 | 69.0   | 79.6                    | 98.0   | 92.6 | 92.4  | 76.8 | 98.0  | 71.6  | 98.0 |             |
| OtherSPFNet            | 98.2 | 84.0     | 91.6     | 56.7 | 69.5 | 69.0   | 79.6                    | 98.0   | 92.6 | 92.4  | 76.8 | 98.0  | 71.6  | 98.0 |             |
| OurSPFNet              | 98.4 | 84.9     | 92.2     | 47.6 | 54.1 | 65.9   | 73.6                    | 76.2   | 93.0 | 86.4  | 85.3 | 86.4  | 75.0  | 95.2 |             |

**Table I:** The per-class, class, and category IOU evaluation on the Cityscapes test set. "-" indicates the corresponding result is not reported by the methods.
are conducted on both the Camvid and Cityscapes datasets, utilizing an Nvidia GTX1080 GPU. Specifically, we investigate four scenarios with ResNet18 (SPFNet18L) as the backbone and another four scenarios with ResNet34 (SPFNet23L), considering different values for $s$, namely $s = 2, 4, 8, 16$. Initially, to establish a baseline, we replace the SPF with a $1 \times 1$ Convolution layer, and the results for the Camvid dataset are reported in Table III. When $s = 2$, the SPF fuses two Reduced Pyramid Fusion Modules to generate multi-scale features. As evident in Table III leveraging SPF with varying numbers of splits demonstrates higher accuracy, measured in terms of mean Intersection over Union (mIoU), compared to the baseline. Moreover, models with larger values of $s$ exhibit improved segmentation accuracy, indicating that higher $s$ enhances the network’s ability to extract complex multi-scale features for both SPFNet34L and SPFNet18L, albeit at the expense of speed (Frames Per Second, FPS). Table III and Figure 6 reveal a performance drop in SPF when $s = 8$ for both backbones in comparison to the baseline. Interestingly, utilizing 2 splits adds more parameters and Floating-Point Operations per Second (FLOPS) than larger $s$ options, resulting in faster inference. We further compared SPF with four splits in terms of FLOPS, number of parameters, and speed. As outlined in Table III a larger $s$ introduces fewer FLOPS and parameters, yet it diminishes speed. Considering both efficiency and segmentation accuracy, our analysis concludes with the selection of $s = 4$ for SPF.

Hyperparameter analysis on SPF Module on the Cityscapes dataset: We conducted similar analysis to examine the hyperparameter $s$ of the SPF 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 7 Experimental results and evaluation shows that the hyperparameter $s$ follows the same patterns as on Camvid dataset.

3) Ablation of SPF and Other Multi-scale Modules: To comprehensively investigate the impact of various modules within our proposed network architecture, we initially configured a modified ResNet34 by abstaining from downsampling in the second stage of the encoder. Subsequently, we integrated the Dilated Spatial Convolution (DSC) module and the Effi-
TABLE V: PERFORMANCE COMPARISON OF SPF 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+ESAM                    | 300.0  | 38.7   | 13.6 | 75.0     |
| ResNet34+DSC+SPPM                    | 286.6  | 40.8   | 16.3 | 75.8     |
| ResNet34+DSC+ESAM+Dense ASPP         | 380.7  | 77.9   | 8.7  | 75.21    |
| ResNet34+DSC+ESAM+ASPP               | 304.8  | 41.0   | 12.3 | 75.64    |
| ResNet34+DSC+ESAM+Vortex Pooling     | 322.0  | 49.4   | 11.1 | 75.72    |
| ResNet34+DSC+ESAM+SPFM               | 317.0  | 41.8   | 12.4 | 78.04    |

efficient Shuffle Attention Module (ESAM) module as baselines to assess different multi-scale feature extractor modules. These ablation experiments were conducted over 500 epochs.

In particular, we performed a series of experiments to compare our proposed Subspace Pyramid Fusion Module (SPFM) with other multi-scale pyramid modules, namely Atrous Spatial Pyramid Pooling (ASPP) from Deeplab [10], Vortex Pooling [48], and DenseASPP [14]. In these experiments, we opted for a somewhat lighter backbone, employing the aforementioned ResNet18 and a modified ResNet34. The results, summarized in Table [V], reveal that the multi-scale modules enhance the baseline. Notably, the baseline model incorporating our proposed SPFNet outperforms those with ASPP (75.64% mIoU), Vortex Pooling (75.72% mIoU), and DenseASPP (75.21% mIoU), achieving the highest accuracy with a 78.04 mIoU. Moreover, the SPF module variant requires fewer parameters and exhibits a speed closer to ASPP. Additionally, SPFNet incurs lower computational cost, measured in terms of parameters and Floating-Point Operations per Second (FLOPS), compared to both DenseASPP and Vortex Pooling. This superiority could be attributed to the SPFNet module’s efficacy in extracting complex multi-scale semantic information more efficiently. SPFNet concurrently learns multiple Reduced Pyramid Pooling (RPP) blocks, utilizing small dilation rates to generate feature maps of different resolutions, thereby minimizing noise and enhancing its overall performance.

C. Results on Cityscapes Dataset

Table [VI] presents a comprehensive comparison of the proposed SPFNet against other state-of-the-art models on the Cityscapes dataset, considering factors such as Floating-Point Operations per Second (FLOPS), parameters, and speed. The experiments were conducted on both the training and validation sets, utilizing a reduced input resolution of 2048 × 1024 2048x1024 to enhance training efficiency. Subsequently, model segmentation accuracy was evaluated on the test set, and the results were submitted to the Cityscapes dataset online server (https://www.cityscapes-dataset.com/submit/) to obtain benchmark results. The comparison encompasses both smaller networks such as ICNet [44], BiSeNet [57], and larger networks including PSPNet [5], RefineNet [15], and Deeplab [48]. Notably, SPFNet34H achieves a mean Intersection over Union (mIoU) of 75.7%, featuring 41.9 million parameters and an inference speed of 12.7 frames per second (FPS). On the other hand, SPFNet18L achieves a mIoU of 71.9%, with 31.7 million parameters and an inference speed of 46.5 FPS. It is worth highlighting that SPFNet employs ResNet34 and ResNet18, which have fewer parameters compared to many larger models in Table [VI] yet achieve comparable results. Importantly, our models do not employ multi-scale testing or multi-crop evaluation, techniques often utilized to enhance accuracy by other practitioners. Figure [8] illustrates visual examples of SPFNet’s performance on the Cityscapes validation set, providing qualitative insights into its segmentation capabilities.

D. Results on Camvid Dataset

Table [VIII] presents the segmentation accuracy of our proposed method on the Camvid test set, offering a comparative analysis with other state-of-the-art methods. The model was trained with an input resolution of 360 × 480 pixels, utilizing only the training and validation samples for training and validation, respectively. The testing samples were reserved to obtain results for comparison with other state-of-the-art models. Notably, our model exhibits superior accuracy across various configurations. The SPFNet34H, leveraging ResNet34 as its backbone, achieves a significant 75.1% mean Intersection over Union (mIoU) on the test set, accompanied by an inference speed of 33.0 frames per second (FPS), representing a well-balanced trade-off between speed and accuracy. The SPFNet43L, incorporating downsampling at stage 2, achieves a 71.4% mIoU with a faster inference speed of 79.3 FPS. Additionally, we evaluate our methods with two configurations of ResNet18. SPFNet18H attains a 72.2% mIoU with an inference speed of 39.7 FPS, while SPFNet18L achieves a 70.3% mIoU with a higher inference speed of 109 FPS. These experiments validate the efficacy of the proposed architecture design. As demonstrated in Table [VIII] training the model with extra data enhances performance for some methods. For instance, BiSeNetV2* [28], when trained without extra data, achieves a 72.4% mIoU, and its performance improves to 76.7% when the model is initialized on Cityscapes data. Similarly, BiSeNetV2-L* improves from 73.2% mIoU, without extra data, to 78.5% with the same settings. Individual category results on the Camvid test set are reported in Table [VIII]. Finally, we showcase visual results of our SPFNet method on the Camvid test set.

V. CONCLUSION

This paper introduces the Subspace Pyramid Fusion Module (SPFM), a novel method designed for learning multi-scale context information by partitioning input feature maps into distinct subspaces. Additionally, we propose the Efficient Shuffle Attention Module (ESAM), which leverages the channel-shuffle operator to facilitate enhanced information communication across various sub-features. Combining these innovations, we present a semantic segmentation architecture, termed SPFNet, specifically tailored to address the challenges of multi-scale feature fusion. Ablation studies conducted on the Cityscapes
| Method          | Backbone       | Resolution | Params (M) | FLOPS (G) | Speed (FPS) | test set | mIoU  |
|-----------------|----------------|------------|------------|-----------|-------------|----------|-------|
| CRF-RNN[50]     | VGG16          | 512×1024   | -          | 457.8     | 1.4         | ✓        | 62.5  |
| Deeplab[58]     | VGG16          | 512×1024   | 262.1      | -         | 0.25        | ✓        | 63.1  |
| FCN[5]          | VGG16          | 512×1024   | 134.5      | 136.2     | 2           | ✓        | 65.3  |
| Dilation10[51]  | VGG16          | 512×1024   | 140.8      | -         | -           | ✓        | 67.1  |
| RefineNet[15]   | ResNet101      | 512×1024   | 118.1      | 526       | 9.1         | ✓        | 73.6  |
| PSPNet[5]       | ResNet101      | 713×713    | 250        | 412.2     | 0.78        | ✓        | 78.4  |
| CRF-RNN[50]     | VGG16          | 512×1024   | -          | 457.8     | 1.4         | ✓        | 62.5  |
| Deeplab[58]     | VGG16          | 512×1024   | 262.1      | -         | 0.25        | ✓        | 63.1  |
| FCN[5]          | VGG16          | 512×1024   | 134.5      | 136.2     | 2           | ✓        | 65.3  |
| Dilation10[51]  | VGG16          | 512×1024   | 140.8      | -         | -           | ✓        | 67.1  |
| RefineNet[15]   | ResNet101      | 512×1024   | 118.1      | 526       | 9.1         | ✓        | 73.6  |
| PSPNet[5]       | ResNet101      | 713×713    | 250        | 412.2     | 0.78        | ✓        | 78.4  |

Fig. 8: 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.
Fig. 9: 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.

Table VII: The accuracy, speed and parameters comparison of the proposed method against other semantic segmentation methods on the Camvid test set. * 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    | 2017 | 720 × 960  | 262.1      | 4.9         | 61.6  |
| PSPNet       | 2017 | 720 × 960  | 250        | 5.4         | 69.1  |
| DenseDecoder | 2016 | 720 × 960  | -          | -           | 70.9  |
| SPFNet34H    | 360  | 41.8       | 41.8       | 37          | 71.4  |
| SPFNet18H    | 360  | 31.7       | 31.7       | 109         | 70.3  |

Dataset underscore the effectiveness of the proposed SPFNet modules. The performance evaluation on the Camvid dataset demonstrates the efficacy of SPFNet across various configurations. Specifically, SPFNet34H achieves a mean Intersection over Union (mIoU) of 75.1%, SPFNet34L achieves 71.4% mIoU, SPFNet18H achieves 72.2% mIoU, and SPFNet18L achieves 70.3% mIoU. Furthermore, SPFNet34H and SPFNet18L exhibit competitive results on the Cityscapes test sets, achieving mIoUs of 75.7% and 71.9%, respectively. These findings underscore the potential of SPFNet in addressing semantic segmentation tasks with improved accuracy and efficiency.

References

1. 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.

2. M. Saha and C. Chakraborty, “Her2net: A deep framework for semantic segmentation and classification of cell membranes and nuclei in breast cancer evaluation,” IEEE Transactions on Image Processing, vol. 27, no. 5, pp. 2189–2200, 2018.

3. M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, “A comparative study of real-time semantic segmentation for autonomous driving,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 587–597.
| Method              | Building | Tree | Sky | Car | Sign | 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   |
| BiSeNet11 [19]      | 82.2     | 74.4 | 91.9 | 80.8 | 42.8 | 93.3 | 53.8  | 49.7 | 25.4     | 77.3      | 50.0   |
| BiSeNet21 [19]      | 83.0     | 75.8 | 92.0 | 83.7 | 46.5 | 94.6 | 58.8  | 53.6 | 31.9     | 81.4      | 54.0   |
| AGLNet [20]         | 82.6     | 76.1 | 91.8 | 87.0 | 45.3 | 95.4 | 61.5  | 39.5 | 39.0     | 83.1      | 62.7   |
| LBN-AAS [21]        | 83.2     | 70.5 | 92.5 | 81.7 | 51.6 | 93.0 | 55.6  | 53.2 | 36.3     | 82.1      | 47.9   |
| BiSeNetV2/BiSeNetV2L [23] | -       | -    | -   | -   | -   | -   | -    | -   | -        | -         | 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   |
| SPFNet34L (ours)    | 84.6     | 77.2 | 91.6 | 90.2 | 49.9 | 95.6 | 61.5  | 47.9 | 35.4     | 85.0      | 66.1   |
| SPFNet18H (ours)    | 84.7     | 78.0 | 92.5 | 89.9 | 52.0 | 95.4 | 66.0  | 43.1 | 40.7     | 84.1      | 67.7   |
| SPFNet18L (ours)    | 83.7     | 76.7 | 91.8 | 86.9 | 49   | 95.4 | 59.7  | 45.2 | 34.4     | 83.5      | 63.7   |

[4] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, “Psanet: Point-wise spatial attention network for scene parsing,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 267–283.

[5] 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.

[6] 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.

[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] Z. Zhou, M. M. R. Siddiquie, 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.

[9] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” arXiv preprint arXiv:1511.07122, 2015.

[10] 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.

[11] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[12] 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.

[13] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[14] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9522–9531.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.

[22] 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.

[23] C. Yu, J. Wang, C. Gao, G. Yu, and N. Sang, “Bisenet: Bilateral network with guided aggregation for real-time semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9522–9531.

[24] G. E. Glaser, D. C. Geiger, and T. D. Hospedales, “Panoptic feature pyramid networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.

[25] A. Kirillov, R. Girshick, K. He, and P. Dollár, “Panoptic feature pyramid networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6399–6408.

[26] T.-Y. Lin, P. Dollar, 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.

[27] 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.
