SPADe: Spatial Plaid Attention Decoder for Semantic Segmentation of Street Views

Lijun Xie, Xiaohui Yuan, *Senior Member, IEEE*, Abolfazl Meyarian, Zhinan Qiao, Zhenchun Wei, *Member, IEEE*, and Lichuan Gu

***Abstract*—The decoder is a key component in deep networks for the semantic segmentation of street views. The existing methods rely on the limited receptive field for feature extraction without considering the contextual information, which could lead to errors in understanding complex scenes. Moreover, a balance of contextual information and computational cost must be considered to meet the needs of real-world applications. To address these problems, we introduce a Spatial Plaid Attention Decoder network, which uses a lightweight decoder with Spatial Plaid Attention to perform highly efficient operations for semantic segmentation. With approximately 4 million parameters (9.75% of the UPerNet), our decoder achieves state-of-the-art performance on public datasets such as Cityscapes and ADE20K, with 84.84% and 54.0% mIoU, respectively. In addition, our method reduces the total Flops by 34.95% and 32.85%, respectively. We demonstrate how contextual information helps the network in object recognition and how object features and contextual features contribute to the scene segmentation and recognition.**

***Index Terms*—Image segmentation, deep learning, machine vision.**

I. Introduction

UNDERSTANDING street views plays an important role in many applications such as situation awareness in transportation, event detection, and urban planning [1], [2]. To achieve high-level analysis, objects in images need to be separated and classified, i.e., image semantic segmentation. In semantic segmentation, pixels in the vicinity present the typical context of an object. Correctly segmenting objects usually requires more than the characteristics of the objects.

To leverage the contextual information, the network needs to extract and integrate features from the object and its vicinity, which often implies long-range involvement as a representation of context. Although combining features of various scales in pyramid networks helps recognize objects [3], [4], these methods lack meticulous investigation of contextual information. To address this issue, methods have been developed to extract the context of objects [5], [6], [7], [8]. Techniques include leveraging large convolution kernels, more layers of convolution, and spatial attention. Large convolution kernels help the network assess pixel relations in local and global contexts depending on where they are utilized in the network [9]. More convolution layers grant a larger receptive field, helping analyze the context to a greater extent [10], [11]. Spatial attention refines the pixel features based on the spatial setting of similar pixels in their context, making it easier for the network to make clear boundaries between the semantic categories [5], [12], [13].

Alternatively, contextual information has been integrated into decoders [14], [15]. The U-shaped networks process the features in a hierarchy of scales, fusing features using skip-connections while keeping rich semantic information [14], [16]. The skip connections, however, have a limited capacity to add contextual features [17]. The use of dilated convolution in DeepLab V3+ [18] allows the decoder to capture long-range contextual information with a low computational need. Yet, the performance gains using different atrous rates are limited. Methods following the structure of the population Receptive Fields (pRFs) model human vision for object detections [19], which allows changing the receptive field adaptively according to object size and context. However, it is costly due to the grouped dense convolutions. The limited receptive field restricts the encoding of long-range dependencies, which leads to errors in understanding complex scenes where classification depends on understanding broader relationships.

Inspired by the pRF models of humans, utilizing various receptive fields of different sizes and dilation densities [20], we present a decoder for dense pixel prediction that possesses a receptive field. Our idea forms a Spatial Plaid Attention Decoder (SPADe), a lightweight network capable of capturing long-range dependencies using Spatial Plaid Attention (SPA). Our SPA allows the decoder to compare high- and low-level features and highlight the prominent contextual features at different scales. The contribution of this paper is twofold:

- A hierarchical, lightweight, and accurate decoder that achieves state-of-the-art performance in semantic segmentation on public datasets.
- A Spatial Plaid Attention module to capture the long-range dependencies with a significantly low

Received 5 July 2024; revised 3 April 2025 and 13 June 2025; accepted 27 August 2025. The Associate Editor for this article was J. Hemanth. *(Lijun Xie, Xiaohui Yuan, and Abolfazl Meyarian contributed equally to this work.) (Corresponding author: Xiaohui Yuan.)*

Lijun Xie is with the Department of Orthopedic Surgery, the Second Affiliated Hospital, School of Medicine, Orthopedics Research Institute, and the Key Laboratory of Motor System Disease Research and Precision Therapy of Zhejiang Province, Clinical Research Center of Motor System Disease of Zhejiang Province, Zhejiang University, Hangzhou, Zhejiang 310009, China.

Xiaohui Yuan, Abolfazl Meyarian, and Zhinan Qiao are with the Department of Computer Science and Engineering, University of North Texas, Denton, TX 76205 USA (e-mail: xiaohui.yuan@unt.edu).

Zhenchun Wei is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230002, China.

Lichuan Gu is with the School of Information and Artificial Intelligence, Anhui Agricultural University, Hefei 230036, China.

Data is available on-line at https://github.com/mlghost/SPADe

Digital Object Identifier 10.1109/TITS.2025.3604708

computational cost. Flops by 34.95% and 32.85% on Cityscapes and ADE20K, respectively.

The rest of this paper is organized as follows: Section II reviews the decoder techniques for semantic segmentation. Section III presents our proposed SPADe network with SPA modules. Section IV discusses the experimental results, including a comparison with the state-of-the-art methods and an analysis of network components. Section V concludes this paper with a summary of key findings.

II. Related Work

Decoders following the U-shaped architecture [2], [16] extract features at different scales using a backbone and a decoder upscales the features by integrating information using skip connections from the immediately higher scale map. The context of the pixels in the early stages of the backbone highlights the texture and boundary of objects. Therefore, using them may grant better boundary consistency. Such context in the later stages of the backbone or the stages of the decoder offers more semantics, such as the adjacency of objects of certain classes. SegNet [14] employs a similar structure to the U-Net. However, it uses the pre-trained feature maps of its backbone network, i.e., VGG-16. Swin Transformer [21] has also been combined with this U-shape structure [22], showing promising results in medical image segmentation.

Another strategy to use multiscale contextual information is the pyramidal structure [23]. One of the decoders with a pyramid structure is FPN [3], which uses a pre-trained backbone network to extract features from different stages, extracting information at multiple scales. This operation fuses the information of pixels in different scales, leveraging the semantic consistency of the lower scale and richer edges of the higher scale. UPerNet [4] uses the FPN structure but leverages a PSP [15] module on top of the feature map of the fifth stage. The pooling technique aggregates the feature map and is often performed at different rates to derive features of multiple scales. Networks such as PSPNet [15] are built upon such ideas. Features are upscaled and concatenated channel-wise with the original features. The concatenation provides more contextual information at each pixel location.

The Multi-Layer Perceptron (MLP) layers have been used as lightweight decoders, given the recent development of transformer-based models. SegFormer [24] uses an MLP head that takes feature maps of 4 different scales from the backbone as input and upsamples them to match the size of the largest feature map. Features are flattened for the MLP to produce a prediction. SegNext [25] uses the same decoder architecture with additional Hamburger attention [26] before the final layer. Although MLP layers give the advantage of memory efficiency, the relation between pixels cannot be assessed thoroughly, given the limited number of layers and parameters.

Dynamic decoders focus on updating networks with each sample. K-Net [27] generates sample-specific parameters. To take advantage of the clues provided by context, K-Net makes an initial segmentation to lay out the pixel labels and extract deep features, then it generates and refines convolutional kernels specific to each object and class. AdaptIS [28] uses a backbone to generate object centers, and instance-specific filters are created to produce an accurate instance mask.

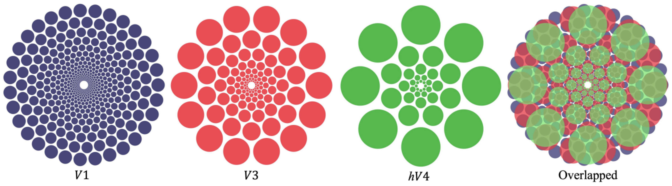


Fig. 1. The spatial array of human receptive fields of $V1$, $V3$, and $hV4$.

The limited receptive field in decoders presents a great challenge in integrating contextual information. While techniques such as multi-scale aggregation, dilated convolutions, and attention mechanisms have improved context modeling, they come with trade-offs in computational complexity, memory usage, and precision. The key research gaps lie in balancing modeling long-range context and computational efficiency. Investigation into efficient decoder architectures that leverage rich contextual information without excessive computational overhead is needed for real-time and edge applications.

III. Spatial Plaid Attention Decoder Network

A. Human Vision Inspiration

The human visual system is a complex network of perception units that process images at various levels of abstraction [20]. The primary cortex, a.k.a. V1, receives the visual inputs to recognize edges and shapes. Subsequently, the V3 unit captures intricate structures that aid in object recognition. High-level parts, such as hV4, specialize in color perception and its integration with shapes for object identification.

Techniques such as functional magnetic resonance imaging (fMRI) have facilitated the understanding of the receptive field structure of V1, V3, and hV4 in the human vision [20]. Fig. 1 visualizes pRFs of V1, V3, and hV4, which shows that as it gets farther from the center, the eccentricity and receptive fields (the circles) increase. Our vision system places greater importance on information within a small vicinity of the center. Nonetheless, regions farther from the center provide a large search space, enabling the benefit from contextual information. The variations in size and spacing between each circle suggest that our vision system analyzes the image using various receptive fields at different levels of spatial density for perception. This inspires us to create a decoder that mimics the receptive field of human vision with a low number of parameters and state-of-the-art performance.

B. Spatial Plaid Attention

The human perception system cares more about the information in a small neighborhood of the center of the field of view, and takes advantage of the information obtainable from distance, serving as the context. To implement such a structure, we consider two components in our SPA module, as shown in Fig. 2: a *Spatial Feature Collection* module that gathers features from the context of the pixel and a *Feature Fusion* module that refines features in a low-dimensional space.

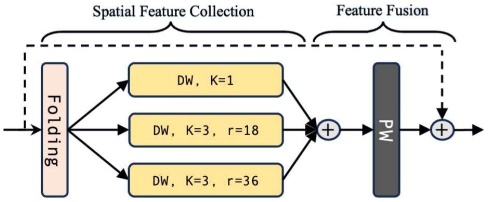


Fig. 2. The structure of SPA. PW: pointwise convolution; DW: depthwise convolution; K: kernel size; r denotes the atrous dilation rate.

The spatial feature collection module uses contextual information for close and long-range coverage. To capture the context in a neighborhood, we devise a folding operation. The features of the 4 immediately adjacent pixels of each pixel are concatenated along the channel dimension:

$$\Gamma(f, i, j) = f_{(i,j)} \odot f_{(i-1,j)} \odot f_{(i+1,j)} \odot f_{(i,j-1)} \odot f_{(i,j+1)} \quad (1)$$

where f denotes the feature map and i and j are the coordinates of a pixel. $\odot$ denotes channel-wise concatenation. Each feature component $f_{(i,j)} \in \mathbb{R}^{1\times 1\times C}$ consists of C channels. This folding operation achieves a spatial feature augmentation, which concatenates pixel features into an integrated form of pixel features and spatial features. It allows the network to extract contextual information. Multiple folding operations in SPA increase the field of view such that features in long-range contextual regions are obtained. We expand the contextual coverage using multi-path, depth-wise (DW [29]) convolutions with different atrous rates to capture the long-range relations.

Although the use of folding in our SPA increases the channel size, the DW convolution only creates $K \times K \times 5C$ kernels, which have a complexity of $\mathcal{O}(C)$ in terms of the number of parameters, versus the normal convolution that is $\mathcal{O}(C^2)$, given the fact that $C \gg 5$.

In the feature fusion module, feature maps extracted by the multi-path, depth-wise convolutions are fused using an element-wise addition, depicted with $\bigoplus$ in Fig. 2. A pointwise (PW) convolution [29] is used to aggregate and refine the features of each pixel, which reduces the number of features from $5C$ to C. This operation reduces the computational cost of processing large feature maps with the SPA. The skip connection ensures smoothness in gradient passing.

C. Network Architecture

Our network, as shown in Fig. 3, uses a backbone (encoder) network to extract features at different scales to ensure the contextual information of objects is derived. The Spatial Plaid Attention Decoder (SPADe) decoder uses multi-contextual features to generate the semantic segmentation map. The network passes three feature maps of high-resolution F_h from the backbone stage of $Back.S_1$, mid-resolution F_m from stage $Back.S_2$, and low-resolution F_h from stage $Back.S_3$ to the SPADe. The extracted features are processed with 1×1 convolutions for channel reduction. Given that the spatial size of the largest feature map F_l is smaller than the input, we perform upsampling to match the size. Unlike self-attention [30] with dense matrix multiplications, our SPA leverages depthwise convolution as the building block to reduce the computation.

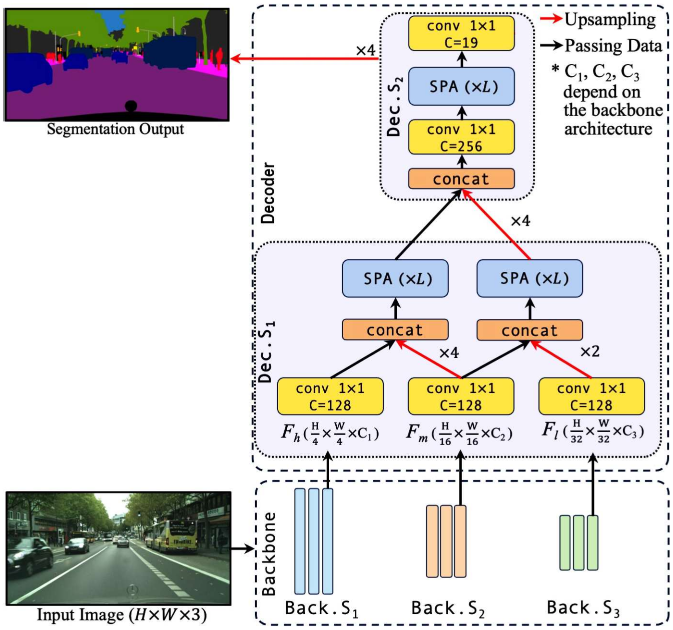
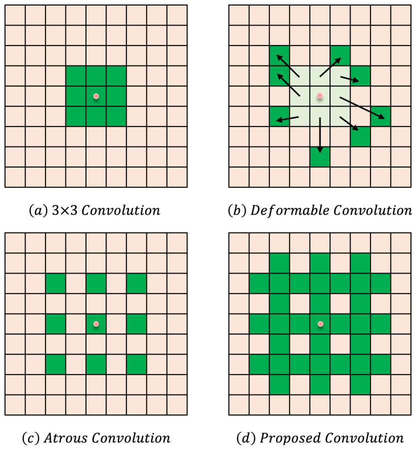


Fig. 3. Network architecture of SPADe with three contextual coverages.

The semantic gap between the backbone's high- and low-resolution features is large. It is important to gradually combine features of each contextual coverage with only the feature map at the two immediately adjacent scales. Fig. 3 shows a SPADe of two stages: $Dec.S_1$ and $Dec.S_2$, in which the feature maps from two consecutive contextual coverage coming from the backbone network are processed together, e.g., $[F_h, F_m]$ and $[F_m, F_l]$. The feature map of lower resolution is upsampled and concatenated channel-wise to the higher resolution one. A SPA module refines the resulting feature map. The same process is repeated on the output of the SPA for each pair of $[F_h, F_m]$ and $[F_m, F_l]$.

Combining features of different scales using our SPA allows the network to fill the semantic gaps between the features of a lower and a higher resolution, leveraging the semantic information of low-resolution features and fine-detail edges of high-resolution ones. The SPA module is repeated multiple times (the number of repetitions is denoted with L in Fig. 3) to process the object features in multi-tier contexts.

The receptive field of a single depth-wise convolution with an atrous rate of at 3, i.e., $r = 3$, is shown in Fig. 4. A 3×3 convolution has a receptive field of 9 cells (see Fig. 4(a)). To increase the contextual coverage, either the kernel size has to be increased or the convolution layers are stacked; both are computationally expensive. Deformable convolution [31] uses offsets to achieve flexible coverage, as shown in Fig. 4(b). The light green cells are the original kernel elements, and the arrows depict the offset vectors. However, the extra offsets increase the computation. Atrous convolution enlarges the receptive field without introducing additional parameters (as shown in Fig. 4(c)). The sparsity makes it necessary to be paired with normal convolution to cover a considerable spatial extent of a feature map [19]. In contrast, our


(a) 3×3 *Convolution*

(b) *Deformable Convolution*

(c) *Atrous Convolution*

(d) *Proposed Convolution*

Fig. 4. The receptive field of (a) conventional convolution, (b) deformable convolution, (c) atrous convolution, and (d) our SPA.

proposed stacked-depth-wise convolutions (Fig. 4(d)) cover a larger area with fewer parameters. The receptive field is significantly enlarged when such a convolution is used in a sequence.

Algorithm 1 SPADe Network for Semantic Segmentation

```
Require: image $I$, number of classes $N$
Ensure: semantic segmentation map $M$
Encoder (Backbone)
$E_1 \leftarrow \text{Conv}(I, k1)$
$E_2 \leftarrow \text{Conv}(\text{Pool}(E_1), k2)$
$E_3 \leftarrow \text{Conv}(\text{Pool}(E_2), k3)$
Bottleneck
$B \leftarrow \text{Conv}(\text{Pool}(E_4), k4)$
Decoder
$D_3 \leftarrow \text{UpConv}(B, E_3, k3)$
$D_2 \leftarrow \text{UpConv}(D_3, E_2, k2)$
$D_1 \leftarrow \text{UpConv}(D_2, E_1, k1)$
$D_1' \leftarrow SPA(Concat(D_1, D_2))$
$D_2' \leftarrow SPA(Concat(D_1, D_2))$
$D_1'' \leftarrow SPA(\text{UpConv}(Concat(D_1', D_2')))$
Output Layer
$M \leftarrow \text{Conv}(D_1'', \text{N}, \text{“softmax”})$
return $M$
```

We summarize our method in Algorithm 1, which presents a two-stage SPADe decoder as shown in Fig. 3. When more stages are desired, the decoder needs to be extended, and the encoder will include more levels of convolutions.

D. Loss Function and Decoder Variations

The loss function is the softmax cross-entropy between the predicted segmentation and the ground truth as follows:

$$\mathcal{L}(P, G) = \sum_{i=1}^{N} G_i \log(P_i) \tag{2}$$

where G is an $H \times W \times S$ tensor for the one-hot encoded ground-truth semantic class (S classes), P is predicted probabilities for the pixels, N is the total number of pixels, and i is the pixel index.

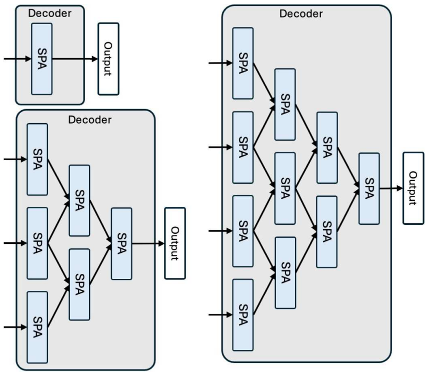


Fig. 5. The decoder network structures with a different number of inputs. The inputs are features extracted by the backbone networks.

The scalability in the design of the SPADe network allows for a different number of backbone networks to be used for feature extraction. Fig. 5 illustrates three variations of the decoder network. For clarity, convolutions and concatenation operations are omitted from these illustrations.

The top left case depicts the decoder that takes one input from the backbone network; the bottom left case depicts a decoder that takes three inputs from the backbone network; and the right case depicts a decoder that takes four inputs from the backbone network. A hierarchical strategy is employed to integrate contextual features. As the number of input features increases, the depth of the SPA modules increases.

IV. Results and Discussion

A. Datasets and Settings

Our experiments use Cityscapes [32] and ADE20K [33]. Cityscapes consists of 5,000 images of 19 semantic classes, which are randomly divided into 2,975 for training, 500 for validation, and 1,525 for testing. Each image has a size of 1024×2048. The other dataset is ADE20K, which consists of 150 semantic classes with 20,210 training images, 2,000 for validation, and 3,000 for testing.

We use ConvNeXt, Swin Transformer, ResNet, InternImage, and DaViT as the backbone networks in comparison. AdamW is used with a weight decay of 0.05 for ConvNeXt, InternImage, and ResNet backbones, and 0.01 for the Swin and DaViT. The initial learning rate used is 1e-6, and we use *‘poly’* scheduling with a factor of 0.9 to adjust the learning rate. Our training has a warm-up stage of 1,500 steps. The batch size for the training using the Cityscapes dataset is 8, and the batch size for the training using ADE20K is 16. Our models are trained on Cityscapes and ADE20K for 80K and 160K steps, respectively. All the trainings are on a node with either 4 NVIDIA A40 or A100 GPUs. Except for Swin-L and ResNet-101, we use the Diagonalwise Refactorization [34] to implement the depthwise convolution in our method. For Swin-L and ResNet-101, we use the DW convolution [29].

B. Effect of Contextual Information

Objects in Cityscapes contain a wide variety of sizes and complex scenarios, which makes it challenging. The local

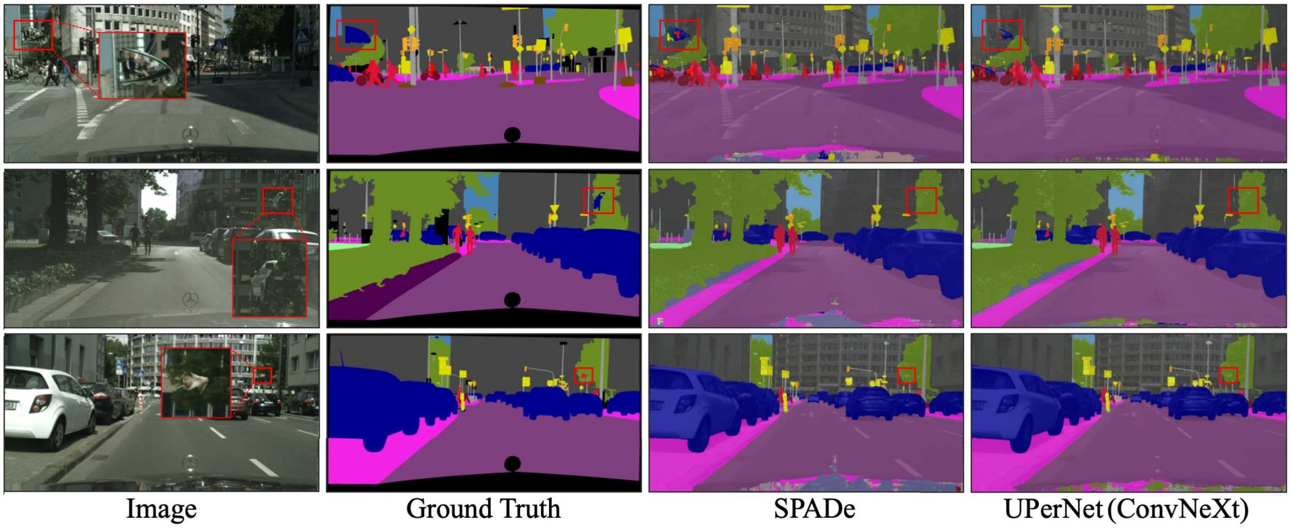


Fig. 6. Results of modified Cityscapes images using SPADe, UPerNet, and SegFormer. The red boxes show the objects randomly copied to a different region.

context pertains to the surrounding pixels within a small range. These pixels may belong to the same or different objects. The global context provides a more comprehensive view of the semantics present in a large scope within the image.

We modified images by copying an object to a random region. We hypothesize that even the hard-to-segment objects with weak levels of feature representation and small sizes can be correctly segmented with the right semantic context. On the other hand, if such objects are out of context, the model would segment them incorrectly. The modified images are processed by UPerNet and SPADe, both of which are implemented with the ConvNeXt-XL network.

Fig. 6 depicts our results. The top row shows that a car resembles a bike. In its original location, all three models consider the object as a bike due to the neighboring bikes. When the object is pasted to a different location, UPerNet segments it as mostly a building with parts of it as cars. Our model identifies the object as the car in the majority of cases; however, the overall prediction is inaccurate. The middle row shows a car that is copied from the lower left corner of the red box into a tree. In its original location, all three models predict it correctly as a car; however, due to the change in the contextual information, the same object is classified as a part of the tree by both models. In the bottom row, we copied a person from the lower right of the red box and pasted it into the tree. Even though the object is small, all the models can detect it correctly; however, for the modified image, the models find no correlation between the object and its surroundings and fail to predict correctly. Both models detect objects in their original context and miss them when the object is transferred out of context. This underscores the importance of semantic context, especially the global context for weakly presented objects.

To illustrate how context affects the performance of SPADe, we visualize the attention map of the model in Fig. 7, for two images in the center pixel. The attention map is also obtained for a modified image. We also provide a highlighted attention map that illustrates the most contributing context pixel among the pixels of the first attention map. In the first and third rows of Fig. 7, the identified objects have highly activated pixels in both attention and highlighted attention maps. The number of local context pixels that fall inside the object is much smaller than the global context pixels that surround the object. When the same object is moved to a different location in the second and fourth rows, we observe that the model fails to get activated not only inside the object but also globally. This is an indicator of the harmony and correlation between local and global contextual information.

C. Comparison With the State-of-the-Art Methods

We evaluate our SPADe by using several backbone networks and report in GFlops and mIoU adopted from MMSegmentation GitHub [35]. Table I compares the state-of-the-art methods using the Cityscapes. We report the total number of parameters of the model (Param.), the number of parameters of the backbone (B.Param.), and the number of parameters of the decoder (D. Param.). Additionally, we report the total GFlops for the entire model (GFlops), the total GFlops of the backbone (B. GFlops), and the total GFlops of the decoder (D. GFlops). Two groups of models, namely small and large, in terms of backbone size, are presented. Within each group, we evaluated SPADe with one or more backbone networks.

Table II reports the results of using the ADE20k. The methods are grouped into small, medium, and large sizes. If available, the mIoU is provided in single-scale (ss) and multi-scale (ms) testing modes. To perform multi-scale inference, we perform augmentation by flipping the original image and resizing the original image with a scaling factor of [0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0], giving a total of 8 augmented copies of the image. A label is predicted for each case, and the labels are merged through averaging. The best performance in each metric is highlighted with bold-face fonts, and the second best is underlined. The results are sorted according to the mIoU (ms) for the methods of the same backbone.

1) Performance Analysis: On the Cityscapes dataset, we obtained the highest ss and ms mIoU with ResNet-101 backbone, with an ms mIoU improvement of 1.48%. Our model

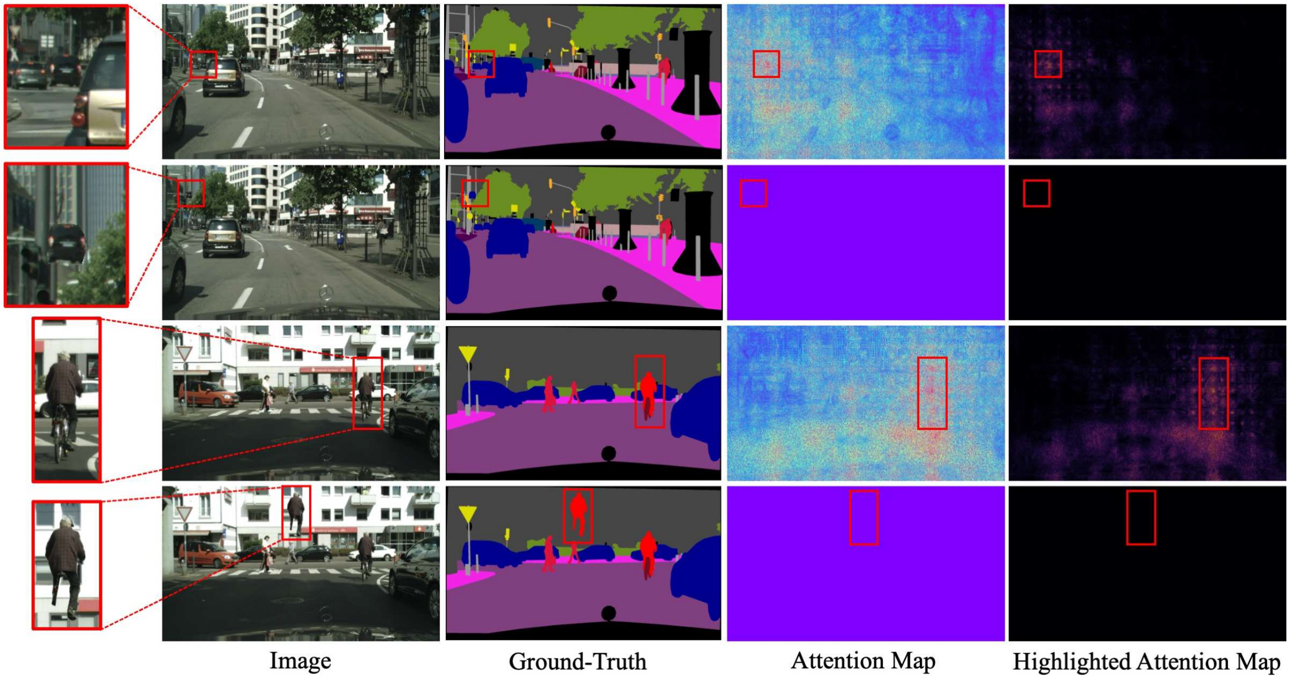


Fig. 7. GradCAM attention of SPADe over two images and the modified versions. The red boxes show the objects in the original and modified versions.

TABLE I

PERFORMANCE COMPARISON ON CITYSCAPES VAL.. THE GFLOPS IS CALCULATED BASED ON 1024 × 2048 IMAGES. THE SS AND MS DENOTE SINGLE- AND MULTI-SCALE TESTING. THE * DENOTES THAT THE BACKBONE IS DILATED RESNET-101

	Model	Backbone	Decoder	Para (M)↓	B. Para (M)↓	D. Para (M)↓	GFlops↓	B. GFlops↓	D. GFlops↓	mIoU(ss/ms)↑
Small	FCN	ResNet-101*	FCN	69	42	27	2203	1432	771	- /76.6
	EncNet	ResNet-101*	EncNet	52	42	10	1752	1432	320	76.1/77.0
	FPN	ResNet-101	FPN	48	42	6	520	337	183	75.8/77.4
	PSPNet	ResNet-101	PPM	66	42	24	**376**	337	**39**	- /78.5
	SenFormer	ResNet-101	SenFormer	163	42	121	1473	337	1136	80.3/ -
	CCNet	ResNet-101*	CCNet	66	42	24	2236	1432	804	79.5/80.7
	DeeplabV3+	ResNet-101*	ASPP	60	42	18	2035	1432	603	- /80.9
	UPerNet	ResNet-101	UPerNet	83	42	41	2052	337	1715	- /81.5
	Ours	ResNet-101	SPADe	**46**	42	**4**	652	337	315	**80.7/82.1**
Large	GSS-FT-W	Swin-L	-	-	195	-	-	-	-	- /80.0
	DiversePatch	Swin-L	DiversePatch	234	195	39	3190	2616	574	82.7/83.6
	SenFormer	Swin-L	SenFormer	233	195	38	4368	2616	1752	82.8/84.0
	Ours	Swin-L	SPADe	**199**	195	**4**	**1914**	1752	**162**	**83.3/84.2**
	ConvNeXt	ConvNeXt-XL	UPerNet	391	348	43	4264	2552	1712	**83.8**/84.3
	Ours	ConvNeXt-XL	SPADe	**352**	348	**4**	**2864**	2552	**312**	83.5/**84.5**
	InternImage	InternImage-XL	UPerNet	368	329	39	4022	2300	1722	83.6/84.3
	Ours	InternImage-XL	SPADe	**333**	329	**4**	**2616**	2300	**316**	**84.0/84.8**

also shows the best ss mIoU with a Swin-Large backbone, with a 0.6% improvement margin compared to the second-best. With ConvNeXt-XL, we performed the best with ms mIoU, showing superiority over UPerNet with the same backbone with 0.2% mIoU. With the InternImage-XL [36] as the backbone, SPADe obtains a higher performance than the case with the UPerNet decoder in both ss and ms evaluations of mIoU. The SPADe with InternImage-XL is also the highest-performing model among all the evaluated models in Table I.

Table II compare the state-of-the-art methods using ADE20k. Our SPADe decoder achieves the highest ss mIoU with both ResNet-101 and ResNet-101-D8 with a 0.2% and 1.8% improvement over the second-best model, of the same backbone, respectively. Our performance in terms of mIoU (ms) is also very competitive. To show the compatibility of the SPADe with various architectures, we also test SPADe with DaViT-B and Swin-B. In the case of Swin-B, our model obtained the highest ss mIoU of 51.2%, among all the medium-sized models, and a competitive ms mIoU of 51.8%. DaViT performed as well as UPerNet with the same backbone. Our model with Swin-L gets the best ms mIoU and the second-best ss mIoU with 0.5% improvement in ms over KNet. Also, we got the best ss mIoU of 53.8 using ConvNeXt-XL, outperforming UPerNet and the same ms as UPerNet.

Fig. 8 depicts a comparison of SPADe with UPerNet. On the top row, the UPerNet misses a significant part of the sidewalk. In the second case, both methods perform equally well; however, in the identified red box, UPerNet misclassifies the building, while ours accurately classifies the pixels. In contrast to the state-of-the-art, our proposed SPADe network demonstrates a superior performance in terms of mIoU.

TABLE II

COMPARISON ON ADE20K VAL.. GFLOPS IS CALCULATED USING 512 × 512 IMAGES OR 640 × 640 IMAGES †. THE SS AND MS DENOTE SINGLE- AND MULTI-SCALE TESTING. *: BACKBONE IS DILATED RESNET-101

	Model	Backbone	Decoder	Para (M)↓	B. Para (M)↓	D. Para (M)↓	GFlops↓	B. GFlops↓	D. GFlops↓	mIoU(ss/ms)↑
Small	FPN	ResNet-101	FPN	48	42	6	**65**	42	**23**	39.4/40.7
	DeepLabV3+	ResNet-101	ASPP	63	42	21	255	42	213	- /44.1
	PSPNet	ResNet-101	PPM	68	42	26	256	42	214	- /44.1
	EncNet	ResNet-101*	EncNet	52	42	10	**219**	179	**40**	42.6/44.0
	ResNet	ResNet-101	UPerNet	83	42	41	258	42	215	43.8/**44.9**
	CCNet	ResNet-101*	CCNet	66	42	24	279	179	100	43.7/**45.0**
	Ours	ResNet-101*	SPADe	**46**	42	**4**	224	179	45	**44.5**/44.9
Medium	DaViT	DaViT-B	UPerNet	121	87	34	293	84	209	**49.4**
	Ours	DaViT-B	SPADe	**91**	87	**4**	**124**	84	**40**	**49.4**
	Shuffle Transformer	Shuffle-S	UPerNet	81	-	-	261	-	-	48.4/49.6
	FocalNet	Focal-B	UPerNet	126	90*	-	1354	-	-	49.0/50.5
	QFormer-B	QFormer-B$_h$	UPerNet	123	90*	-	-	-	-	49.5/50.6
	UniFormer	UniFormer-B$_{h_{32}}$	UPerNet	80	50*	-	276	-	-	49.5/50.7
	CEDNet-NeXt-B	ConvNeXt-B	CED	123	-	-	296	-	-	49.9/51.0
	Swin Transformer	Swin-B	UPerNet	120	87	33	306	94	212	50.8/**52.4**
	Ours	Swin-B	SPADe	**91**	87	**4**	**133**	94	**40**	**51.2**/51.8
Large	KNet†	Swin-L	KNet	245	195	50	659	291	368	52.2/53.3
	Swin Transformer†	Swin-L	UPerNet	232	195	37	662	327	334	- /53.5
	SenFormer†	Swin-L	SenFormer	233	195	38	546	327	219	53.1/ -
	Ours	Swin-L	SPADe	**199**	195	**4**	**246**	**206**	**40**	52.6/**53.8**
	ConvNeXt†	ConvNeXt-XL	UPerNet	391	348	43	834	498	336	53.6/**54.0**
	Ours†	ConvNeXt-XL	SPADe	**352**	348	**4**	**560**	498	**62**	**53.8/54.0**

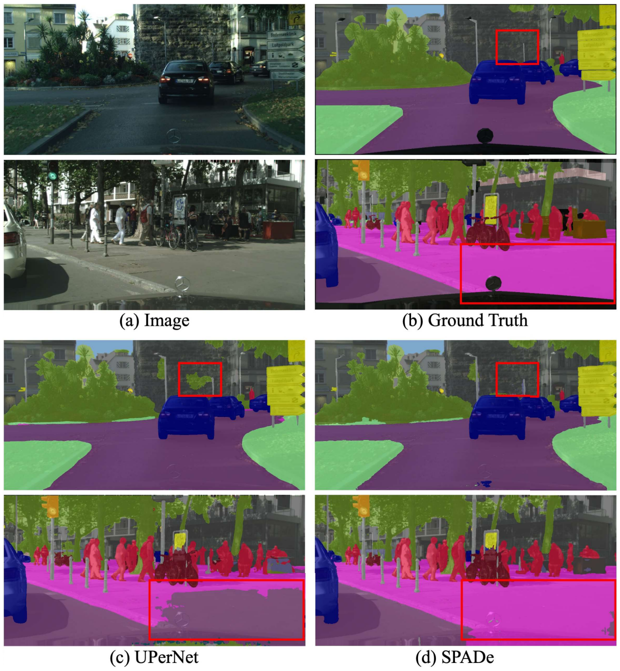

(a) Image
(b) Ground Truth
(c) UPerNet
(d) SPADe

Fig. 8. Comparison of SPADe and UPerNet on two images of Cityscape val.

2) Efficiency Analysis: Tables I and II also report the efficiency in terms of parameter size and GFlops. Testing with Cityscapes, our SPADe consistently achieves the lowest parameter size of 4M for the decoder. SPADe with ResNet-101 has a low GFlops profile while obtaining the best mIoU. With Swin-L it improves the decoder GFlops by 71.8% while for ConvNeXt-XL this gain reaches 81.78% compared to the second-best. When SPADe is used, the model size is reduced from 14.59% and 9.97% for Swin-L and ConvNeXt-XL compared to the second-best. Using the InternImage backbone, SPADe obtained a significant reduction of decoder parameters and GFlops at 89.74% and 81.64% in contrast to UPerNet.

In the experiments using ADE20k, the best-performing model in ms mIoU with ResNet-101 backbone has a GFlops of 215, while SPADe obtains a similar performance with 90.24% reduction in decoder size and 81.39% lower decoder GFlops. Compared to CCNet, the top-performing method with ResNet-101-D8 backbone, SPADe has lowered the decoder computation by a margin of 55.0% while its size is 16.66% of CCNet. We also note that the best-performing decoders in terms of GFlops, with ResNet-101 and ResNet-101-D8, sacrifice the mIoU to gain lower computational cost. With the DaViT our model has 30M fewer parameters overall and only 19.13% decoder GFlops with the same mIoU.

For the large backbones, such as Swin-L and ConvNeXt-XL, the same trend is observed as both decoder size and GFlops have best-performing stats, with 89.19% and 81.73% for the decoder size and GFlops improvement of SPADe over the second-best for Swin-L, as well as 90.69% and 81.54% of decoder size and GFlops improvement with ConvNeXt-XL. In conclusion, our SPADe model achieves a very competitive efficiency, boasting the smallest parameter size and GFlops on Cityscapes and ADE20k benchmarks in many cases.

3) Performance and Efficiency Trade-off: In practice, we often aim at balancing performance and efficiency. Fig. 9 illustrates a diagram of the computational cost (GFlops), the performance (mIoU), and the size of decoders. The colors of the circles categorize them in terms of backbone, while the size of the decoder for each model is visualized using the size of the solid circle representing the model.

A model with high GFlops and low mIoU falls closer to the upper left corner of the space, while a competitive model with high mIoU and low GFlops tends to lie near the lower right corner of the space. Models such as FPN lie in the lower left corner, showing low computational cost, decoder size, and low mIoU for both datasets. On the contrary, UPerNet lies close to the upper right corner, demonstrating high performance, but with a high cost and decoder size. Our SPADe networks are at the lower range of the GFlops and, depending on the backbone networks used, achieve very competitive mIoU. That is, a smaller decoder size, competitive mIoU, and lower cost are achieved compared to the models using the same backbone,

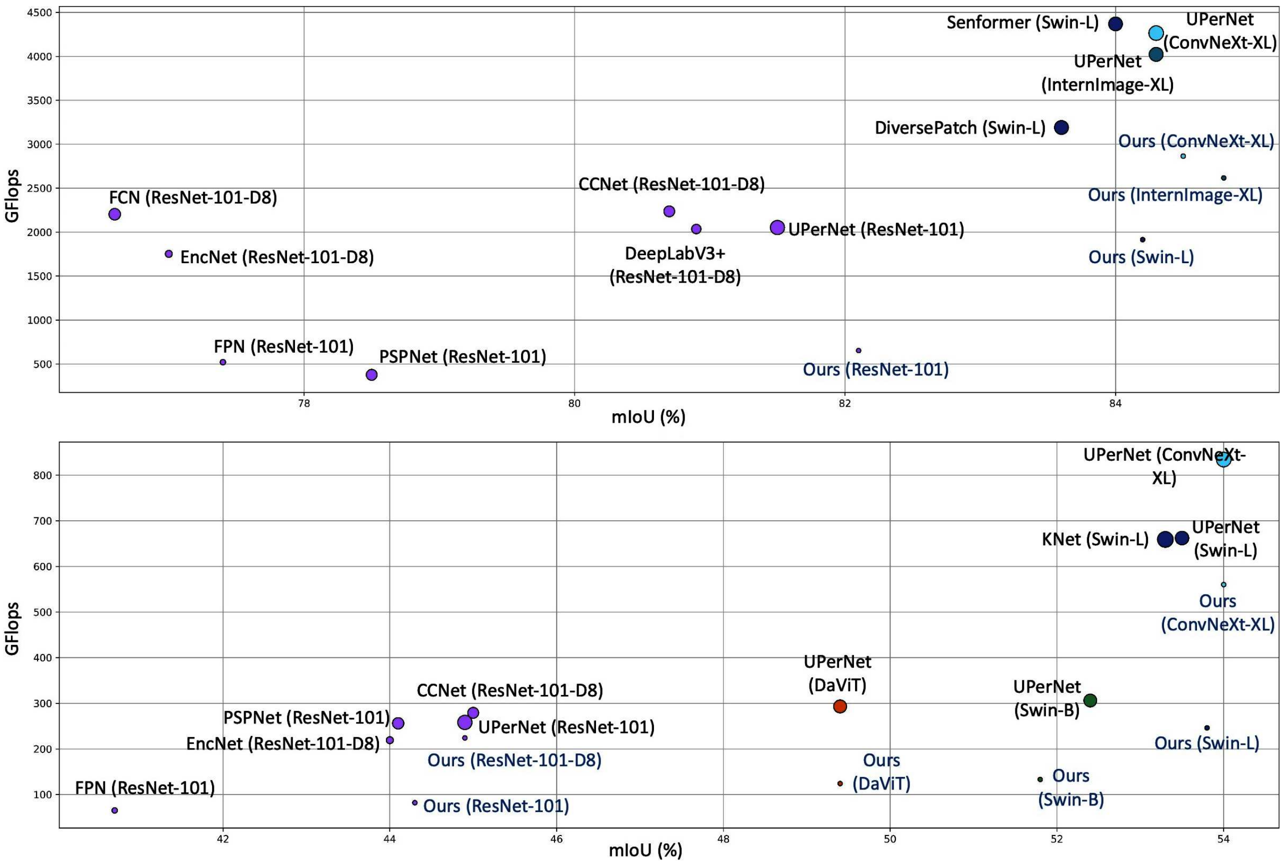


Fig. 9. GFlops vs. mIoU of the compared methods.

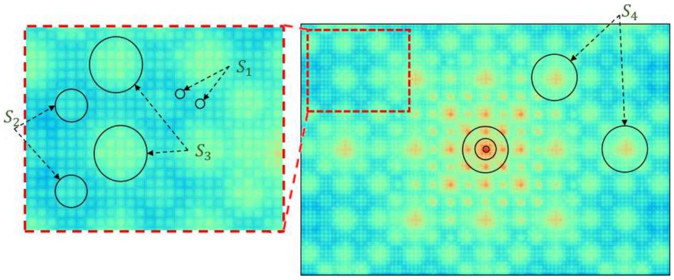


Fig. 10. The average receptive field of the SPADe.

letting it land closer to the lower right region of the space. This is evident that the proposed method achieves a superior balance of performance and cost.

D. Effect of Decoder on the Receptive Field

Fig. 10 illustrates the average receptive field of the SPADe for 100 images when predicting the label of the center pixel. Our model captures the contextual information in four groups of circles according to the radius size. Hot colors represent greater gradients, which indicate the most attention and the regions that contribute in a higher magnitude towards the prediction. The S_1 category groups pixels in a small neighborhood, with a relatively lower level of importance, according to their color. Given that the network keeps processing the features, the receptive field increases, leading to the formation of large pixel groups such as S_2, S_3, and S_4.

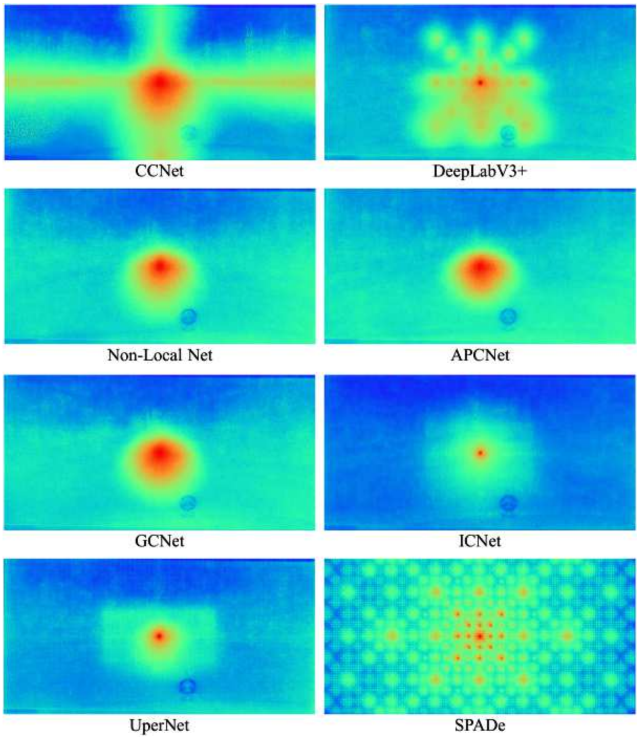
Additionally, the overlap of pixel groups, e.g., S_1 and S_2, increases the attention to the pixel in their intersection, which leads to assigning greater gradients and consequently warmer colors to them. A sample of our overlapping pattern is highlighted with a double circle in the middle of the image, where S_1, S_2, and S_3 patterns are interlaced.

To illustrate the difference in the overall receptive field of a model, we implement the network using ResNet-101 as the backbone network for the encoder, followed by different decoders.

The receptive fields of the networks are shown in Fig. 11. In contrast to ICNet, UPerNet, GCNet, APCNet, and Non-Local Net, which mainly attend to a small neighborhood around the center pixel, our model attends to most parts of the image. It covers an appropriate contextual scope concerning the objects. CCNet focuses more on a narrow vertical and horizontal region near the center of the pixel of interest. Although DeepLab-V3+ shows a similar receptive field pattern, repetition of multi-path convolution and folding enables our decoder to adapt the receptive field for a larger coverage. Hence, our SPADe network ensures the capture of local features and provides extensive contextual information with a large receptive field.

E. Evaluation on the Repetition of Processing Block

SPADe uses several repetitions (denoted with L) of SPA. The choice of L has an impact on the receptive field of the network. By using more parallel convolutions of different



CCNet DeepLabV3+

Non-Local Net APCNet

GCNet ICNet

UperNet SPADe

Fig. 11. Average receptive field of decoders for 100 images of Cityscapes val. The backbone is ResNet-101. SPADe considers a larger context.

TABLE III

EFFECT OF L USING CITYSCAPES VAL. USE SINGLE-SCALE TESTING

L	Param.(M)	GFlops	mIoU
1	**44.16**	**229**	79.64
2	45.24	278	80.01
3	46.32	327	**80.65**

Atrous rates, the network adjusts the contribution of the spatial context. Table III reports the results using different values for L. When L is increased, the mIoU of SPADe also increases. Comparing the cases when $L = 1$ to $L = 2$, the model obtains 0.5% higher mIoU while adding only 1.08 M additional parameters. An $L = 3$ improves the performance further by another 0.5% mIoU. In each increment of L value, we observe only an increase of 49 GFlops. It is clear that as we increase the number of repetitions of the SPA module, the mIoU improves. However, more SPA modules used in the SPADe network require a greater memory space and computation. In the rest of our experiments, we set the number of SPA repetitions to 3 in our SPADe network.

Table IV gives a detailed view of the class-wise performance of our method when different L values are used. The average mIoU of all classes is also reported at the end of the table. A larger value for L improves the model performance on the majority of classes, which includes the classes comprising small objects such as traffic lights and persons, and the ones primarily composed of large objects such as roads and cars. For the classes with small objects, the large receptive field helps SPADe to recognize the object's surroundings and make better predictions. For bigger objects, due to the size, not only is contextual information around the object important, but keeping high prediction consistency across all pixels of the object is a challenge. This problem is alleviated by larger Ls, leading to an overall improved performance. Overall, a deeper SPADe network exhibits greater performance in accurately segmenting and labeling the objects, such as the increased mIoU by 0.71%, 1.99%, 3.12%, and 4.32% for sidewalk, wall, terrain, and train classes, respectively. Such improvements are achieved at the expense of a small cost of 49 GFlops.

TABLE IV

CLASS-WISE MIOU OF SPADE WITH RESNET-101 ON CITYSCAPES VAL

Model	road	swalk	build.	wall	fence	pole
L=1	98.14	85.26	92.90	50.42	62.74	68.27
L=2	98.24	85.69	**93.09**	53.84	62.30	68.90
L=3	**98.32**	**86.40**	93.03	**55.83**	**62.99**	**69.23**
tlight	sign	veg.	terrain	sky	person	rider
73.49	81.37	92.62	66.34	**95.26**	84.23	**67.69**
73.85	**82.07**	92.69	64.07	95.15	84.11	67.11
74.41	81.80	**92.77**	**67.19**	95.17	**84.33**	66.79
car	truck	bus	train	mbike	bike	mIoU
95.69	82.49	88.74	77.27	70.86	79.46	79.64
95.72	**86.75**	90.05	76.77	**71.83**	**79.59**	80.01
95.82	84.91	**90.44**	**81.09**	71.61	79.54	**80.65**

TABLE V

THE MIOUS USING CITYSCAPES VAL.. THE CHECK MARKS INDICATE THE INCLUSION OF THE COMPONENT

Case	Folding	Multi-Path DW	mIoU(ss)
1	✗	1, 8, 18	79.66
2	✓	1	76.51
3	✓	1, 8	80.72
4	✓	1, 8, 18	80.61
5	✓	1, 18, 36	**80.72**

F. Multi-Path Depthwise Convolutions

We conduct experiments using different dilation rates with and without folding to evaluate folding and multi-path feature processing in SPA. Five cases are reported in Table V. In the first case, we use no 4-neighbor folding and only use the three parallel convolutions. The rest of the cases include folding but have a different number of convolutions.

Cases 1 and 4 differ in the inclusion of folding. Without folding, the mIoU is 79.66%, whereas the inclusion of folding improves the performance to 80.61%, which is an increase of 1.19%. When folding is used with a 1 × 1 convolution, the performance degrades dramatically by 5.5%, comparing cases 2 and 5. Involving two or three parallel convolutions benefits the performance to a similar extent. However, we use the configuration in case 5 as a larger atrous range benefits the model when processing the images as a whole rather than using a sliding window operation. Both folding and multi-path convolutions are essential parts of SPA to achieve improved performance compared to the baseline. Folding benefits SPA by providing contextual information in a locality, while multi-path depthwise convolution leverages long-range dependencies of pixels to improve the performance of the SPADe.

V. Conclusion

This paper presents a SPADe network with a decoder module SPA for semantic segmentation. The SPADe network uses a multi-contextual feature fusion approach, combining feature maps from different spatial contexts to produce high-quality segmentation results with low computation. SPA decoder increases the receptive field, capturing a larger contextual scope, with a sparse convolution kernel to balance the accuracy and efficiency. Our results demonstrate that SPADe with InternImage-XL backbone achieved performance using Cityscapes, with 34.95% fewer Flops and only 9.75% of the parameters compared to UPerNet. Using ADE20K, our method with ConvNeXt-XL achieved comparable performance to the state-of-the-art while reducing Flops by 32.85% compared to UPerNet. Additionally, when using the ResNet-101 backbone on Cityscapes, our decoder achieves the second-best in Flops. Increasing the number of levels of SPA improved the mIoU by 1%. Additionally, incorporating folding and dilated depthwise convolution helped SPADe achieve a 1.06% mIoU gain. Notably, SPADe offers the largest effective receptive field when compared to state-of-the-art using the same backbone.

Despite the enhancement in semantic segmentation accuracy and a reduction in both model size and computational efficiency, there is still room for improvement, particularly for real-world applications where computational power is limited and sub-second response is expected. Our future study will explore hierarchical SPAs with sparse connections to reduce the computation costs. In addition, a dual-branch network structure could be investigated to alleviate the demand for large video memories and, hence, enable the deployment in edge devices.

References

[1] Z. Wei et al., "Multi-step regression network with attention fusion for airport delay prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 7093–7105, Jul. 2024.

[2] X. Yuan, J. Shi, and L. Gu, "A review of deep learning methods for semantic segmentation of remote sensing imagery," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114417.

[3] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[4] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 418–434.

[5] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[6] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 108–126.

[7] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2021, pp. 17864–17875.

[8] C. Zhuang, X. Yuan, and W. Wang, "Boundary enhanced network for improved semantic segmentation," in *Proc. Int. Conf. Urban Intell. Appl.*, Aug. 2020, pp. 172–184.

[9] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4353–4361.

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

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[12] A. Dosovitskiy et al., "An image is worth 16*x*16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–21.

[13] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. NIPS*, 2021, pp. 15908–15919.

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

[15] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[17] H. Wang, P. Cao, J. Wang, and O. R. Zaiane, "UCTransNet: Rethinking the skip connections in U-Net from a channel-wise perspective with transformer," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, 2022, pp. 2441–2449.

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

[19] S. Liu and D. Huang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 385–400.

[20] B. A. Wandell and J. Winawer, "Computational neuroimaging and population receptive fields," *Trends Cognit. Sci.*, vol. 19, no. 6, pp. 349–357, Jun. 2015.

[21] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[22] H. Cao et al., "Swin-UNet: UNet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 205–218.

[23] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[24] E. Xie et al., "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Sys. (NIPS)*, vol. 34, Dec. 2021, pp. 12077–12090.

[25] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M. Cheng, and S. Hu, "SegNeXt: Rethinking convolutional attention design for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1140–1156.

[26] Z. Geng, M.-H. Guo, H. Chen, X. Li, K. Wei, and Z. Lin, "Is attention better than matrix decomposition?," 2021, *arXiv:2109.04553*.

[27] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-Net: Towards unified image segmentation," in *Proc. NIPS*, 2021, pp. 10326–10338.

[28] K. Sofiiuk, O. Barinova, and A. Konushin, "AdaptIS: Adaptive instance selection network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7355–7363.

[29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[30] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[31] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

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

[33] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 633–641.

[34] Z. Qin. (2018). *Diagonalwise Refactorization: An Efficient Training Method for Depthwise Convolutions*. [Online]. Available: https://github.com/qinzheng93/diagonalwise-refactorization-pytorch

[35] (2020). *MMSegmentation: Openmmlab Semantic Segmentation Toolbox and Benchmark*. [Online]. Available: https://github.com/open-mmlab/mmsegmentation

[36] W. Wang et al., "InternImage: Exploring large-scale vision foundation models with deformable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14408–14419.