Towards Efficient Scene Understanding via Squeeze Reasoning

Xiangtai Li, *Graduate Student Member, IEEE*, Xia Li, Ansheng You, Li Zhang, Guangliang Cheng, Kuiyuan Yang, Yunhai Tong, and Zhouchen Lin, *Fellow, IEEE*

***Abstract*— Graph-based convolutional model such as non-local block has shown to be effective for strengthening the context modeling ability in convolutional neural networks (CNNs). However, its pixel-wise computational overhead is prohibitive which renders it unsuitable for high resolution imagery. In this paper, we explore the efficiency of context graph reasoning and propose a novel framework called Squeeze Reasoning. Instead of propagating information on the spatial map, we first learn to squeeze the input feature into a channel-wise global vector and perform reasoning within the single vector where the computation cost can be significantly reduced. Specifically, we build the node graph in the vector where each node represents an abstract semantic concept. The refined feature within the same semantic category results to be consistent, which is thus beneficial for downstream tasks. We show that our approach can be modularized as an end-to-end trained block and can be easily plugged into existing networks. Despite its simplicity and being lightweight, the proposed strategy allows us to establish the considerable results on different semantic segmentation datasets and shows significant improvements with respect to strong baselines on various other scene understanding tasks including object detection, instance segmentation and panoptic segmentation. Code is available at https://github.com/lxtGH/SFSegNets.**

***Index Terms*— Channel attention, efficient global context modeling, scene understanding.**

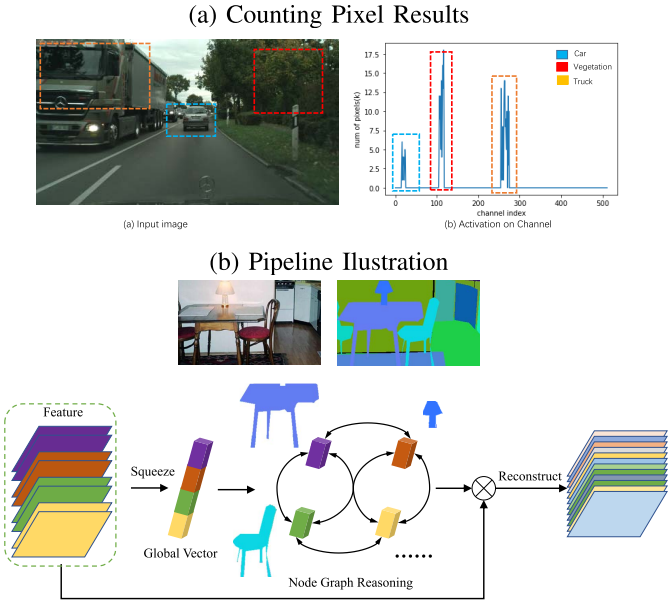

Fig. 1. (a) Toy Experiment results by counting pixels given specific classes using trained model. (b) Illustration of our proposed module for semantic segmentation task. Best view zoom in.

I. Introduction

CONVOLUTIONAL neural networks have proven to be effective and useful to learn visual representations in an end-to-end fashion with a certain objective task such as, semantic segmentation [1], image classification [2], object detection [3], instance segmentation [4] and panoptic segmentation [5]. However, the effective receptive field [6] of CNNs grows slowly if we simply stack local convolutional layers. Thus, the limited receptive field prevents the model from taking all the contextual information into account and thus renders the model insufficiently covering all the regions of interest.

A broad range of prior research has investigated network architecture designs to increase the receptive field-of-view such as self-attention/Non-local [7], channel attention [8], and graph convolution network (GCN) [9]. Although they have been shown to be effective in strengthening the representations produced by CNNs, modeling the inter-dependencies in a high-dimensional space prevents them from fully exploiting the sparse property required by the final classifier. Furthermore, they suffer from the prohibitively expensive computation overhead during training and inference, *e.g.*, the large affinity matrix at each spatial position [8] or channel position [9], which renders these methods unsuitable for high-resolution inputs. Although recent methods reduce such cost by involving fewer pixels [10] or selecting representative nodes [11],

Manuscript received January 3, 2021; revised June 18, 2021; accepted July 17, 2021. Date of publication July 30, 2021; date of current version August 11, 2021. This work was supported by the National Key Research and Development Program of China under Grant 2020YFB2103402. The work of Zhouchen Lin was supported in part by NSF, China, under Grant 61625301 and Grant 61731018, and in part by the Major Scientific Research Project of Zhejiang Lab under Grant 2019KB0AC01 and Grant 2019KB0AB02. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ming-Ming Cheng. *(Corresponding author: Yunhai Tong.)*

Xiangtai Li, Xia Li, Ansheng You, Yunhai Tong, and Zhouchen Lin are with the Key Laboratory of Machine Perception, School of EECS, Peking University, Beijing 100871, China (e-mail: lxtpku@pku.edu.cn; yhtong@pku.edu.cn).

Li Zhang is with the School of Data Science, Fudan University, Shanghai 200433, China (e-mail: lizhangfd@fudan.edu.cn).

Guangliang Cheng is with SenseTime, Beijing 100036, China (e-mail: guangliangcheng2014@gmail.com).

Kuiyuan Yang is with DeepMotion, Beijing 100080, China (e-mail: kuiyuanyang@deepmotion.ai).

Digital Object Identifier 10.1109/TIP.2021.3099369

their computation is still huge given high resolution image inputs.

Could we find another way to eliminate the limitation of high-cost spatial information while capturing global context information? We first carry out toy experiments using a pretrained Deeplabv3+ model [12]. We count the pixels on the final normalized feature (512 dimensions before classification) given ground truth masks whose activation values are beyond 0.8. As shown in Fig 1(a), we find different classes lie in different groups along channels sparsely. We only show three classes for simplicity. This motivates us to build an information propagation module on channel solely where each group represents one specific semantic class while the cost of spatial resolution can be avoided. Inspired by SE-networks [8], we first squeeze the feature into a compact global vector and then perform reasoning operation on such compact vector. Benefit from squeezing, the computation cost can be significantly reduced compared with previous works. The schematic illustration of our proposed method is shown in Fig 1(b). Compared with previous work modeling pair-wised affinity maps over the input pixels [10], [13]–[17], our method is totally different by building node graph conditionally on the whole image statistics and also results in efficient inference. After reasoning, the most representative channels of input features can be selected and enhanced which solves the inconsistent segmentation results on large objects.

Our framework mainly contains three steps. First, we perform the node squeezing operation to obtain the global vector. This can be done by simply a global average pooling or using Hadamard product pooling to capture the second-order statistics. Then we carry out node reasoning by dividing such vector into different groups, and the inter-dependencies can be diffused through the reasoning process. Finally, we reconstruct the original feature map by multiplying the reasoned vector with the original input. Our approach can serve as a lightweight module and can be easily plugged into existing networks. Compared to Non-local [7] or graph convolution network [9], which model the global relationship on feature spatial or channel dimension, our approach instead models the inter-dependencies on the squeezed global vector space, and notably, each node consists of a group of atom/channel. Therefore, our method uses substantially fewer floating-point operations and fewer parameters and memory costs. Moreover, our method achieves the best speed and accuracy trade-off on the Cityscapes test set, which shows its practical usage. In particular, our method achieves 77.5% mIoU on Cityscapes test set while running at 65 FPS on single 1080-TI device.

Besides its efficiency, our method is also verified to be effective on long-range context modeling. As shown in Tab. I, our method results in a significant gain on dilated FCN with ResNet50 backbone. In particular, the segmentation results of larger objects in the scene can be improved significantly by over 10% mIoU per category. Moreover, our methods only require a few extra computation cost(1.5% relative increase over the baseline models). Fig.2 gives the visualization results on the corresponding models in Tab. I. As shown in that figure, the inconsistent noise on the train and truck can be removed by our proposed SR module.

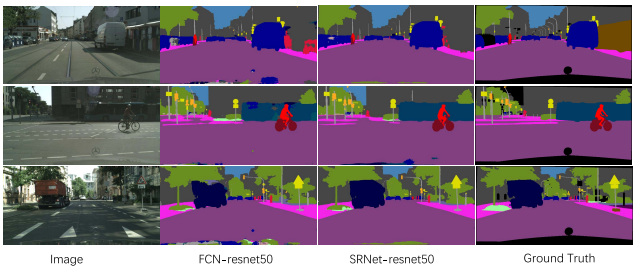

Fig. 2. Visualization results on Cityscapes validation set. Best viewed zoom in.

Moreover, our proposed method is also verified to other tasks including instance level taks on instance segmentation and panoptic segmentation on COCO datasets [18], image classification on ImageNet [19]. Our method can obtain consistent improvements over mask-rcnn baseline models [4] with negligible cost and considerable results on ImageNet classification with SE-net [8]. More detailed information can be found in the experiment parts. Those experiments further demonstrate the generality of our proposed approach.

The contributions of this work are as follows:

(i) We propose a novel squeeze reasoning framework for highly efficient deep feature representation learning for scene understanding tasks.

(ii) An efficient node graph reasoning is introduced to model the inter-dependencies between abstract semantic nodes. This enables our method to serve as a lightweight and effective module and can be easily deployed in existing networks.

(iii) Extensive experiments demonstrate that the proposed approach can establish new state-of-the-arts on four major semantic segmentation benchmark datasets including Cityscapes [20], Pascal Context [21], ADE20K [22] and Camvid [23] while keeping efficiency, and show consistent improvement with respect to strong baselines on several scene understanding tasks with negligible cost. More experiments on different tasks through datasets [18], [19] including instance segmentation and image classification prove the generality of proposed SR module.

II. Related Work

In this section, we will review the related work in two aspects: global context aggregation and semantic segmentation.

A. Global Context Aggregation

Beyond the standard convolutional operator used for short-range modeling, many long-range operators are proposed to aggregate information from large image regions, even the whole image. Global Average Pooling (GAP) [2], which bridges local feature maps and global classifiers, is widely used for long-range modeling. In Squeeze-and-Excitation network [8], GAP is used in more intermediate layers for coupling global information and local information more thoroughly. In Pyramid Pooling Module(PPM) [24], a pyramid of average pooling operators is used to harvest features. In addition to

TABLE I

DETAILED RESULTS ON CITYSCAPES VALIDATION SET. IN PARTICULAR, OUR METHOD CAN OBTAIN A LARGE IMPROVEMENT ON LARGE OBJECTS IN THE SCENE INCLUDING TRAIN(24.1%), TRUCK(18.1 %) AND BUS(17.4 %)

Method	road	swalk	build	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU	GFlops
dilated FCN	97.9	83.8	92.3	49.2	58.7	65.3	72.4	79.7	92.1	61.3	94.5	82.5	62.5	93.9	**66.0**	**71.1**	**51.0**	66.0	77.8	74.8	241.05G
+SR Head	98.3	85.9	93.2	62.2	62.4	66.4	73.2	81.1	92.8	66.3	94.8	83.2	65.0	95.7	**84.1**	**89.6**	**75.1**	67.7	78.8	79.9(+5.1)	+3.64G

first-order statistics captured by GAP, bilinear pooling [25] extracts image-level second-order statistics as complementary of convolutional features. Besides pooling-based operators, generalized convolutional operators [26] are also used for long-range modeling. Astrous convolution enlarges kernels by inserting zeros in between [27], which is further used by stacking kernels with multiple astrous rates pyramidally [28]–[30] or densely [31]. Deformable convolution [32], [33] generalizes atrous convolution by learning the offsets for convolution sampling locations. Global average pooled features are concatenated into existing feature maps in [34]. In PSPNet [24], average pooled features of multiple window sizes including global average pooling are upsampled to the same size and concatenated together to enrich global information. The DeepLab series of papers [28]–[30] propose atrous or dilated convolutions and atrous spatial pyramid pooling (ASPP) to increase the effective receptive field. DenseASPP [31] improves on [29] by densely connecting convolutional layers with different dilation rates to further increase the receptive field of network. In addition to concatenating global information into feature maps, multiplying global information into feature maps also shows better performance [15], [35]–[37]. In particular, EncNet [15] and SqueezeSeg [38] use attention along the channel dimension of the convolutional feature map to account for global context such as the co-occurrences of different classes in the scene. CBAM [35] explores channel and spatial attention in a cascade way to learn task specific representation.

Recently, advanced global information modeling approaches initiated from non-local network [7] are showing promising results on scene understanding tasks. In contrast to convolutional operator where information is aggregated locally defined by local filters, non-local operators aggregate information from the whole image based on an affinity matrix calculated among all positions around the image. Using non-local operator, impressive results are achieved in OCNet [13],CoCurNet [39], DANet [40], A2Net [41], CCNet [10] and Compact Generalized Non-Local Net [36]. OCNet [13] uses non-local bolocks to learn pixel-wise relationship while CoCurNet [39] adds extra global average pooling path to learn whole scene statistic. DANet [40] explores orthogonal relationships in both channel and spatial dimension using non-local operator. CCNet [10] models the long range dependencies by considering its surrounding pixels on the criss-cross path through a recurrent way to save both computation and memory cost. Compact Generalized non-local Net [36] considers channel information into affinity matrix. Another similar work to model the pixel-wised relationship is PSANet [42]. It captures pixel-to-pixel relations using an attention module that takes the relative location of each pixel into account. EMANet [16] proposes to adopt expectation-maximization algorithm [43] for the self-attention mechanism.

Another way to get global representation is using graph convolutional networks, and do reasoning in a non-euclidean space [44]–[46] where messages are passing between each node before projection back to each position. Glore [9] projects the feature map into interaction space using learned projection matrix and does graph convolution on projected fully connected graph. BeyondGrids [44] learns to cluster different graph nodes and does graph convolution in parallel. SPGNet [45] performed spatial pyramid graph reasoning while DGMNet [46] proposed dynamic graph reasoning framework for more efficient learning. In our work, a global vector squeezed from the whole image is organized as a small graph for reasoning, where each node contains rich global information. Thus reasoning is carried on a high level, which is more efficient and robust to noises than previous methods.

B. Semantic Segmentation

Recent years have seen a lot of work on semantic segmentation using deep neural network. FCN [1] removes global information aggregation layers such as global average pooling layer and fully-connected layers for semantic segmentation. Later, FCN-based methods dominate the area of image semantic segmentation. We review related methods in two different settings: non-real-time methods for better segmentation results and real-time models for fast inference. The work [27] removed the last two downsample layers to obtain dense prediction and utilized dilated convolutions to enlarge the receptive field. Meanwhile, both SAC [47] and DCN [32] improved the standard convolutional operator to handle the deformation and various scales of objects, which also enlarge the receptive fields of CNN operator. Several works [48]–[51] adopted encoder-decoder structures that fuses the information in low-level and high-level layers to make dense prediction results. In particular, following such architecture design, GFFnet [52], CCLNet [53] and G-SCNN [54] use gates for feature fusing to avoid noise and feature redundancy. CRF-RNN [55] used graph model such CRF, MRF for semantic segmentation. AAF [56] used adversarial learning to capture and match the semantic relations between neighboring pixels in the label space. DenseDecoder [57] built multiple long-range skip connections on cascaded architecture. DPC [58] and auto-deeplab [59] utilized architecture search techniques to build multi-scale architectures for semantic segmentation. Besides, there are also several works aiming for real time application. ICNet [60], BiSegNet [61] and SFNet [62] were designed for real-time semantic segmentation by fusing multi scale inputs or feature pyramids. DFANet [63] utilizes a light-weight

backbone to speed up its network and proposes a cross-level feature aggregation to boost accuracy, while SwiftNet [64] uses lateral connections as the cost-effective solution to restore the prediction resolution while maintaining the speed. There are also specially designed video semantic segmentation works for boosting accuracy [65], [66] and saving inference time [67], [68]. Our proposed module can work in both real-time setting to obtain the best speed and accuracy trade-off due to its efficiency or non-real-time setting to achieve the better consistent segmentation results.

III. Method

In this section, we first review related works [7], [8], [69] as preliminary knowledge. Then detailed description and formulation of our SR module are introduced. Finally, we elaborate on how to apply it to several different computer vision tasks.

A. Preliminaries

1) Graph Convolution: Assume an input matrix $\mathbf{X} \in \mathbb{R}^{D\times N}$, where D is the feature dimension and $N = H \times W$ is the number of locations defined on regular grid coordinates $\mathbf{\Omega} = \{1, \ldots, H\} \times \{1, \ldots, W\}$. In standard convolution, information is only exchanged among positions in a small neighborhood defined by the filter size (typically 3×3). In order to create a large receptive field and capture long-range dependencies, one needs to stack numerous layers after each other, as done in common architectures [2]. Graph convolution [69], is a highly efficient, effective and differentiable module that generalizes the neighborhood definition used in standard convolution and allows long-range information exchange in a single layer. This is done by defining edges $\mathbb{E}$ among nodes $\mathbb{V}$ in a graph $\mathbb{G}$. Formally, the graph convolution is defined as

$$\tilde{\mathbf{X}} = \sigma(\mathbf{WXA}), \tag{1}$$

where $\sigma(\cdot)$ is the non-linear activation function, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix characterising the neighbourhood relations of the graph and $\mathbf{W} \in \mathbb{R}^{D\times \tilde{D}}$ is the weight matrix. So the graph definition and structure play a key role in determining the information propagation.

2) Non-Local Network: We describe non-local network [7] in view of a fully connected graphical model. For a 2D input feature with the size of $C \times H \times W$, where C, H, and W denote the channel dimension, height, and width respectively, it can be interpreted as a set of features, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^\intercal$, $\mathbf{x}_i \in \mathbb{R}^C$, where N is the number of nodes ($e.g. N = H \times W$), and C is the node feature dimension.

$$\tilde{\mathbf{X}} = \delta(\mathbf{X}_\theta \mathbf{X}_\phi{}^\intercal)\mathbf{X}_\mathbf{g} = \mathbf{A}(\mathbf{X})\mathbf{X}_\mathbf{g}, \tag{2}$$

where $\mathbf{A}(\mathbf{X}) \in \mathbb{R}^{N\times N}$ indicates the affinity matrix, $X_\theta \in \mathbb{R}^{N\times C'}$, $X_\phi \in \mathbb{R}^{N\times C'}$, and $X_g \in \mathbb{R}^{N\times C'}$ which are projection matrix. In summary, according to Equ. 1 and Equ. 2 both the computation and affinity cost are highly dependent on the number of node N.

3) 'Squeeze' Operation: The 'squeeze' operation is commonly used in networks for image classification, and adopted in SE-net [8] to summarize the global contexts for intermediate layers for channel weights re-calibration. One simple implementation of this operation is the Global Average Pooling (GAP).

B. SR Module Formulation

As discussed, pixels or nodes' choices are essential for reducing computation cost for both graph convolution models and non-local models. Recent works [10], [14] follow this idea to achieve less computation cost. However, both the affinity memory and computation cost are still linearly dependent on the input resolution. In particular, this will limit their usage for some applications such as road scene understanding with high-resolution image inputs. Different from their approaches, we propose a simple yet effective framework named Squeeze Reasoning. Our approach mainly contains three steps. First, we squeeze the input feature map into a compact global vector. We then split such vector into different groups or nodes and perform graph reasoning operation on such input node graphs. Finally, we reconstruct the original feature map by multiplying the reasoned vector back into the input feature. We specify the details of these three steps in the following parts. Fig. 3 shows the detailed pipeline of our SR module.

1) Node Squeezing: It is well known that the global vector describes whole image statistics, which is a key component in many modern convolutional network architectures for different tasks such as object detection, scene parsing and image generation. The simplest way to calculate the global vector is the global average pooling, which calculates the first-order whole image statistics.

Recent works [25], [41] use the bilinear pooling to capture second-order statistics of features and generate global representations. Compared with the conventional average and max pooling, which only computes first-order statistics, bilinear pooling can better capture and preserve complex relations. In particular, bilinear pooling gives a sum pooling of second-order features from the outer product of all the feature vector pairs $(\mathbf{b}_i, \mathbf{c}_i)$ within two input feature maps $\mathbf{B}$ and $\mathbf{C}$. It can be expressed as follows:

$$\mathbf{G}_{bilinear}(\mathbf{B}, \mathbf{C}) = \mathbf{B}\mathbf{C}^T = \sum_{i=1}^{HW} \mathbf{b}_i \mathbf{c}_i^T, \tag{3}$$

where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_i, \ldots, \mathbf{b}_{HW}] \in \mathbb{R}^{C\times HW}$ and $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_i, \ldots, \mathbf{c}_{HW}] \in \mathbb{R}^{C\times HW}$. The output size is $C \times C$.

To get a more compact vector for each node, in this paper, instead of generating outer product of all feature pairs from $\mathbf{b}_i$ and $\mathbf{c}_i$ within two input feature maps $\mathbf{B}$ and $\mathbf{C}$, we calculate the Hadamard product:

$$\mathbf{G}_{global}(\mathbf{B}, \mathbf{C}) = \sum_{i=1}^{HW} \mathbf{b}_i \circ \mathbf{c}_i, \tag{4}$$

where $\circ$ means the Hadamard product operation. To be note that, we first reduce channel dimension of input feature X by $1\times$ 1 convolution layer into $\tilde{X}$ and then we perform pooling operation using Equ. 4 or simple global average pooling.

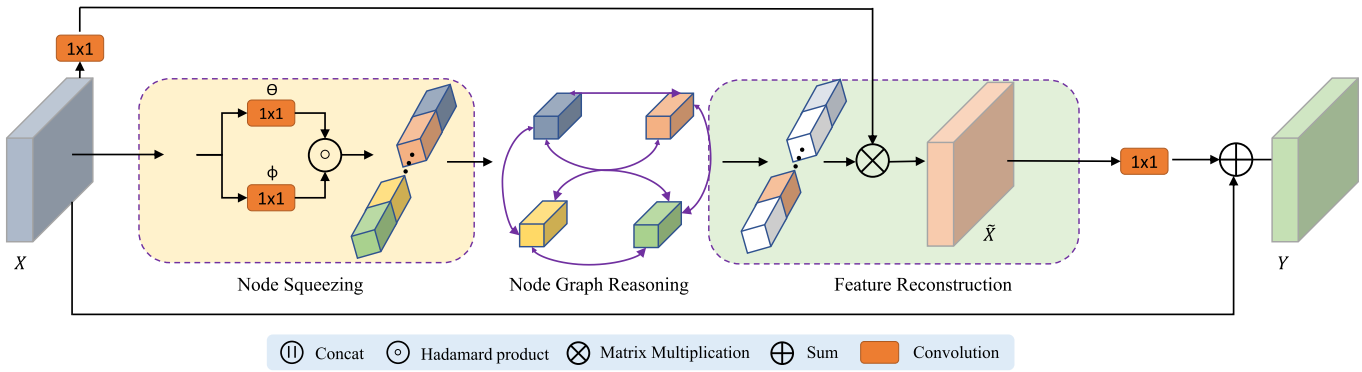

Fig. 3. Schematic illustration of our proposed SR module. Our module contains three steps. Node Squeezing: squeeze the feature into separate nodes. Node Graph Reasoning: perform GCN reasoning in node space. Feature Reconstruction: reconstruct the feature by the reasoned global vector.

2) Node Graph Reasoning: To form a node graph, we divide vector g into k different groups with each group size of d where $C = k \times d$. We use the graph convolution to model the relationship between nodes and consider it a fully-connected graph. As for the transformation $\mathbf{W}$ in Equ. 1, we adopt a 1×1 convolutional layer to implement it. Moreover, for the adjacency matrix $\mathbf{A}$, we will show by experiments that our Squeeze Reasoning mechanism is not sensitive to these choices, indicating that the generic graph reasoning behavior is the main reason for the observed improvements. We will describe two specific choices in the following part:

a) Learned matrix: We follow the same settings in GloRe [9], a simple choice of $\mathbf{A}$ is a 1×1 convolutional layer that can be updated by the general backpropagation. Similar to previous works [9], [44], we consider adopting the Laplacian matrix $\left(\mathbf{I} - \mathbf{A}_g\right)$ to propagate the node features over the graph, where the identity matrix $\mathbf{I}$ serves as a residual sum connection. In this setting, the Graph Reasoning can be formulated as follows:

$$\mathbf{G}_{output} = \sigma\left(\mathbf{W}_g \mathbf{G}_{input}\left(\mathbf{I} - \mathbf{A}_g\right)\right), \tag{5}$$

where σ is the ReLU operation.

b) Correlation matrix: Another choice is to adopt the self-attention mechanism for information exchange [7], [40] where the correlation matrix (or dense affinity) is calculated by the projection of node feature itself, by which the reasoning process can be written as follows:

$$\mathbf{G}_{output} = \sigma\left\{\rho_g(\mathbf{G}_{input})[\phi_g(\mathbf{G}_{input})^T \theta_g(\mathbf{G}_{input})]\right\}, \tag{6}$$

where ϕ_g, θ_g and ρ_g are three 1×1 convolutions. ϕ_g and θ_g are named 'query' and 'key', respectively. The ρ_g operation here, named 'value', functions the same as the $\mathbf{W}_g$ in the 'Learned matrix' mechanism. The results generated by operations inside the [.] form the adjacency matrix $\mathbf{A}$. To be noted that, either reasoning process can be adopted in our framework and more detailed results can be referred to the experiment part.

3) Feature Reconstruction: The final step is to generate the representation $\mathbf{R}$. To reconstruct the feature map, we first reshape the reasoned vector and multiply it with X to highlight different channels according to the input scene where $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{G}_{output}$. Then we adopt another 1×1 convolution layers W_R to project the $\tilde{X}$ into original shape. Following the same idea of residual learning [2], [7], we get the final output $\mathbf{Y}$ by adding original input $\mathbf{X}$. Then the feature map can be reconstructed as follows:

$$\mathbf{Y} = \mathbf{W}_\mathbf{R}\tilde{\mathbf{X}} + \mathbf{X} \tag{7}$$

where W_R is a learnable linear transformation and Y is the final output feature.

C. Analysis and Discussion

1) Relationship With Previous Operators: Compared with the Non-local block [7], instead of affinity modeling on pixel-level, our SR extends the reasoning on channel dimension, which captures statistics of whole feature map space. Compared with the SE block [70], our SR module captures more relational information and performs channel diffusion more efficiently than the fixed fully connected layers. Moreover, the experiment results show the advantages of our module.

2) Computation and Memory Analysis: Compared with previous self-attention methods [7], [10], [41], we compare our module computation in Tab. II which shows our module is both lightweight on both computation and memory compared with previous methods. To be noted, we only consider the reasoning part in both computation cost and affinity memory. As shown in the last row, our module linearly depends on the input resolution in terms of computation and has no relation with the input resolution in terms of affinity memory. More analysis can be found in the experimental part.

3) Discussion With EncNet [15]: Despite both EncNet and our SRnet both adopt the channel attention framework [8], our method is different from EncNet. For squeezing process, EncNet learns an inherent dictionary to represent the semantic context of the dataset which is dataset dependent, while our method squeezes the entire feature map into one compact vector via learnable convolutions which is data dependent. The dataset dependent dictionary captures the class relation prior on specific dataset and thus it is hard for generalization. Our approach is more general and it can be a plug-in-play module. In addition, we have proven this point on many other tasks including object detection, instance segmentation and panoptic segmentation. For reasoning process, our method divides the

TABLE II

TIME COMPLEXITY COMPARISON OF NON-LOCAL OPERATIONS WITH OUR PROPOSED SR MODULE WHERE H AND W IS SPATIAL RESOLUTION AND C IS CHANNEL DIMENSION. P IS THE ORDER OF TAYLOR EXPANSION OF KERNEL FUNCTION IN [36] AND $KM = C/2$. NOTE THAT WE IGNORE THE CHANNEL REDUCTION PROCESS SINCE THEY ARE EQUAL FOR COMPUTATION COST

Methods	Computation	Affinity Memory
Non-local [7]	$O(C(HW)^2)$	$O((HW)^2)$
A2Net [41]	$O(C^2(HW))$	$O(C^2)$
CGNL [36]	$O(CHWP))$	$O(P^2)$
CCNet [10]	$O(CHW(H+W)$	$O(HW(H+W))$
DANet [40]	$O(C(HW)^2 + HW(C)^2)$	$O((HW)^2 + (C)^2)$
SRNet	$O(CHW + C)$	$O(K^2 + M^2)$

TABLE III

ABLATION STUDY ON THE COMPONENTS OF THE PROPOSED SR MODULE. (A) EXPLORATION ON SR MODULE DESIGN. **GHP**: GLOBAL HADAMARD POOLING. **FC**: FULLY-CONNECTED LAYERS. **GCN**: REASONING WITH THE GRAPH CONVOLUTION AS EQ. 5. **SA**: REASONING USING THE SELF-ATTENTION MECHANISM AS EQ. 6

Squeeze		Reasoning			mIoU (%)	Δ(%)
GAP	GHP	FC	GCN	SA		
-	-	-	-	-	74.8	-
✓	-	-	-	-	75.3	+0.5
-	✓	-	-	-	76.3	+1.5
✓	-	✓	-	-	76.6	+1.8
-	✓	✓	-	-	76.8	+2.0
✓	-	-	✓	-	**79.1**	+4.3
-	✓	-	✓	-	**79.9**	+5.1
-	✓	-	-	✓	**79.5**	+4.7

global feature into different groups. Each group represents a latent class and the information diffusion is achieved by one graph convolution layer while EncNet adopts SE-like architecture. Our divided and reasoning process shows better results than SENet in Table-X. Moreover, EncNet also uses a semantic loss to achieve better results while our method only uses the cross-entropy loss as naïve FCN with much less tricks. Finally, we show the full advantages over different datasets under different settings over EncNet in the experimental part.

D. Network Architecture

The proposed SR module can be easily incorporated into the existing CNN architectures. We detail our network design in the task of semantic segmentation and instance level segmentation.

1) Semantic Segmentation: We adopt the Fully Convolution Networks (FCNs) [1] as the backbone model. In particular, we choose ImageNet [19] pretrained ResNet [2], remove the last two down-sampling operations and adopt the multi-grid [27] dilated convolutions. We remove the last two down-sampling operations and use the dilation convolutions instead to hold the feature maps from the last two stages $\frac{1}{8}$ of the input image. Concretely, all the feature maps in the last three stages have the same spatial size. Following the same setting [10], [11], we insert our proposed module between two 3×3 convolution layers (both layers output $D = 512$ channels), which are appended at the end of the FCN. Following [24], our model has two supervisions: one after the final output of our model while another at the output layer of Stage4 as auxiliary cross-entropy loss. For real-time segmentation models, we choose DF-seg models [71] as a baseline and we replace their head with our SR module.

2) Instance Level Segmentation: For instance segmentation and panoptic segmentation, We choose two-stage mask-rcnn-like architectures [4], [50] We insert our module on the outputs of bottleneck in ResNet [2] layer4 for context modeling.

IV. EXPERIMENT

We verify the proposed module on four scene understanding tasks, including semantic segmentation, object detection, instance segmentation, panoptic segmentation and image classification. Our method outperforms several state-of-the-art methods on four benchmarks for semantic segmentation, including Cityscapes, ADE20K, Pascal Context and Camvid, with much less computation cost. Experiments on the other four vision tasks also demonstrate the effectiveness of the proposed module. All the experiments are under the same setting for each task and each dataset for a fair comparison.

A. Ablation Experiments on Cityscapes Dataset

1) Experimental Settings on Ablation Studies: We carry out detailed ablation studies and visual analysis on proposed approaches. We implement our method based on the PyTorch framework [72]. For the Cityscapes dataset, following the same settings in PSPNet [24]where momentum and weight decay coefficients are set to 0.9 and 5e-4 respectively, and "poly" learning rate schedule is used. For ablation studies, we choose ResNet-50 as the backbone where momentum and weight decay coefficients are set to 0.9 and 5e-4 respectively, and "poly" learning rate schedule is used which decays initial learning rate of 0.01 by multiplying $(1 - \frac{\text{iter}}{\text{total\_iter}})^{0.9}$. Synchronized batch normalization is used [15]. For data augmentation, random cropping with size 769 and random left-right flipping are adopted. For the ablation studies, all models are trained by 50,000 iterations and evaluated by sliding-window crop inference. For the real time models, we use single scale full image inference.

2) Ablation on SR Framework Design: We first present a detailed analysis of each component of SR through ablation study and report results in Tab. III(a). Comparing with the baseline, all SR versions equipped with different components achieve considerable improvements. By switching different squeezing operations, we find Global Hadamard Pooling (GHP) performs consistently better than Global Average Pooling (GAP) across differently used reasoning methods. Moreover, reasoning with Graph Convolutional Network and Self-Attention brings more improvements, even comparing with methods using global information from squeezing operation and further transformed by fully-connected layers(FC). Our segmentation models are trained under the best setting in this table.

3) Ablation on Hyper-Parameter Settings: To select the best hyper-parameter K, we also carry out an ablation study on the

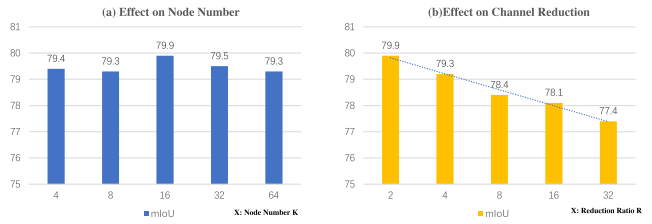

Fig. 4. Ablation on hyper-parameter settings. (a) Effect on node number. (b) Effect on channel reduction.

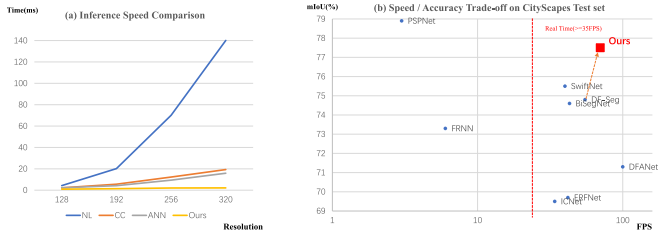

Fig. 5. (a).Speed comparison with non-local and its variants. (b). Speed and accuracy trade-off on Cityscapes Test Set for real time models. Best view it zoom in.

TABLE IV

COMPARISON EXPERIMENTS USING DIFFERENT CONTEXT MODELING METHODS ON THE CITYSCAPES VALIDATION SET, WHERE DILATED FCN (RESNET-50) SERVES AS THE BASELINE METHOD. THE FLOPS AND THE MEMORY (MEM) ARE COMPUTED OVER THE INPUT IMAGE OF SIZE $768 \times 768 \times 3$ AND INPUT FEATURE MAP OF SIZE $96 \times 96 \times 2048$. FOR MODELS OTHER THAN ASPP AND PSP, THE OVERHEAD OF THE 3×3 CONVOLUTIONS BEFORE AND BEHIND THE MODULES ARE SHOWN SEPARATELY AS THE ROW $+2 * 3 \times 3$. $\uparrow$ MEANS THE RELATIVE OVERHEADS OVER THOSE OF $+2 * 3 \times 3$. ALL THE METHODS ARE EVALUATED UNDER THE SAME SETTING FOR THE FAIR COMPARISON

Method	mIoU (%)	FLOPS	Params	Mem
dilated FCN	74.8	241.05G	23.63M	3249M
+ASPP [30]	77.4	+148.37G	+15.54M	+191.20M
+PSP [24]	77.2	+174.09G	+23.07M	+221.05M
$+2 * 3 \times 3$	-	+108.74G	+11.80M	+108.00M
+SE [8]	74.6	**9.47M** $\uparrow$	**0.03M**$\uparrow$	**18.88M**$\uparrow$
+NL [7]	78.0	48.36G$\uparrow$	0.53M$\uparrow$	865.52M$\uparrow$
+A2Net [41]	78.1	4.94G$\uparrow$	0.53M$\uparrow$	183.32M$\uparrow$
+CGNL [36]	78.2	4.91G$\uparrow$	0.53M$\uparrow$	201.94M$\uparrow$
+RCCA [10]	78.5	11.55G$\uparrow$	0.53M$\uparrow$	394.28M$\uparrow$
+Encoding [15]	77.5	12.31G$\uparrow$	0.59M$\uparrow$	257.12M$\uparrow$
+ANN [11]	78.4	10.41G$\uparrow$	0.68M$\uparrow$	421.12M$\uparrow$
+EMAU [16]	77.9	6.97G$\uparrow$	0.54M$\uparrow$	132.64M$\uparrow$
+SR (GAP)	**79.1**	2.43G$\uparrow$	0.26M$\uparrow$	82.31M$\uparrow$
+SR (GHP)	**79.9**	3.64G$\uparrow$	0.40M$\uparrow$	110.75M$\uparrow$

number of groups. To control independent variables, we fix $KM = C/2$, and only adjust K. We also explore the effect of channel reduction ratio with fixed K. The results are shown in Fig 4. From which, we can see that the selection of K doesn't influence too much while reducing channel leads to inferior results. We set $K = 16$ and ratio to 2 as default for the remaining experiments.

4) Comparisons With Context Aggregation Approaches: In Tab. IV, we compare the performance of different context aggregation approaches, where SR achieves the best mIoU with ResNet-50 as the backbone. We give detailed and fair comparisons in terms of flops, parameters and memory cost. In particular, SR performs even better than all the non-local methods [7], [10], [15], [36], which aggregates long-range contextual information in a pixel-wise manner. This indicates the effectiveness of cross-channel relationships in building compact and better representations with fewer computation FLOPS. Fig 5(a) gives inference time comparison with different resolution image inputs on V100-GPU, which shows the advantages with high-resolution image inputs.

5) Visualization and Analysis on Learned Node Representation: Here, we give visualization analysis on different channel activation on the reasoned feature map. From Fig 6, we can see that each item corresponds to some abstract conceptions in the image. For example, the third-row item attends on the trucks and the cars while the second column shows that items focus on the stuff and background, and the fourth column items in group-10 are more sensitive to the small objects like poles and boundaries.

6) Visualization on Predictions and Feature Maps: Fig. 7 compares segmentation results with and without reasoning. Without reasoning, pixels of large objects such as trucks and buses are often misclassified into similar categories due to ambiguities caused by limited receptive fields. The reasoning module resolves the above issue and delivers more consistent segmentation inside objects. Fig. 8 further investigates the effects of SR by directly comparing its input and output feature maps, where SR significantly improves the consistency of features inside objects, which is also the reason for consistent semantic map prediction. After SR, the features inner the objects have similar color and clearer boundaries shown the second column in Fig 8.

B. Experiments on Cityscapes in Real-Time Settings

1) Experiment Settings: Due to the efficiency of the proposed approach, we extend our method into real-time training settings. We mainly follow the DFANet [63]. The networks with SR head are trained with the same setting, where stochastic gradient descent (SGD) with batch size of 16 is used as optimizer, with momentum of 0.9 and weight decay of 5e-4. All models are trained for 50K iterations with an initial learning rate of 0.01. As a common practice, the "poly" learning rate policy is adopted to decay the initial learning rate by multiplying $(1 - \frac{\text{iter}}{\text{total\_iter}})^{0.9}$ during training. Data augmentation contains random horizontal flip, random resizing with scale range of [0.75, 2.0], and random cropping with crop size of 1024×1024. During inference, we use the whole picture as input to report performance. To be more specific, we replace PPM head [24] in DF-Seg-v2 [71] with our module. Tab. V shows the results of our real-time model. Compared with baseline DFSeg [71], Our method has similar parameters but with more accurate and faster speed.

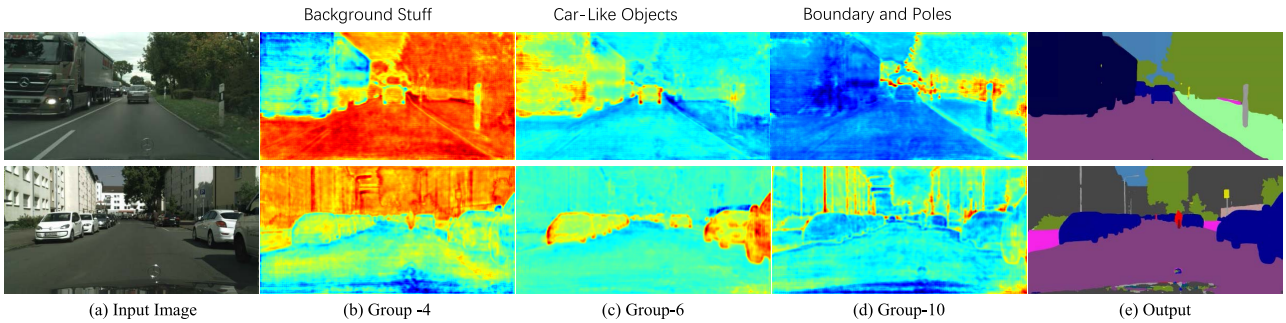

Fig. 6. Visualization on learned group representation. We select the most salient channel from each node group. Such items capture specific concepts in the images. Best view zoom in.

TABLE V

COMPARISON ON CITYSCAPES *test* SET WITH STATE-OF-THE-ART REAL-TIME MODELS. FOR FAIR COMPARISON, INPUT SIZE IS ALSO CONSIDERED, AND ALL MODELS USE SINGLE SCALE INFERENCE

Method	InputSize	mIoU (%)	#FPS	#Params
ESPNet [73]	512 × 1024	60.3	132	0.4M
ESPNetv2 [74]	512 × 1024	62.1	80	0.8M
ERFNet [75]	512 × 1024	69.7	41.9	-
BiSeNet(ResNet-18) [61]	768 × 1536	74.6	43	12.9M
BiSeNet(Xception-39) [61]	768 × 1536	68.4	72	5.8M
ICNet [60]	1024 × 2048	69.5	34	26.5M
DFv1 [71]	1024 × 2048	73.0	80	9.37M
SwiftNet [64]	1024 × 2048	75.5	39.9	11.80M
SwiftNet-ens [64]	1024 × 2048	76.5	18.4	24.7M
DFANet [71]	1024 × 1024	71.3	100	7.8M
CellNet [76]	768 × 1536	70.5	108	-
DFv2(baseline) [71]	1024 × 2048	74.8	55	18.83M
SRNet(DFv2)	1024 × 2048	**77.5**	**65**	**18.87M**

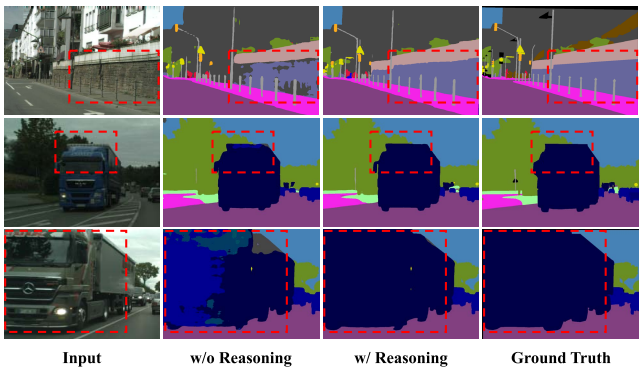

Fig. 7. Comparison of our results from cropped images where dilated-FCN with GAP- Squeeze operation as the baseline model. Best view it zoom in.

2) Comparisons With Real Time Models on Cityscapes: We set a new record on **best speed and accuracy trade-off** in Cityscapes test set with 77.5 %mIoU and 70 FPS shown in Fig. 5 with full image resolution inputs (1024×2048), which indicates the practical usage of our method. During inference, we use the whole picture as input to report performance. Tab V shows the results of our real-time model. Compared with previous real time models, our method achieves the best speed and accuracy trade-off. Compared with baseline DFSeg [71], Our method has similar parameters but with more accurate and faster speed.

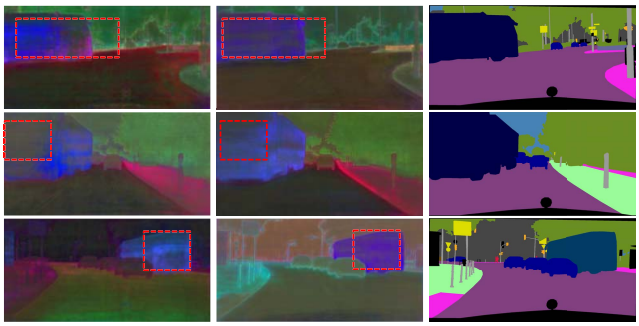

Fig. 8. The input and the output feature maps of the SR module. They are projected from 512-d to 3-d by PCA. Best view it zoom in.

C. Comparisons With State-of-the-Art Methods in Non-Real-Time Setting

In this section, we compare our method with state-of-the-art methods on four semantic segmentation benchmarks using multi scale inference setting. Without bells and whistles, our method outperforms several state-of-the-art models while costing less computation.

1) Results on Cityscapes: We train our model for 120K iterations using only the finely annotated data (trainval set), online hard negative mining is used following [40]. Multi-scale and horizontal flip testing is used as previous works [10]. Tab. VII(a) compares the results, where our methods achieves **82.2%** mIoU and outperforms all previous state-of-the-art models by a large margin. In particular, our method is 0.7% mIoU higher than DANet [40], which uses non-local-like operator and is much efficient in both computation and memory cost due to the design of squeeze and reasoning. Our ResNet-50 based model achieves 81.0% mIoU and outperforms DenseASPP [31] by 0.4% with much larger backbone [77], which shows the effectiveness of our method. After replacing stronger backbone Wider-ResNet [78], we achieve **83.3%** mIoU with **only** fine annotated data, which outperforms previous state-of-the-art methods by a large margin. Note that we follow the G-SCNN [54] setting by using Deeplabv3+ based Wider-ResNet pretrained on Mapillary [79]. The detailed results are shown in Tab. VI for reference.

TABLE VI

PER-CATEGORY RESULTS ON THE CITYSCAPES TEST SET COMPARED WITH ACCURATE MODELS. NOTE THAT ALL THE MODELS ARE TRAINED WITH ONLY FINE ANNOTATED DATA. OUR METHOD WITH RESNET101 BACKBONE OUTPERFORMS EXISTING APPROACHES ON **15** OUT OF 19 CATEGORIES, AND ACHIEVES **82.2%** MIOU. SS MEANS SINGLE SCALE INFERENCE

Method	road	swalk	build	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
DUC-HDC [80]	98.5	85.5	92.8	58.6	55.5	65.0	73.5	77.8	93.2	72.0	95.2	84.8	68.5	95.4	70.9	78.7	68.7	65.9	73.8	77.6
ResNet38 [79]	98.5	85.7	93.0	55.5	59.1	67.1	74.8	78.7	93.7	72.6	95.5	86.6	69.2	95.7	64.5	78.8	74.1	69.0	76.7	78.4
PSPNet [24]	98.6	86.2	92.9	50.8	58.8	64.0	75.6	79.0	93.4	72.3	95.4	86.5	71.3	95.9	68.2	79.5	73.8	69.5	77.2	78.4
AAF [56]	98.5	85.6	93.0	53.8	58.9	65.9	75.0	78.4	93.7	72.4	95.6	86.4	70.5	95.9	73.9	82.7	76.9	68.7	76.4	79.1
SegModel [81]	98.6	86.4	92.8	52.4	59.7	59.6	72.5	78.3	93.3	72.8	95.5	85.4	70.1	95.6	75.4	84.1	75.1	68.7	75.0	78.5
DFN [37]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	79.3
BiSeNet [61]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78.9
PSANet [42]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	80.1
DenseASPP [31]	98.7	87.1	93.4	60.7	62.7	65.6	74.6	78.5	93.6	72.5	95.4	86.2	71.9	96.0	**78.0**	**90.3**	80.7	69.7	76.8	80.6
BFPNet [82]	98.7	87.1	93.5	59.8	**63.4**	68.9	76.8	80.9	93.7	72.8	95.5	87.0	72.1	96.0	77.6	89.0	86.9	69.2	77.6	81.4
DANet [40]	98.6	87.1	93.5	56.1	63.3	69.7	77.3	81.3	93.9	72.9	95.7	87.3	72.9	96.2	76.8	89.4	86.5	**72.2**	78.2	81.5
SRNet(ss)-ResNet101	98.7	86.7	93.5	60.9	61.7	68.3	76.6	79.9	93.7	72.4	95.8	86.9	72.3	96.1	76.1	87.2	88.2	70.3	77.5	81.2
SRNet-ResNet101	**98.8**	**88.0**	**93.9**	**64.6**	63.3	**71.5**	**78.9**	**81.8**	**93.9**	**73.7**	**95.8**	**87.9**	**74.5**	**96.4**	72.4	88.2	**86.2**	72.0	**79.0**	**82.2**
SRNet-WiderResNet	98.8	87.9	94.2	65.2	66.0	72.4	77.8	81.5	94.1	75.0	96.5	87.9	74.9	96.4	78.4	93.4	88.1	73.9	78.6	83.3

TABLE VII

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON ROAD-DRIVING SCENE DATASETS INCLUDING CITYSCAPES AND CAMVID

Method	Backbone	mIoU (%)
DFN [37]	ResNet-101	79.3
CFNet [39]	ResNet-101	79.6
DenseASPP [31]	DenseNet-161	80.6
GloreNet [9]	ResNet-101	80.9
BAFPNet [82]	ResNet-101	81.4
CCNet [10]	ResNet-101	81.4
ANNet [11]	ResNet-101	81.3
DANet [40]	ResNet-101	81.5
RGNet [17]	ResNet-101	81.5
DGMN [87]	ResNet-101	81.6
OCRNet [14]	ResNet-101	81.8
SRNet	ResNet-50	**81.0**
SRNet	ResNet-101	**82.2**
G-SCNN [54]	Wider-ResNet-38	82.8
SRNet	Wider-ResNet-38	**83.3**

(a) Results on the Cityscapes test set. All methods use only finely annotated data.

Method	Backbone	mIoU (%)
SegNet [70]	VGG-16	60.1
RTA [84]	VGG-16	62.5
BiSeg [61]	ResNet-18	68.7
PSPNet [24]	ResNet-50	69.1
SRNet	ResNet-50	**74.3**
DilatedNet [27]	ResNet101	65.3
Dense-Decoder [57]	ResNext-101	70.9
BFP [82]	ResNet101	74.1
VideoGCRF [83]	ResNet101	75.2
SRNet	ResNet-101	**78.3**

(b) Results on the CamVid test set.

TABLE VIII

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON MORE SCENE PARSING DATASETS INCLUDING ADE20K AND PASCAL CONTEXT

Method	Backbone	mIoU (%)
EncNet [15]	ResNet-50	49.2
DANet [40]	ResNet-50	50.1
SRNet	ResNet-50	**50.8**
EncNet [15]	ResNet-101	51.7
Ding *et al.* [53]	ResNet-101	51.6
DANet [40]	ResNet-101	52.6
SGR [85]	ResNet-101	52.5
ANN [11]	ResNet-101	52.8
BAFPNet [82]	ResNet-101	53.6
EMANet [16]	ResNet-101	53.1
GFFNet [52]	ResNet-101	54.2
CFNet [39]	ResNet-101	54.1
SPNet [86]	ResNet-101	54.5
APCNet [87]	ResNet-101	**54.7**
SRNet	ResNet-101	**54.7**

(a) Results on Pascal Context dataset.

Method	Backbone	mIoU (%)
PSPNet [24]	ResNet-50	42.78
PSANet [42]	ResNet-50	42.97
UperNet [49]	ResNet-50	41.55
EncNet [15]	ResNet-50	41.11
GCUNet [44]	ResNet-50	42.60
SRNet	ResNet-50	**43.42**
PSPNet [24]	ResNet-101	43.29
PSANet [42]	ResNet-101	43.77
SAC [47]	ResNet-101	44.30
EncNet [15]	ResNet-101	44.65
GCUNet [44]	ResNet-101	44.81
ANN [11]	ResNet-101	45.24
SRNet	ResNet-101	**45.53**

(b) Results on the ADE20K dataset.

2) Results on CamVid: is another road driving dataset. Camvid involves 367 training images, 101 validation images and 233 testing images with resolution of 960×720. We use a crop size of 640 and training with 100 epochs. The results are shown in Tab VII(b). We report results with ResNet-50 and ResNet-101 backbone. With ResNet-101 as backbone, our method achieves **78.3%** mIoU, outperforming the state-of-the-art approach [83] by a large margin (3.1%).

3) Results on Pascal Context: This dataset provides detailed semantic labels for the whole scenes [21]. It contains 4998 images for training and 5105 images for validation. We train the network for 100 epochs with a batch size of 16, a crop size of 480. For evaluation, we perform multi-scale testing with the flip operation, which boosts the results by about 1.2% in mIoU. Fig. 9 shows the results of our method

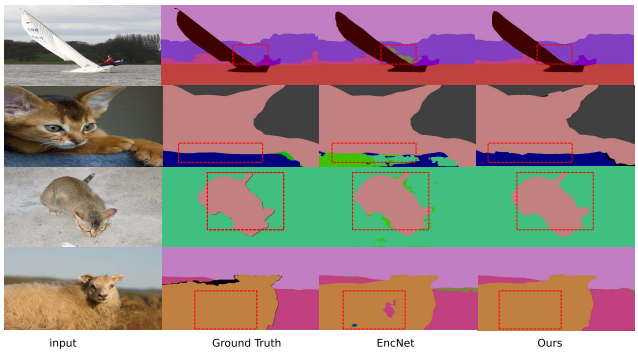

Fig. 9. Comparison of our results on Pascal Context to the state-of-the-art EncNet [15]. Note that our results are more consistent and have fewer artifacts. Best view zoom in.

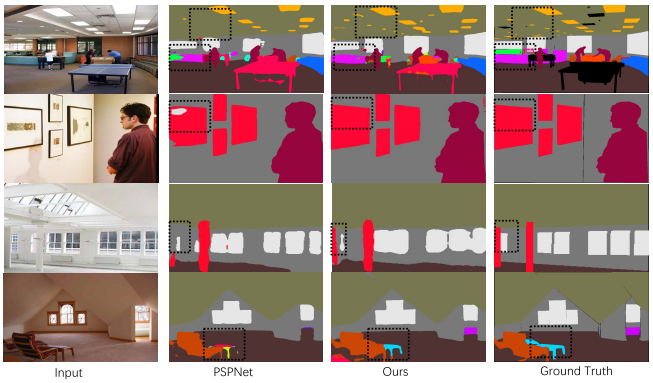

Fig. 10. Comparison of our results with the state-of-the-art PSPNet [24] method on the ADE-20k dataset. The black boxes show that our method can get more consistent results and missing objects. Best view zoom in.

and EncNet. Compared with EncNet [15], our method achieves better consistent results on the object inner parts, benefited from better reasoned features. Tab. VIII(a) reports results on Pascal Context. With ResNet-101 as the backbone, our method achieves 54.7% in mIoU with multi-scale inference, surpassing state-of-the-art alternatives by a large margin. Additionally, using the ResNet-50 backbone, we achieve 50.8% mIoU, which also outperforms the previous work [40] under the same setting.

4) Results on ADE20K: This is a more challenging scene parsing dataset annotated with 150 classes, which it contains 20k and 2k images for training and validation, respectively. We train the network for 120 epochs with a batch size of 16, a crop size of 512 and an initial learning rate of 1e-2. We perform multi-scale testing with the flip operation as the common setting in [24]. The visible results are shown in Fig. 10. Compared to PSPNet [24], our method better handles inconsistent results and missing objects in the scene. In Tab. VIII(b), Both results using ResNet-50 and ResNet-101 backbone are reported. As shown in Tab. VIII(b), Our method with ResNet-101 achieves the best results and comparable results with ResNet-50 backbone. Our methods have less computation cost in the head part. Thus it results in an efficient inference.

TABLE IX

EXPERIMENTS ON COCO DATASET. (A) DETECTION RESULTS ON THE COCO 2017 VALIDATION SET. **R-50**: RESNET-50. **R-101**: RESNET-101. **X-101**: RESNEXT-101 [90]. **NL**: NON-LOCAL BLOCKS [7]. (B) PANOPTIC SEGMENTATION RESULTS ON THE COCO 2017 VALIDATION SET

Backbone	Detector	AP-box	AP-mask	GFlops
R-50	Mask-RCNN	37.2	33.8	275.58
R-50	+NL [7]	38.0	34.7	+30.5
R-50	**+SR**	**38.4**	**34.9**	+0.56
R-101	Mask-RCNN	39.8	36.0	351.65
R-101	+NL	40.5	36.7	+45.7
R-101	**+SR**	**40.8**	**36.9**	+1.32
X-101	Mask-RCNN	41.2	37.3	355.37
X-101	**+SR**	**42.0**	**37.8**	+1.32
X-101	Cascaded-Mask-RCNN	44.7	38.3	519.90
X-101	**+SR**	**45.5**	**39.0**	+1.32

(a) Experiments on COCO Object Detection and Instance Segmentation using various baseline models.

Method	Backbone	PQ	PQ (things)	PQ (stuff)	GFlops
Base	ResNet-50	39.0	45.9	28.7	270.8
+NL	ResNet-50	39.8	46.8	28.9	+30.5
+SR	ResNet-50	**40.3**	**47.2**	**29.9**	+0.56
Base	ResNet-101	40.3	47.5	29.5	346.87
+NL	ResNet-101	40.8	48.5	30.4	+45.7
+SR	ResNet-101	**41.8**	**48.7**	**31.3**	+1.32

(b) Experiments on COCO Panoptic Segmentation using PanopticFPN as baseline model.

D. Results on MS COCO

To verify our module's generality, we further conduct experiments on MS COCO [18] for more tasks, including object detection, instance segmentation and panoptic segmentation. The trainval set has about 115k images, the minival set has 5k images. We perform training on trainval set and report results on minival set. For the first two tasks, our model is based on the state-of-the-art method Mask R-CNN [4] and its variants [32], [88]. For panoptic segmentation, we choose Panoptic FPN as our baseline [50]. We use open-source tools [89] to carry out all the experiments and report results on the MS COCO validation dataset. The GFlops are measured with 1200×800 inputs. Our models and all baselines are trained with the typical '1x' training schedule and setting from the public mmdetection [89] for all experiments on COCO.

1) Results on Object Detection and Instance Segmentation: Tab. IX(a) compares results of both object detection and instance segmentation with various backbone networks [90] and advanced method [88], where our method achieves consistently better performance on all backbones with much less computation cost compared with Non-Local blocks [7].

2) Results on Panoptic Segmentation: Panoptic Segmentation [5] uses the PQ metric to capture the performance for all classes (stuff and things) in a unified way. We use the PanopticFPN [50] as our baseline model and follow the standard settings in mmdetection. We re-implement the baseline model using mmdetection tools and achieve the similar results with original paper [50]. The results are shown in Tab. IX(b). Our method improves baseline and outperforms the non-local based methods through both overall evaluation and evaluations separated into thing and stuff, and the improvements are across both backbones, ResNet-50 and ResNet-101 with less computation cost.

(a) mask-rcnn-baseline (b) + SR

Fig. 11. Comparison of our results on COCO with Mask-RCNN with ResNet101 backbone. Best view zoom in.

TABLE X

EXPERIMENTS RESULTS ON IMAGENET. OUR METHOD ACHIEVES BETTER TOP-1 ACCURACY WITH LOWER PARAMETER AND COMPUTATION COST ON STRONG BASELINE SETTINGS. R50 MEANS RESNET50 AS BACKBONE WHILE R101 MEANS RESNET101 AS BACKBONE

Method	Top-1	Params	GFLOPs
ResNet-r50	77.5	25.56M	4.122
SENet-r50 [8]	77.8	28.09M	4.130
CBAM-r50 [35]	78.0	28.09M	4.139
SRNet-r50	**78.1**	25.64M	4.127
ResNet-r101	78.7	44.55M	7.849
SENet-r101 [8]	79.1	49.33M	7.863
SRNet-r101	**79.3**	44.70M	7.858

3) Visualization Results on COCO: Fig 11 shows the qualitative results on COCO validation set. We use Mask-RCNN [4] with ResNet101 as the baseline model. The first two rows show our SR module can handle small missing objects (red boxes) while the last two rows show our method can also avoid false positives (blue boxes) in the scene.

E. Extension on ImageNet Classification

We also perform experiments for image classification on ImageNet dataset [19], where SENet [8] is used as our baseline model. To be noted, we only verify the effectiveness and generalization of SR framework for classification task. Our model is designed by replacing fully-connected layer with our proposed graph reasoning module, where the global hadamard pooling is not used for both saving computation and fair comparison with SEnet. All models are trained under the same setting, and results are shown in Table X. All networks are trained following the same strategy as [2] using Pytorch [72] with 8 GTX 1080Ti GPUs. In particular, cosine learning rate schedule with warm up strategy is adopt [91], and weight decay is set to 1e-4. SGD with mini-batch size 256 is used for updating weights. Top-1 and top-5 classification accuracy on validation set using single 224×224 central crop are reported for performance comparison. Due to the usage of cosine learning rate schedule [91], our baseline models on ImageNet are higher than the original paper [2], [8]. Compared with both SENet and CBAM [35], our SRNet improves the strong baseline SENet [8] by 0.3 in Top-1 accuracy with much less parameter and GFlops. Our method leads to higher accuracy with fewer parameters and FLOPs, which demonstrates both effectiveness and efficiency of the proposed method.

V. CONCLUSION

This paper proposes a novel Squeezing and Reasoning framework for highly efficient deep feature representation learning for the scene understanding tasks. It learns to squeeze the feature to a node graph space where each node represents an abstract semantic concept while both memory and computation costs are significantly reduced. Extensive experiments demonstrate that our method can establish considerable results on semantic segmentation while keeping efficiency compared with previous the-state-of-the-art models. It also shows consistent improvement with respect to strong baselines over instance segmentation and panoptic segmentation with much less computation. It also verified to be effective on image classification task and better results over SENet. The further work can be exploring cross layer reasoning over entire network.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of SenseTime Research for providing the computing resources.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, Oct. 2017, pp. 2961–2969.
[5] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. CVPR*, Jun. 2019, pp. 9404–9413.
[6] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. NIPS*, 2016, pp. 4905–4913.
[7] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. CVPR*, Jun. 2018, pp. 7794–7803.
[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, Jun. 2018, pp. 7132–7141.

[9] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proc. CVPR*, Jun. 2019, pp. 433–442.

[10] Z. Huang *et al.*, "CCNet: Criss-cross attention for semantic segmentation," 2018, *arXiv:1811.11721*. [Online]. Available: http://arxiv.org/abs/1811.11721

[11] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. ICCV*, Oct. 2019, pp. 593–602.

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

[13] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "OCNet: Object context network for scene parsing," 2018, *arXiv:1809.00916*. [Online]. Available: http://arxiv.org/abs/1809.00916

[14] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," 2019, *arXiv:1909.11065*. [Online]. Available: http://arxiv.org/abs/1909.11065

[15] H. Zhang *et al.*, "Context encoding for semantic segmentation," in *Proc. CVPR*, Jun. 2018, pp. 7151–7160.

[16] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. ICCV*, Oct. 2019, pp. 9167–9176.

[17] C. Yu, Y. Liu, C. Gao, C. Shen, and N. Sang, "Representative graph neural network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Nov. 2020, pp. 379–396.

[18] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[19] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[20] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, Jun. 2016, pp. 3213–3223.

[21] R. Mottaghi *et al.*, "The role of context for object detection and semantic segmentation in the wild," in *Proc. CVPR*, Jun. 2014, pp. 891–898.

[22] B. Zhou *et al.*, "Semantic understanding of scenes through the ADE20K dataset," 2016, *arXiv:1608.05442*. [Online]. Available: http://arxiv.org/abs/1608.05442

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

[24] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. CVPR*, Jul. 2017, pp. 2881–2890.

[25] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. ICCV*, Dec. 2015, pp. 1449–1457.

[26] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. ICCV*, Oct. 2019, pp. 3286–3295.

[27] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016, pp. 1–13.

[28] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. ICLR*, 2015, pp. 1–34.

[29] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[30] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[31] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

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

[33] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9308–9316.

[34] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*. [Online]. Available: http://arxiv.org/abs/1506.04579

[35] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. ECCV*, Sep. 2018, pp. 3–19.

[36] K. Yue, M. Sun, Y. Yuan, F. Zhou, E. Ding, and F. Xu, "Compact generalized non-local network," in *Proc. NIPS*, 2018, pp. 6511–6520.

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

[38] Z. Zhong *et al.*, "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13065–13074.

[39] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 548–557.

[40] J. Fu *et al.*, "Dual attention network for scene segmentation," 2018, *arXiv:1809.02983*. [Online]. Available: http://arxiv.org/abs/1809.02983

[41] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A^2-nets: Double attention networks," in *Proc. NIPS*, 2018, pp. 352–361.

[42] H. Zhao *et al.*, "PSANet: Point-wise spatial attention network for scene parsing," in *Proc. ECCV*, Sep. 2018, pp. 267–283.

[43] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[44] Y. Li and A. Gupta, "Beyond grids: Learning graph representations for visual recognition," in *Proc. NIPS*, 2018, pp. 9225–9235.

[45] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8950–8959.

[46] L. Zhang, D. Xu, A. Arnab, and P. H. S. Torr, "Dynamic graph message passing networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3726–3735.

[47] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan, "Scale-adaptive convolutions for scene parsing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2031–2039.

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[49] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. ECCV*, Sep. 2018, pp. 418–434.

[50] A. Kirillov, R. Girshick, K. He, and P. Dollar, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6399–6408.

[51] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[52] X. Li, Z. Houlong, H. Lei, T. Yunhai, and Y. Kuiyuan, "Gff: Gated fully fusion for semantic segmentation," in *Proc. AAAI*, 2020, pp. 11418–11425.

[53] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2393–2402.

[54] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5229–5238.

[55] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[56] T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu, "Adaptive affinity fields for semantic segmentation," in *Proc. ECCV*, Sep. 2018, pp. 587–602.

[57] P. Bilinski and V. Prisacariu, "Dense decoder shortcut connections for single-pass semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6596–6605.

[58] L.-C. Chen *et al.*, "Searching for efficient multi-scale architectures for dense image prediction," in *Proc. NIPS*, 2018, pp. 8713–8724.

[59] C. Liu *et al.*, "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 82–92.

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

[61] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. ECCV*, Sep. 2018, pp. 325–341.

[62] X. Li *et al.*, "Semantic flow for fast and accurate scene parsing," in *Proc. ECCV*, Aug. 2020, pp. 775–793.

[63] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9522–9531.

[64] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12607–12616.
[65] R. Gadde, V. Jampani, and P. V. Gehler, "Semantic video CNNs through representation warping," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4453–4462.
[66] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6819–6828.
[67] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2349–2358.
[68] Y. Li, J. Shi, and D. Lin, "Low-latency video semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5997–6005.
[69] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–13.
[70] 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.
[71] X. Li, Y. Zhou, Z. Pan, and J. Feng, "Partial order pruning: For best speed/accuracy trade-off in neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9145–9153.
[72] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. NeurIPS Workshops*, 2017, pp. 1–4.
[73] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. ECCV*, Sep. 2018, pp. 552–568.
[74] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9190–9200.
[75] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.
[76] V. Nekrasov, H. Chen, C. Shen, and I. Reid, "Fast neural architecture search of compact semantic segmentation models via auxiliary cells," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9126–9135.
[77] 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.
[78] Z. Wu, C. Shen, and A. van den Hengel, "Wider or deeper: Revisiting the ResNet model for visual recognition," 2016, *arXiv:1611.10080*. [Online]. Available: http://arxiv.org/abs/1611.10080
[79] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4990–4999.
[80] P. Wang *et al.*, "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1451–1460.
[81] F. Shen, R. Gan, S. Yan, and G. Zeng, "Semantic segmentation via structured patch prediction, context CRF and guidance CRF," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1953–1961.
[82] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6819–6829.
[83] S. Chandra, C. Couprie, and I. Kokkinos, "Deep spatio-temporal random fields for efficient video segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8915–8924.
[84] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun, "Efficient uncertainty estimation for semantic segmentation in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 520–535.
[85] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, "Symbolic graph reasoning meets convolutions," in *Proc. NIPS*, 2018, pp. 1853–1863.
[86] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng, "Strip pooling: Rethinking spatial pooling for scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4003–4012.
[87] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7519–7528.
[88] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[89] K. Chen *et al.*, *Mmdetection*. Accessed: Sep. 2018. [Online]. Available: https://github.com/open-mmlab/mmdetection
[90] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[91] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.

Xiangtai Li (Graduate Student Member, IEEE) received the B.E. degree from Beijing University of Post and Telecommunications (BUPT), Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the Key Laboratory of Machine Perception, School of Electrical Engineering and Computer Science, Peking University (PKU). His research interests include computer vision, segmentation, and scene understanding.

Xia Li received the bachelor's degree from Beijing University of Posts and Telecommunications (BUPT) in 2017 and the master's degree from Peking University (PKU) in 2020. He is currently pursuing the Ph.D. degree with ETH Zurich, Switzerland. His research interest lies in image segmentation, object tracking, and scene understanding.

Ansheng You received the bachelor's and master's degrees from Peking University (PKU) in 2017 and 2020, respectively. He is currently an Algorithm Engineer at Alibaba DAMO Academy for Discovery. His research interests lie in image segmentation and fast video processing.

Li Zhang received the Ph.D. degree in computer science from Queen Mary University of London. He was a Research Scientist at Samsung AI Center Cambridge and a Postdoctoral Research Fellow with the University of Oxford. He is currently an Associate Professor with the School of Data Science, Fudan University. His research interests include computer vision and deep learning.

Guangliang Cheng received the Ph.D. degree from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing. He is currently a Senior Research Manager at SenseTime. Before that, he was a Postdoctoral Researcher with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences. His research interests include autonomous driving, scene understanding, domain adaptation, and remote sensing image processing.

Kuiyuan Yang received the B.E. and Ph.D. degrees in automation from the University of Science and Technology of China, Hefei, China, in 2007 and 2012, respectively. Before joining DeepMotion as a Co-Founder in 2017, he worked as a Researcher at Microsoft Research Asia from 2012 to 2017. His current research interests include computer vision, deep learning, and autonomous driving. He was a recipient of the Best Paper Award at the International Multimedia Modelling Conference 2010.

Yunhai Tong received the Ph.D. degree in computer science from Peking University in 2002. He is currently a Professor with the School of Electronics Engineering and Computer Science, Peking University. His main research interests include data mining, media intelligent computing, and deep learning.

Zhouchen Lin (Fellow, IEEE) received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is a fellow of IAPR. He is the Area Chair of CVPR 2014/2016/2019/2020/2021, ICCV 2015, NIPS 2015/2018/2019/2020, AAAI 2019/2020, IJCAI 2020/2021, and ICML 2020. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*.